

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Màster:

AUTOMÀTICA I ROBÒTICA

Tesi de Màster

ESTUDI I MODELITZACIÓ DELS SGBD ROUND ROBIN
PEL TRACTAMENT DE SÈRIES TEMPORALS

Aleix Llusà Serra

Directors: Teresa Escobet Canal i Sebastià Vila-Marta

Curs Acadèmic 2010/11

Juny 2011

Primera edició: juny de 2011.

Primera versió: 1.0.1 (composta 12 de setembre de 2012).

Amb el suport de la Universitat Politècnica de Catalunya (UPC).

Amb la col·laboració del departament de Disseny i Programació de Sistemes Electrònics (DiPSE) i la Càtedra de Programari Lliure (CPL).



Copyright (C) 2011 Aleix Llusà Serra.

Aquesta obra està sotmesa a una llicència de Reconeixement-CompartirIgual 3.0 No adaptada de Creative Commons. Per veure una còpia de la llicència, visiteu <http://creativecommons.org/licenses/by-sa/3.0/deed.ca> o envieu una carta a Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

Aleix Llusà Serra

Departament de Disseny i Programació de Sistemes Electrònics de la Universitat Politècnica de Catalunya (DiPSE-UPC)

Escola Politècnica Superior d'Enginyeria de Manresa (EPSEM), Av. de les Bases de Manresa, 61-73, 08242 Manresa (Barcelona), CATALUNYA

aleix@dipse.upc.edu

El codi font \LaTeX del document es troba a <http://escriny.epsem.upc.edu/projects/rrb/>

Abstract

This project deals with the storage and treatment of time series obtained from data acquisition. The data acquired is treated by time series database management systems.

In this field, RRDtool is a DBMS devoted to time series data acquisition. In this project a formal model is formulated in order to describe RRDtool's data structure and behaviour, which has been absent until now. This model is known as Round Robin Database model (RRD) and improves the knowledge in time series data mining.

Keywords: Time series, data acquisition, Time series data mining, Round Robin Database model (RRD), DBMS RRDtool.

Resum

Aquest projecte s'emmarca en l'emmagatzematge i el tractament de sèries temporals que s'obtenen com a resultat d'una adquisició de dades. Un cop adquirides, les dades són gestionades per sistemes de gestió de bases de dades específics per a sèries temporals.

En aquest context, existeix l'SGBD RRDtool especialitzat en la gestió d'adquisicions de dades provinents de sèries temporals. Aquest projecte proposa un model formal, inexistent fins a l'actualitat, per tal de descriure'n l'estructura i el comportament. Aquest model s'anomena model de dades Round Robin (RRD) i obre les portes a un millor coneixement dels SGBD per a sèries temporals.

Paraules clau: sèries temporals, adquisició de dades, SGBD per a sèries temporals, model de dades Round Robin (RRD), SGBD RRDtool.

Agraïments

Amb el suport de la Universitat Politècnica de Catalunya (UPC). Amb l'agraïment al departament de Disseny i Programació de Sistemes Electrònics (DiPSE) i a la Càtedra de Programari Lliure (CPL).

Al Tobias Oetiker, l'Alex Van den Bogaerdt i companyia per dissenyar i mantenir el sistema de gestió de bases de dades més original que mai s'hagi vist.

Al Sebastià i a la Teresa per la seva gran paciència i dedicació.

Al Melen, per ser el més ràpid de la comarca.

A la Núria, que li'n dec una de molt grossa!

Índex

1	Introducció	7
1.1	Motivació	8
1.2	Objectius	9
1.3	Estructura del document	9
2	Estat actual	11
2.1	Mineria de sèries temporals	11
2.2	SGBD per sèries temporals	12
2.3	Base de dades RRDtool	14
3	Magnituds i comptadors	15
3.1	Comptadors	15
3.1.1	Normalització	21
3.2	Magnituds	25
3.2.1	Emmagatzematge relatiu	25
3.2.2	Emmagatzematge absolut	27
3.3	RRDtool	29
4	L'SGBD RRDtool	31
4.1	Introducció	31
4.2	Comparació amb altres SGBD	33
4.2.1	Gestor de RRDtool	33
4.3	Model Round Robin	34
4.3.1	Interpolació i consolidació	35
4.3.2	Tipus de dades	37
4.4	Operacions	38
4.4.1	Temps: <i>Unix time stamps</i>	39
4.4.2	Creació: <i>create</i>	39
4.4.3	Inserció: <i>update</i>	42
4.4.4	Consulta: <i>fetch</i>	42
4.4.5	Gràfics: <i>graph</i>	43
5	Funcionament intern de RRDtool	47
5.1	Transformació	48
5.1.1	Velocitat o magnitud: <i>Gauge</i>	48
5.1.2	Comptador monòton: <i>Counter</i>	51

5.1.3	Comptador doble: <i>Derive</i>	53
5.1.4	Comptador relatiu: <i>Absolute</i>	56
5.2	Normalització	59
5.2.1	Mostreig en temps real	60
5.2.2	Ultramostreig	63
5.2.3	Inframostreig	66
5.2.4	Tractament de dades desconegudes	68
5.2.5	Algoritme de normalització	76
5.3	Consolidació	77
5.3.1	Dades desconegudes	79
6	Model RRD	81
6.1	Introducció	81
6.2	Mesura	82
6.3	Sèrie temporal	83
6.3.1	Regularitat de les sèries temporals	84
6.3.2	Representació de sèries temporals	84
6.4	Buffer	86
6.4.1	Interpolació	87
6.4.2	Consolidació	88
6.5	Disc	89
6.6	Disc Round Robin	89
6.7	Base de dades Round Robin	91
6.8	Resum	93
7	Implementació de referència del model RRD	95
7.1	Introducció	95
7.2	Mesura	96
7.3	Sèrie temporal	96
7.4	Buffer	99
7.4.1	Interpolador mitjana	101
7.5	Disc	101
7.6	Disc Round Robin	103
7.7	RRD	103
7.8	Funcionament	106
7.9	Resum	108
8	Conclusions	109
8.1	Treball futur	112

1 Introducció

En el camp de l'automàtica, hi ha un àmbit dedicat a construir sistemes per a la interactuació i el control de l'entorn; són coneguts com els sistemes SCADA (*Supervisory Control And Data Acquisition*). Una de les parts importants dels SCADA és la recollecció de dades, a la qual també se l'anomena adquisició de dades o monitoratge.

Monitorar o supervisar significa estar alerta de l'estat d'un sistema. Aquesta tasca és duta a terme pels monitors, els quals recullen dades d'uns sensors i gestionen alarmes quan hi ha problemes en les variables mesurades.

Un monitor pot arribar a recollir un gran nombre de dades, ja que se solen monitorar molts sensors amb una freqüència elevada. No és rar trobar sistemes que monitoren centenars de sensors cada segon. Amb aquestes característiques, només una petita part de les dades pot ser observada en línia amb el procés (*online*). Això no obstant, l'anàlisi d'aquestes dades aporta informació important per diagnosticar el sistema, detectar errors o fer predicció i per tant cal emmagatzemar-les.

Els monitors usen sistemes especialitzats per gestionar l'emmagatzemat de les dades. Aquests sistemes han de facilitar consulta les dades, operar-hi o presentar-les gràficament fins i tot quan el volum de dades arxivades és molt gran.

En aquest context de monitoratge, és útil considerar que les dades recollides són sèries temporals. Aleshores es pot aplicar la teoria sobre les sèries temporals en els diferents procediments de tractament de dades.

Aquest document s'estructura al voltant dels sistemes d'emmagatzemament i tractament de dades com a sèries temporals. Concretament se centra en els sistemes de gestió de bases de dades (SGBD) que s'ocupen de sèries temporals. Els SGBD habituals, com per exemple els de model relacional, no són adequats per aquests casos ja que no disposen de les funcionalitats adequades per gestionar i explotar correctament la informació de les sèries temporals. Existeixen, però, SGBD's especialitzats en la gestió de dades provinents de sèries temporals. Entre aquests destaca RRDtool per haver-se convertit en l'estàndard *de facto* a nivell productiu. RRDtool està basat en uns conceptes que es poden agrupar sota un model anomenat model de dades Round Robin (RRD). El model RRD està específicament pensat per a les sèries temporals. RRDtool, doncs, és un SGBD molt adequat per a ser usant en l'àmbit de monitoratge.

Segons Date, [8], “una base de dades és un contenidor informàtic per a una col·lecció

de dades". En el context dels SGBD's cal diferenciar:

- El model d'una bases de dades, que és el model matemàtic a on es descriu teòricament l'estructura de les dades, per exemple el model relacional o model Round Robin.
- El sistema de gestió de bases de dades, que és la implementació d'un model de dades, per exemple postgresql o RRDtool.
- La base de dades, que és una instància d'un sistema de gestió de bases de dades, per exemple la base de dades dels estudiants o la base de dades de la temperatura de l'escola.

Fins on coneixem, RRDtool és l'únic SGBD que implementa el model RRD. En aquest projecte s'estudia RRDtool per tal de proposar, per primera vegada, una descripció formal del seu model. L'existència d'aquest model obre les portes a un millor coneixement d'aquests tipus d'SGBD's, facilita noves implementacions i, finalment, permet avançar en el seu estudi.

1.1 Motivació

Actualment, els sistemes de telecontrol són complexos i hi ha una gran quantitat d'informació a gestionar. En aquests grans sistemes de telecontrol apareixen problemes en la recollecció de dades, com per exemple els forats, descrits per Quevedo *et al.*, [28], o el tractament de recolleccions massives que descriuen Camerra i Keogh, [5]. Aquestes recolleccions de dades s'estudien com a sèries temporals degut a la seva naturalesa de seqüència de valors, però en l'emmagatzematge a les bases de dades no es té en compte aquesta estructura.

Com s'ha dit a l'apartat anterior, els SGBD's que han de gestionar aquesta informació haurien de poder tractar convenientment la informació que prové de sèries temporals. En aquest sentit, actualment s'estan investigant en algunes tècniques, com iSAX, [13], o T-Time, [2]. En aquest context l'SGBD anomenat RRDtool destaca per ser l'estàndard *de facto* en l'emmagatzematge i tractament de sèries temporals, [24].

RRDtool no disposa de cap model matemàtic que en descriu el comportament. Disposar d'un model matemàtic és una fita important com demostra l'experiència històrica en l'àmbit dels SGBD's relacionals, [8]. Disposar d'un model per a RRDtool, permetrà aprofundir en aquests tipus de SGBD's i estudiar com poden ajudar, per exemple, en l'aplicació de mètodes de validació i predicció com els que proposa Puig *et al.* o les reduccions de dimensió que proposen Camerra i Keogh.

La recerca en l'àmbit de la mineria de dades de sèries temporals, àmbit en el que se situa l'emmagatzematge i tractament de sèries temporals, ha experimentat un important increment darrera dècada. En fan esment Tak-chung Fu, [9], que el

considera un problema no solucionat, i Yang i Wu, [34], que el qualifiquen com un dels deu reptes a resoldre en la mineria de dades.

1.2 Objectius

En aquest projecte es plantegen els objectius següents:

- Situar l'estat actual de l'emmagatzematge i del tractament de sèries temporals.
- Estudiar el sistema de gestió de bases de dades RRDtool.
- Dissenyar un model de dades que descrigui l'estructura i el comportament dels SGBD per a sèries temporals.
- Proposar una implementació de referència del model dissenyat.
- Proposar millores i treballs futurs al voltant del model dissenyat.

1.3 Estructura del document

A continuació d'aquest primer capítol d'introducció als sistemes de gestió de bases de dades de sèries temporals i de definició dels objectius del projecte, aquest document s'estructura en els capítols següents.

En el capítol 2, complementant al capítol d'introducció, s'exposa l'estat actual de la mineria de dades de sèries temporals. En aquest context se situen els sistemes de gestió de bases de dades que emmagatzemen i tracten sèries temporals.

A continuació el cos principal del document se centra en els SGBD Round Robin. En una primera part es descriu el SGBD RRDtool.

En el capítol 3 s'aclareix perquè RRDtool, en el context de sèries temporals, està pensat per velocitats mitjanes de comptadors i com es poden tractar variables que són magnituds.

En el capítol 4 s'introdueix un concepte bàsic de base de SGBD RRD a partir del que es pot observar a RRDtool. S'expliquen algunes particularitats que tenen les bases de dades gestionades per RRDtool, com per exemple que no creixen en mida un cop creades o que són capaces de representar les dades gràficament.

En el capítol 5 es detalla el funcionament intern de RRDtool, tot seguint les etapes per les quals passa una dada des de que s'ha mesurat fins que queda desada.

En una segona part, a partir dels conceptes observats a RRDtool, es proposa el model RRD.

El capítol 6 correspon a la definició d'aquest model de dades. En el capítol 7,

1 Introducció

s'implementa el model utilitzant el llenguatge de programació Python. S'exposa un exemple de funcionament.

Finalment en el capítol 8, es tanca el document amb un resum de l'exposat i les conclusions que es poden extreure del model de dades per a sèries temporals que s'ha dissenyat. També es proposen els treballs futurs que es poden dur a terme a partir del model dissenyat.

2 Estat actual

En aquest capítol se situen els sistemes de gestió de bases de dades (SGBD) per sèries temporals en el context de la mineria de dades de sèries temporals (*time series data mining*), el qual també es considerat com mineria de dades per detectar automàticament coneixement (*knowledge discovery databases*). Els SGBD de model Round Robin (RRD) pertanyen a aquest context ja que emmagatzemen sèries temporals de les quals es vol aconseguir informació rellevant.

El capítol comença resumint l'estat de les sèries temporals en aquest camp de mineria; és a dir d'emmagatzematge i tractament. A continuació es llisten algunes aplicacions informàtiques que han implementat models de la mineria de sèries temporals. Finalment, es descriu l'estat actual de l'aplicació RRDtool, la qual també es classifica en aquest camp.

2.1 Minería de sèries temporals

L'anàlisi de sèries temporals abasta camps molt diferents com ara la predicció econòmica, la medicina, la meteorologia, la qualitat industrial, etc. En aquest context, la mineria de sèries temporals tracta de gestionar col·leccions cronològiques de dades que tenen una mida gran i contínuament estan en creixement. Aquest apartat se centra en l'estat actual de la mineria de sèries temporals, àmbit que, en la darrera dècada, ha experimentat un important increment de la recerca.

En un article molt recent de Tak-chung Fu, [9], es fa esment d'aquest increment de recerca i es resumeix l'estat actual de forma exhaustiva. Fu conclou que la recerca s'ha centrat en tasques de mineria però no s'ha resolt del tot el problema de la representació de sèries temporals.

Segons Keogh i Kasetty, [12], les quatre tasques que centren l'atenció de la recerca actual de sèries temporals són l'indexat, l'agrupament, la classificació i la segmentació. A més, Keogh compara els experiments duts a terme en aquests camps. Un pas comú previ a aquestes quatre tasques és el de representació de la sèrie temporal. Keogh *et al.*, [14, 15], investiguen la representació de sèries temporals a trossos lineals (PLR, *Piecewise Linear Representation*). Keogh fa notar que la representació PLR és l'habitual degut a que la visió de l'ésser humà segmenta les corbes en línies rectes. Més tard, Keogh, [16, 17], explora la representació de sèries temporals per tal de reduir la dimensió d'una sèrie temporal i poder-la indexar més fàcilment i proposa

dues tècniques eficients en el càlcul: la PAA (*Piecewise Aggregate Approximation*) i la APCA (*Adaptive Piecewise Constant Approximation*), ambdues basades en la representació a trossos constants de la sèrie temporal. D'aquestes dues tècniques Keogh conclou que mantenen una bona aproximació a la sèrie temporal i que a més tenen molt menys cost de càlcul que altres de més complicades, com ara la *Discrete Fourier Transform* (DFT), la *Singular Value Decomposition* (SVD) o la *Discrete Wavelet Transform* (DWT).

Tal com expliquen Quevedo *et al.*, [28], en un sistema complex de telecontrol hi ha una gran quantitat d'informació a manipular que s'obté de diversos sensors distribuïts pel camp de mesura, aquesta informació s'anomena variables mesurades. Un SCADA (*Supervisory Control And Data Acquisition*) és el sistema encarregat de recollir i centralitzar les variables de manera periòdica en el temps. En el moment de recollecció de dades apareixen dos problemes: valors que en un instant de temps prefixat no s'han pogut recollir i valors que són falsos. Les tècniques de bases de dades no poden emmagatzemar les dades amb aquests dos tipus de problema ja que aleshores els registres històrics quedarien falsejats. Així doncs, cal comprovar que les dades emmagatzemades són correctes, segons un procés de validació, i modificar-les en el cas que siguin falses, segons un procés de reconstrucció. Quevedo, [28], aplica aquests processos a xarxes de distribució d'aigua.

Els mètodes de validació i reconstrucció es poden basar en anàlisis senzilles del senyal o en comparacions del valor real amb models de predicció de dades. Quan les dades es tracten com a sèries temporals, hi ha mètodes de predicció específics. Tot i que la teoria de sèries temporals permet establir aquests mètodes de predicció i reconstrucció, els SGBD habituals, com ara els de model relacional, no ho faciliten. Per tal d'aplicar aquests mètodes a les sèries temporals de manera eficient, els SGBD s'han d'especialitzar en el tractament de sèries temporals.

2.2 SGBD per sèries temporals

Per poder analitzar les dades de manera eficient cal disposar de bases de dades específiques, a més cada cop el volum de dades a tractar és més crític degut a que hi ha més facilitat a capturar-les i més capacitat per emmagatzemar-les. La diferència principal de les sèries temporals amb altres tipus de dades és que els valors són dependents d'una variable: el temps. Com a conseqüència, qualsevol base de dades que hi vulgui tractar no ho pot fer de manera independent pels valors i pel temps; ha de conservar la coherència temporal.

Tal com diu Aßfalg, [1], la coherència temporal pot ser vista des de dues vessants. La primera, a la qual anomena *bitemporal data*, consisteix en expressar el temps vàlid durant el qual un esdeveniment és cert i el temps de transacció durant el qual l'esdeveniment és guardat a la base de dades, és a dir consisteix a descriure dos estats, cert o fals, per cada observació. La segona, a la qual anomena *time series*

data, consisteix a descriure col·leccions de dades en funció del temps. A més diu que les primeres poden ser expressades amb les segones.

Els SGBD relacionals són capaços d'implementar el primer tipus de coherència, les *bitemporal data*; llavors es classifiquen sota el nom de bases de dades temporals, [8, 31]. Però el model relacional no és suficient pel segon tipus: les sèries temporals. Tot i que en principi no hi hauria cap problema a utilitzar una base de dades relacional per a sèries temporals, enteses com a dades històriques, la pròpia naturalesa dels sistemes relacionals dificulta les operacions necessàries. Aquestes operacions per sèries temporals es basen en rangs de temps i precisen conversions de fusos horaris i rotacions dels registres de les taules, sinó el nombre de files creixeria de forma indefinida.

Els SGBD que implementen operacions per a sèries temporals es poden anomenar *Time Series Database Systems* (TSDS), [32]. Les TSDS Estan optimitzades per gestionar les dades segons les operacions de temps i rotació, les quals són molt comunes en la gestió de les sèries temporals. A més també cal controlar el creixement de la base de dades i la consulta ha de ser flexible i d'alta velocitat, [5]. Per exemple, s'han de poder visualitzar les evolucions tant d'una setmana com d'un any sense haver de fer càlculs complicats amb els valors emmagatzemats. A continuació es llisten dues bases de dades optimitzades per a sèries temporals.

Akfalq, [1], presenta un TSDS que és capaç de cercar similituds, també anomenades distàncies, entre sèries temporals. Principalment utilitza llindars per comparar en cada interval si les dues sèries temporals s'assemblen. A partir d'aquest mètode desenvolupa algorismes que calculen de manera eficient per a les sèries temporals i en concret els implementa en una aplicació anomenada T-Time, la qual descriu a [2].

Keogh i Camerra [5, 29], estudien l'anàlisi i l'indexat de col·leccions massives de sèries temporals. Descriuen que el problema principal del tractament rau en l'indexat de les sèries temporals i proposen mètodes per calcular-lo de manera eficient. El mètode principal que desenvolupen està basat en l'aproximació a trossos constants de la sèrie temporal (PAA, [16]) i ho implementen en una estructura de dades que anomenen iSAX (*indexable Symbolic Aggregate approXimation*), [13]. Amb aquesta eina s'obtenen representacions de sèries temporals que permeten reduir l'espai emmagatzemat i indexar tant bé com altres mètodes de representació més complexos.

En resum, aquests SGBD per sèries temporals bàsicament resolen els problemes d'anàlisi de sèries temporals. Però cap d'aquestes sol atendre la relació entre la base de dades i el sistema de monitoratge, és a dir la manera com s'adquireixen les dades. En aquest pas intermig hi ha un sèrie de problemes, com per exemple forats, dades falses, irregularitat en els temps de mostreig, que cal gestionar correctament. Concretament un dels problemes que no s'atén és el de mostreig irregular ja que es considera que les mostres estan a intervals regulars (equi-espaiades) encara que els sistemes de monitoratge informàtics sovint no són capaços de complir-ho amb exactitud sinó que presenten una certa variació en els temps de mesura.

Així doncs, quan es prenen mesures d'un sistema productiu, aquests problemes apareixen i són de difícil solució. Les bases de dades RRDtool tenen en compte aquests problemes intermitjos entre el sistema de monitoratge i el sistema d'emmagatzematge i tractament.

2.3 Base de dades RRDtool

En aquest apartat es presenta el TSDS anomenat RRDtool. Aquest sistema, que serà objecte d'un estudi acurat en els capítols 4 i 5, s'ha pres com a referència en aquest treball.

RRDtool és un SGBD per a sèries temporals que despunta en l'àmbit de programari lliure. Hi ha una llista de projectes que utilitzen RRDtool que poden trobar-se indicats a l'apartat *Projects using RRDtool* de [24]. Entre d'altres, s'utilitza en sistemes de monitoratge professionals com per exemple Nagios, [21], o Icinga, [11], també populars dins del programari lliure, o en el monitor MRTG (The Multi Router Traffic Grapher), [25], del mateix creador que RRDtool. Aquests monitors fan un ús complet de les possibilitats de RRDtool i li cedeixen tot el control de l'emmagatzematge de mesures i el posterior tractament i representació gràfica de les dades. L'ús de RRDtool permet a aquestes aplicacions centrar-se plenament en la problemàtica de l'adquisició de dades i la gestió d'alarmes.

En l'evolució de RRDtool destaquen dues millores significatives. La primera, descrita per Oetiker a [22], va consistir en independitzar la base de dades RRDtool del sistema de monitoratge MRTG i dissenyar-la amb l'estructura Round Robin que la caracteritza. La segona, feta per Brutlag, [4], ha aportat la possibilitat de fer prediccions i detecció de comportaments aberrants basant-se en algorismes de predicció exponencials i de Holt-Winters.

L'evolució actual de RRDtool se centra en aspectes informàtics i consisteix a millorar la rapidesa i eficiència en el processament de les sèries temporals. És el cas de Plonka i Carder que a [6, 26] dissenyen l'aplicació `rrdcached` per incrementar el rendiment de RRDtool, la qual demostren en un sistema de monitoratge amb moltes bases de dades funcionant simultàniament. També JRobin, [20], que és una implementació en Java de RRDtool que millora els accessos de lectura i escriptura a la base de dades i té una eina de gràfics més perfeccionada. És significatiu l'ús incipient d'aquest sistema en experimentació. Zhang, [35], i Chilingaryan, [7], per exemple, usen RRDtool per emmagatzemar de dades experimentals i posteriorment fer predicció o validació.

En l'àmbit dels SGBD els sistemes relacionals van fixar una fita que ha tingut una transcendència posterior de primer ordre. En bona part aquest èxit dels SGBD relacionals es deu al fet que es basen en un model matemàtic sòlid, [8]. En el cas de RRDtool no existeix cap model que descriu el sistema i es objectiu d'aquest treball proposar-ne un. El model per a SGBD Round Robin es dissenya al capítol 6.

3 Comparació de magnituds i comptadors

Al capítol 5 s'explica profundament com es tracten les dades a RRDtool fins que queden emmagatzemades. Abans de l'etapa de normalització d'interval, hi ha una etapa de transformació a velocitat que pot quedar descontextualitzada si no s'expliquen els motius històrics que han portat a tractar les dades d'aquesta manera.

En aquest capítol es raona per què RRDtool té interès a diferenciar variables que classifica com a comptadors i magnituds i per què aplica l'etapa de transformació a velocitat descrita en detall a l'apartat 5.1. S'anomena velocitat al flux o taxa de variació dels valors quan són expressats com a ràtio magnitud per unitat de temps.

En la primera versió, RRDtool només s'utilitzava per un sistema que monitorava els comptadors de tràfic de xarxa dels encaminadors (*routers*) i els tractava de manera especial, basant-se en la velocitat mitjana del comptador. En el primer apartat es resumeix la idea principal d'aquest tractament de velocitat.

En evolucions posteriors de RRDtool, aquest tractament s'ha mantingut per totes les dades; és a dir que totes les dades es transformen i s'interpreten com a velocitat, fins i tot les que no són comptadors. Es pot concloure que els comptadors han esdevingut el 'cas bàsic' de RRDtool. En un segon apartat s'analitzen les conseqüències que això comporta per a les dades que es mesuren com a magnituds, per exemple la temperatura.

Al tercer apartat es detallen els noms que utilitza RRDtool per emmagatzemar les variables classificades com a comptadors i com a magnituds i quines possibilitats ofereix per representar-les.

3.1 Comptadors

Com ja s'ha comentat anteriorment, cal situar els inicis de RRDtool en el monitoratge de xarxa de comunicacions amb comptadors. Aquests comptadors són una variable que es va incrementant monòtonament; és a dir són una funció creixent infinita. Per aquesta particularitat, els anomenem comptadors monòtons.

Quan es pren una mesura d'un comptador monòton s'obté un valor que indica el total de comptatge des de l'inici del comptador; és a dir que sempre es mesura amb

temps (s)	0	10	20	25	50
valor (bytes)	0	1	3	4	10

Taula 3.1: Mesures d'un comptador monòton

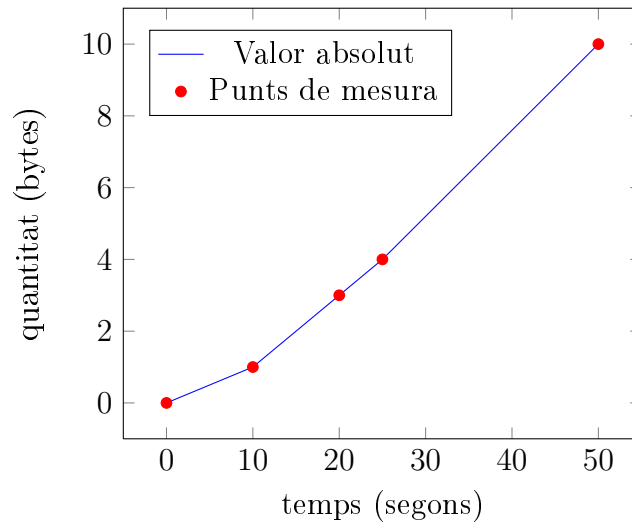


Figura 3.1: Evolució del valor d'un comptador monòton

referència absoluta. Per exemple, en diferents instants de temps es prenen mesures d'un comptador de tràfic de xarxa i s'obté el total de bytes que l'encaminador ha processat fins aleshores. A la taula 3.1 hi ha unes mesures d'exemple que es poden visualitzar gràficament a la figura 3.1, entre mesures es considera que ha comptat a ritme constant; és a dir linealment.

En un pas següent, aquestes mesures preses es poden emmagatzemar a la base de dades, però a RRDtool aquests valors li presenten tres problemes. El primer és que en tractar-se d'una funció sempre creixent, el gràfic que en resulta no és de molta ajuda pels usuaris que l'han d'interpretar. El segon problema és degut a que a la pràctica no es pot comptar fins a un valor infinit sinó que queda acotat en un rang definit per un valor mínim i un valor màxim, anomenat fons d'escala. Així doncs, els comptadors monòtons s'incrementen fins que arriben al fons d'escala i tornen a començar pel valor mínim. Aquest efecte, que s'anomena desbordament, trenca la referència absoluta del comptador, tot i que pot solucionar si el rang del comptador és conegut¹ i si se suposa que entre dues mesures només hi ha hagut un desbordament (v. eq. 3.1). Però aleshores apareix el tercer problema degut a que en les memòries dels computadors tampoc es pot desar un valor infinit.

Així doncs, en comptes del valor del comptador es proposa desar els increments del

¹RRDtool només admet valors mínims de 0 i fons d'escala de 2^{32} (32 bits) o de 2^{64} (64 bits) degut a que fa autodetecció de desbordament per als comptadors d'aparells electrònics, els quals normalment són de 32 o 64 bits.

temps (s)	0-10	10-20	20-25	25-50
valor (bytes)	1	2	1	6

Taula 3.2: Increments d'un comptador monòton

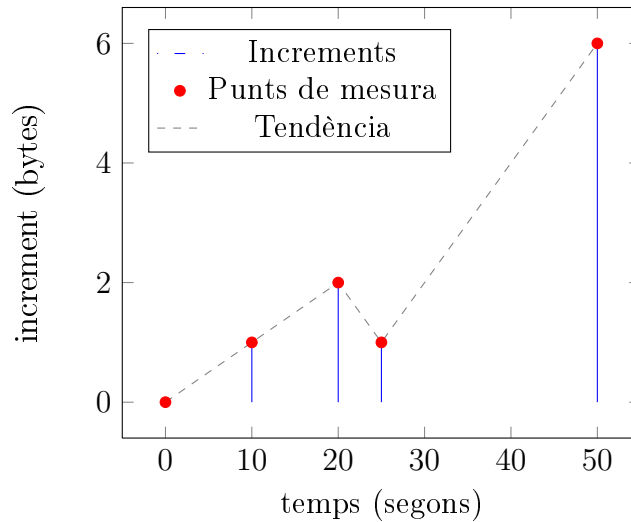


Figura 3.2: Increments d'un comptador

comptador en cada interval, calculats segons l'equació següent:

$$\Delta_i = \begin{cases} x_i - x_{i-1} & \text{si } x_i \geq x_{i-1} \\ (\text{fons d'escala} - x_{i-1}) + (x_i - \text{valor mínim}) & \text{si } x_i < x_{i-1} \end{cases} \quad (3.1)$$

on x_i és el valor mesurat en l'instant actual t_i , x_{i-1} és el valor mesurat en l'instant anterior t_{i-1} i Δ_i és l'increment del comptador en l'interval de temps $(t_{i-1}, t_i]$.

Aplicant l'equació 3.1 a les dades de la taula 3.1 s'obtenen els increments per cada interval a la taula 3.2.

A la figura 3.2 es mostra les noves dades calculades en forma d'increments, els quals es representen amb impulsos per reforçar la idea de variable de naturalesa discreta del comptador com a mesurador d'increments. També es representa la línia de tendència, tal com es representaria si els increments del comptador provinguessin d'una variable contínua, aquesta línia de tendència dóna una idea de com està evolucionant el comptador. Per un interval concret mostra que si la mesura s'hagués pres a un altre punt, l'increment hauria estat un altre valor, més gran o més petit, però aquesta interpolació queda limitada a l'interval, és a dir que només té sentit una mesura per interval.

Si es vol veure un comptador d'increments com a variable de naturalesa contínua, cal representar-ne el valor relatiu com es fa a la figura 3.3. Anomenem valor relatiu al valor del comptador d'increments ja que mostra la quantitat comptada des de l'última mesura. Per representar el valor relatiu, el comptador es posa al valor

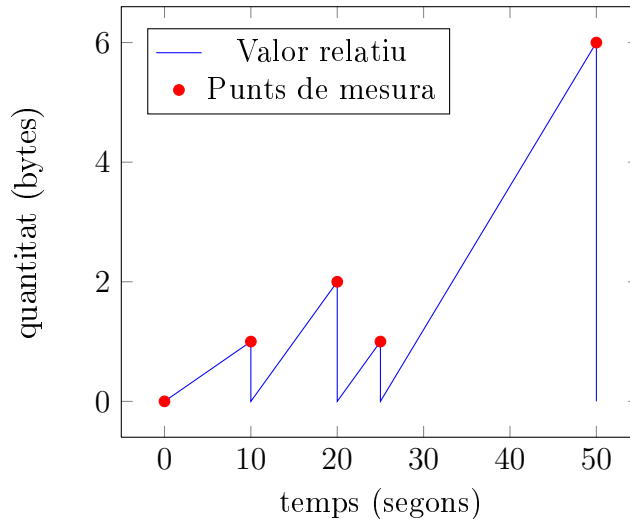


Figura 3.3: Valor relatiu d'un comptador

mínim cada cop que es pren una mesura i se suposa que el comptador s'incrementa linealment, de la mateixa manera que s'ha suposat en el comptador representat en absolut de la figura 3.1.

A diferència de la línia de tendència de la figura 3.2, el valor relatiu sí que pot ser interpolat en més d'un punt de la mateixa manera que el comptador representat en absolut a la figura 3.1, perquè ambdós només indiquen el valor que es llegirà del comptador en aquell moment.

Per a la interpretació dels increments del comptador, és a dir d'un valor per interval, s'entén més si es representen els valors en un gràfic de barres, com es fa a la figura 3.4. Es representa la freqüència de comptatge per cada interval i també la línia de tendència que s'ha d'interpretar com a la figura 3.2.

En el gràfic de barres a l'eix horitzontal hi ha els intervals representats com a categories de manera que no ajuda a comparar quan hi ha intervals de mida diferent. La proposta següent és dibuixar l'histograma, a on l'àrea de cada barra guarda proporció amb la freqüència de l'interval, com es mostra a la figura 3.5.

L'histograma en aquest cas és més informatiu pels usuaris ja que es veu el ritme del comptador sense que l'afecti el mostreig amb mides diferents d'interval. Els valors de l'eix vertical, la densitat de freqüència de cada interval, estan assenyalant la velocitat mitjana del comptador en cada interval. De fet, aquesta velocitat mitjana es pot calcular segons l'equació següent:

$$v_i = \frac{\Delta_i}{t_i - t_{i-1}} \quad (3.2)$$

on Δ_i és l'increment del comptador en l'interval de temps $(t_{i-1}, t_i]$ calculat a l'equació 3.1, t_i és l'instant de temps actual, t_{i-1} és l'instant de temps anterior i v_i és

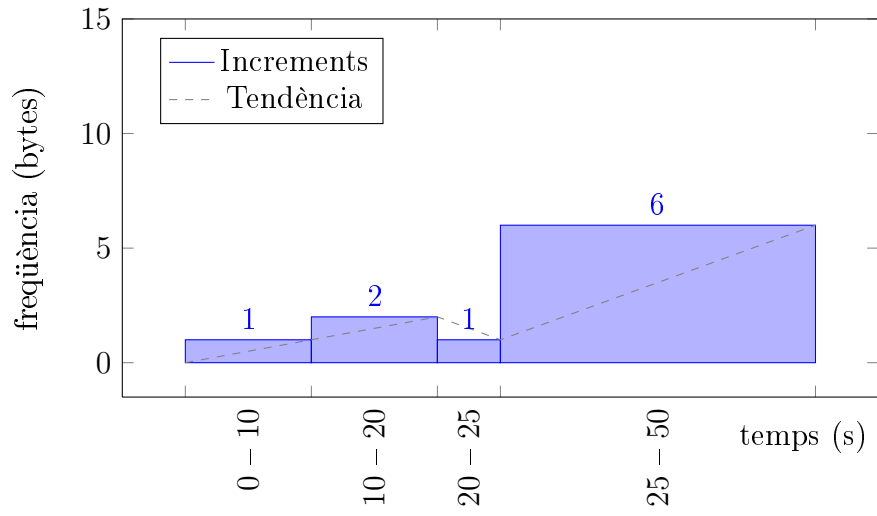


Figura 3.4: Gràfic de barres d'increments d'un comptador

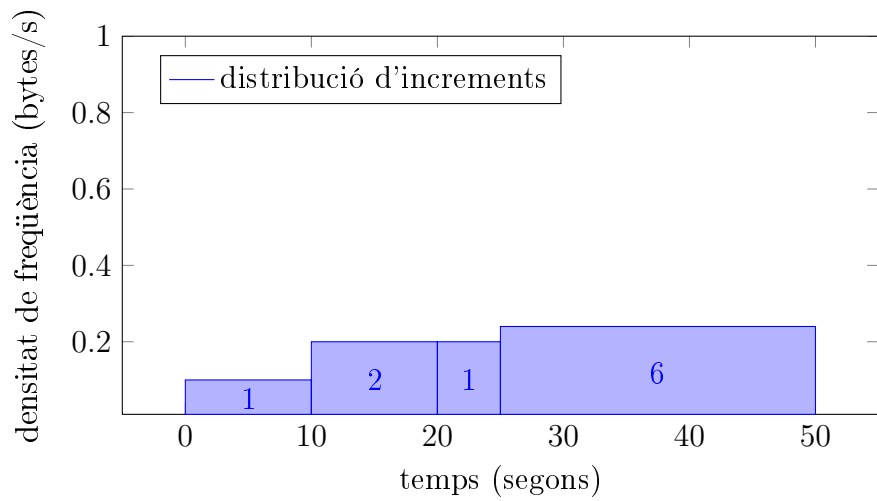


Figura 3.5: Histograma d'increments d'un comptador

temps (s)	0-10	10-20	20-25	25-50
valor (bytes)	0,1	0,2	0,2	0,24

Taula 3.3: Velocitats mitjanes d'un comptador monòton

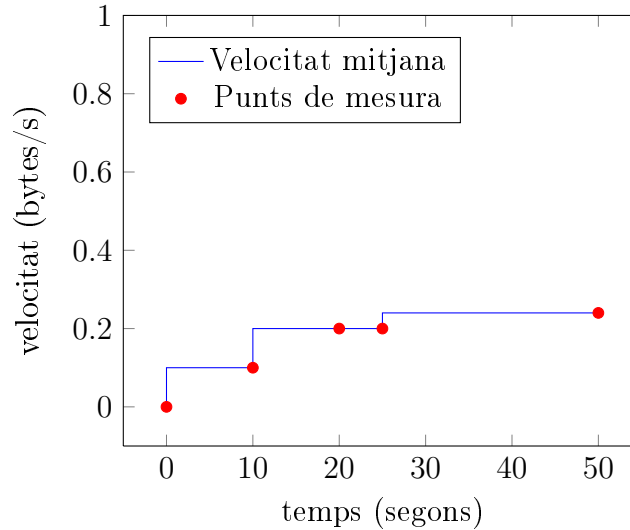


Figura 3.6: Velocitat d'un comptador

la velocitat mitjana del comptador en l'interval de temps $(t_{i-1}, t_i]$.

Aplicant l'equació 3.2 a les dades de la taula 3.2 s'obtenen les velocitats per cada interval a la taula 3.3.

La velocitat mitjana en cada interval s'ha representat a la figura 3.6 a on s'observa la mateixa forma que a l'histograma de la figura 3.5, així, tal com passa a l'histograma, l'àrea del gràfic de velocitat mitjana es correspon a cada interval amb el comptatge incremental i la suma d'increments correspon al comptatge total. També es pot veure que la velocitat mitjana es correspon amb els pendents aproximats linealment del valor del comptador representat en absolut, figura 3.1, i representat en relatiu, figura 3.3.

RRDtool decideix emmagatzemar els valors del comptador calculats amb l'equació 3.2 com a velocitats tot i que podria haver triat l'emmagatzemament dels increments. Segons RRDtool, les velocitats faciliten els càlculs interns que ha de fer la base de dades i es considera que representar la velocitat mitjana és més intuïtiu pels usuaris. Per uniformitzar, RRDtool assumeix que les dades emmagatzemades han estat prèviament transformades a velocitat i per tant aplica les operacions i les representacions amb aquesta idea. A continuació es mostra com RRDtool normalitza els interval següent amb aquesta idea de velocitat i posteriorment es mostra quines conseqüències té per a dades que no són interpretables com a comptador/velocitat, per exemple la temperatura.

temps (s)	0	10	20	25	30	40	50	total
increments (bytes)	0	1	2	1			6	10
increments normalitzats (bytes)	0	1	2		2,2	2,4	2,4	10

Taula 3.4: Normalització dels increments d'un comptador monòton

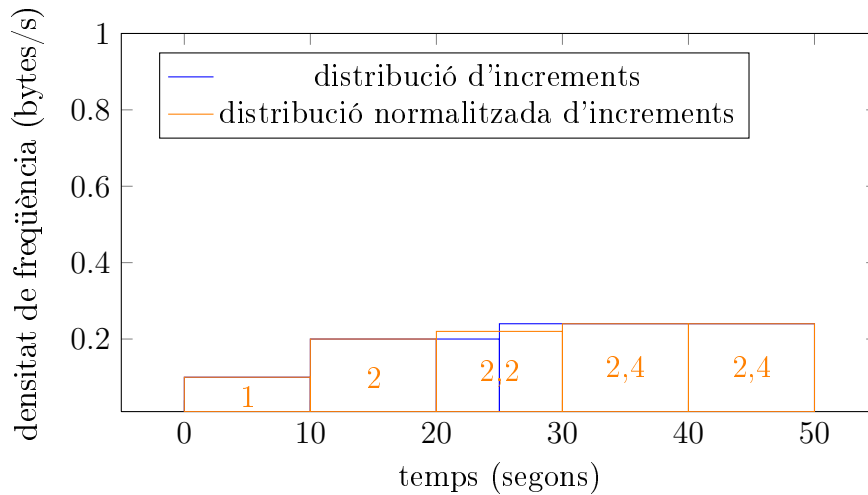


Figura 3.7: Histograma normalitzat d'increments d'un comptador

3.1.1 Normalització

Una de les parts importants de RRDtool consisteix a normalitzar els intervals per tal que tots tinguin la mateixa durada de temps. L'objectiu és doble: simplificar l'estructura de la base de dades i poder aplicar operacions que necessiten que els intervals siguin regulars.

Com que normalitzar implica canvia els valors del mostreig, s'ha de decidir un criteri per la informació que es vol conservar, per exemple la mitjana o el màxim. RRDtool escull segons el criteri de conservar el comptatge total ja que com s'ha vist abans es desen els comptadors amb la forma de velocitats mitjanes, les àrees de les quals són el comptatge incremental del comptador. El criteri del total, també anomenat àrea de sota la corba quan s'aplica a les velocitats mitjanes, és coherent amb els comptadors de tràfic de xarxa ja el primer objectiu d'aquestes base de dades és saber contestar les consultes de quants bytes han circulat entre dos temps.

La normalització a RRDtool es calcula segons l'explicat a l'apartat 5.2. Per exemple, a la taula 3.4 es normalitzen cada 10 segons els increments de la taula 3.2. Aquests valors, calculats com a velocitat, són els que s'emmagatzemen a la base de dades.

Una bona representació gràfica es veu a l'histograma de la figura 3.7, a on ara per ser els intervals regulars la forma és molt semblant a un gràfic de barres.

temps (s)	0	10	20	25	30	40	50
valor (bytes)	0	1	3	4			10
valor normalitzat (bytes)	0	1	3		5,2	7,6	10

Taula 3.5: Normalització d'un comptador monòton

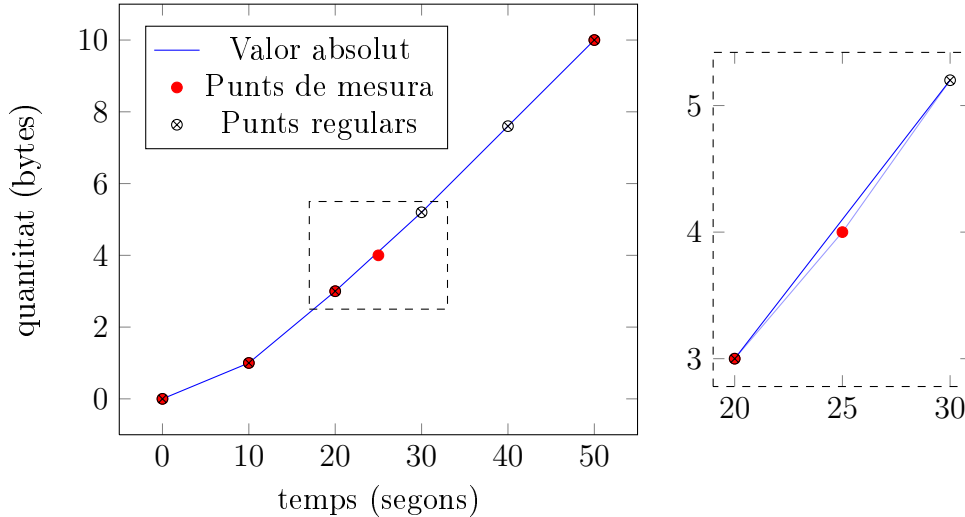


Figura 3.8: Normalització del valor d'un comptador monòton

Els valors normalitzats es calculen en valor absolut a la taula 3.5 i es representen a la figura 3.8. El gràfic és el mateix que a la figura 3.1 excepte a l'interval de temps (20, 30] a on el valor absolut ha canviat de recorregut. Per tant es veu clarament com la normalització de RRDtool reparteix els valors del comptador suposant que s'ha incrementat a velocitat constant i les mostres que queden fora de la linealitat no són recuperables.

A la base de dades hi queden emmagatzemats els valors cada 10 segons i el comptatge del comptador ha quedat repartit en aquests intervals segons el criteri de total per comptadors lineals de RRDtool, descrit a l'apartat 5.2. Ara bé, utilitzant el criteri del total també hi hauria altres normalitzacions possibles, per exemple donant més pes a l'interval amb més comptatge com es fa a l'histograma de la figura 3.9. El criteri del total el mantinc en els dos casos però un dels dos s'aproxima més al valor que ha tingut el comptador entre els punts de mostreig, els quals són els únics valors que coneixem.

Normalització per mitjana

A l'inici de l'apartat, s'ha dit que RRDtool escull el criteri de total per conservar la informació, però que hi ha altres criteris possibles com la mitjana o el màxim. A continuació s'exemplifica una normalització pel criteri de mitjana aritmètica com-

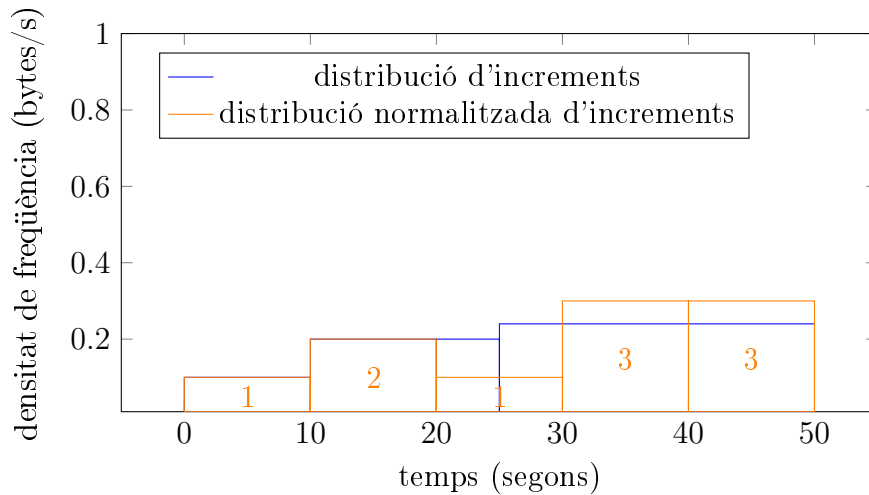


Figura 3.9: Histograma normalitzat d'increments d'un comptador, donant més pes als darrers intervals

normalització (0, 50]	criteri total	criteri mitjana
domini increments (bytes)	10	$\frac{1+2+1+6}{4} = 2,5$
domini velocitats (bytes/s)	0,2	$\frac{\frac{1}{10} + \frac{2}{10} + \frac{1}{5} + \frac{6}{25}}{4} = 0,16$

Taula 3.6: Normalització d'un comptador monòton segons dos criteris

parada amb una pel criteri de total que fa RRDtool. S'utilitzen les dades del comptador anterior i ara es normalitza amb un sol valor tot l'interval de temps (0, 50]. Els resultats es poden veure a la taula 3.6 expressats en el domini d'increments i en el domini de velocitats.

La normalització pel criteri total es mostra a l'histograma de la figura 3.10. A la figura 3.11 es mostren conjuntament la normalització pel criteri total i pel criteri mitjana representats en valor absolut. Com que no s'ha de repartir entre dos intervals més petits, en aquest cas el criteri del total es podria haver calculat com la suma d'increments en comptes de la ponderació que fa RRDtool.

Ens aquests valors es veu com la velocitat mitjana no és el mateix que la mitjana de les velocitats. La velocitat mitjana és el mateix que la mitjana ponderada de les velocitats que és el que fa RRDtool en calcular l'àrea de sota la corba, al qual es pot anomenar criteri del total. La mitjana aritmètica de les velocitats dóna idea del flux mitjana que hi ha, és a dir que reparteix equitativament la velocitat en el temps. A partir de la mitjana de la velocitat no es pot recuperar el comptatge total, en canvi l'àrea de la velocitat mitjana es correspon exactament amb el total.

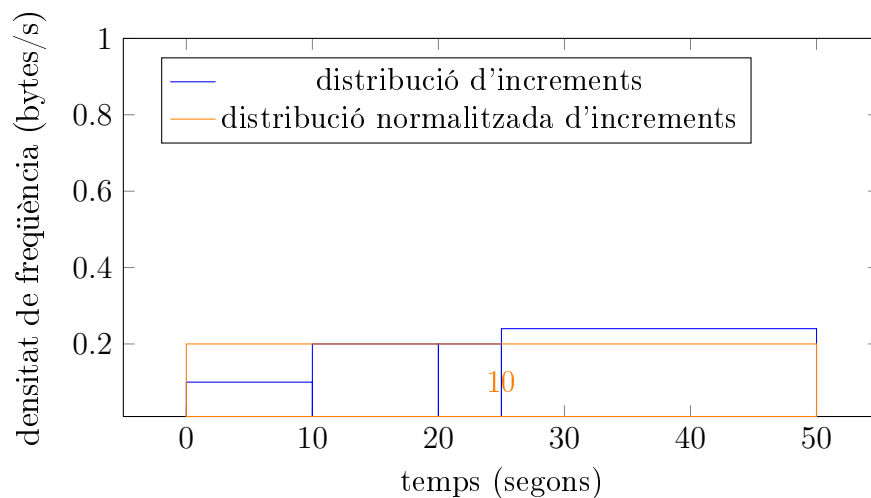


Figura 3.10: Histograma normalitzat per tot l'interval d'un comptador

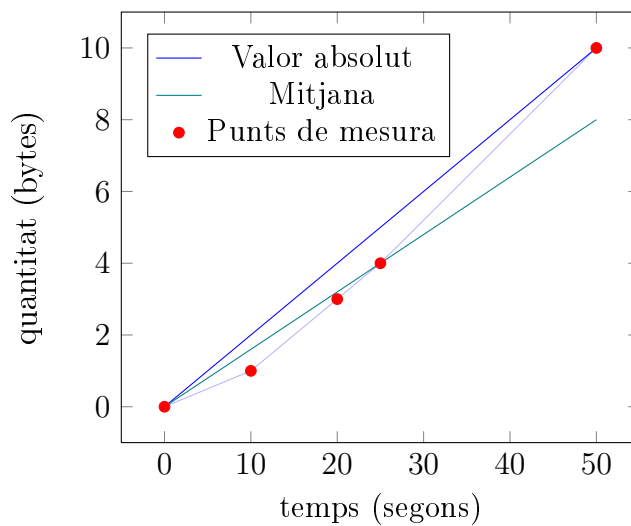


Figura 3.11: Normalització per tot l'interval del valor d'un comptador monòton

temps (s)	0	10	20	25	50
temperatura (°C)	15	25	15	20	25

Taula 3.7: Valors de temperatura

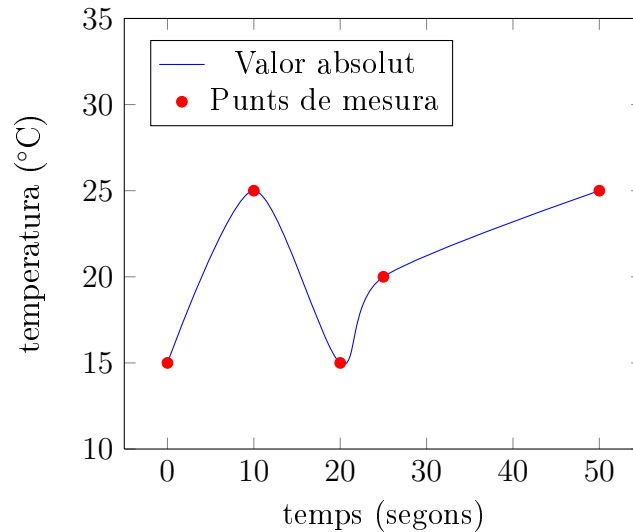


Figura 3.12: Evolució del valor de temperatura

3.2 Magnituds

De l'apartat anterior es conclou que el cas bàsic a RRDtool és la velocitat i que es normalitza i representa d'acord amb aquest concepte. Aquest esquema funciona adequadament per variables de naturalesa comptadora, però que passa quan la variable mesurada és una magnitud física, per exemple la temperatura, i per tant no es pot definir com una funció monòtona.

A continuació s'explica com RRDtool emmagatzema magnituds aplicant el seu mètode de normalització pel criteri del total, primer es considera que una magnitud és com un comptador en valor absolut i després es considera que una magnitud és la velocitat d'un comptador. Es pren com a exemple les mesures de temperatura de la taula 3.7. Aquestes mesures es representen a la figura 3.12, a on el valor absolut es dibuixa amb interpolacions suavitzades que és la forma que solen tenir les magnituds físiques.

3.2.1 Emmagatzematge relatiu

Seguint amb l'esquema de normalització pels comptadors monòtons, es pot veure el valor absolut de temperatura com un comptador que ara ja no és una funció monòtona sinó que pot ser creixent o decreixent arbitràriament en qualsevol punt.

temps (s)	0-10	10-20	20-25	25-50
increments (°C)	10	-10	5	5

Taula 3.8: Increments de temperatura

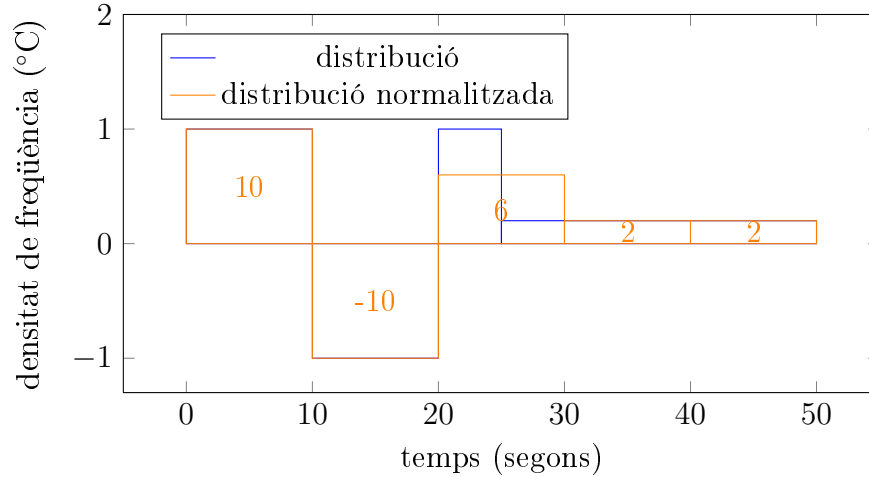


Figura 3.13: Histograma normalitzat per a increments de temperatura

Per no ser una funció monòtona ja no hi segueixen havent els mateixos problemes i es podria desfer en valor absolut en comptes de guardar la velocitat mitjana. Tot i així, RRDtool segueix aplicant els mateixos esquemes encara que en cas de tractar-se de comptadors no monòtons desactiva el detector de desbordaments de l'equació 3.1 i permet que tinguin increments negatius, calculats segons l'equació següent:

$$\Delta_i = x_i - x_{i-1} \quad (3.3)$$

Aplicant l'equació 3.3 a les dades de la taula 3.7 s'obtenen els increments de temperatura per cada interval a la taula 3.8.

A la figura 3.13 es representa la normalització dels increments de temperatura segons el criteri de total vist anteriorment. En el cas de comptadors no té gaire efecte que siguin monòtons o no ja que d'ambdues maneres es guarda el total i la representació d'increments segueix sent útil. Però en el cas de les magnituds el total perd sentit ja que normalment es vol visualitzar els valors 'tal qual'. En canvi si es desen els increments, es perd la referència i no es poden recuperar els valors originals. És més, els increments de les variables com la temperatura, en total són zero ja que es manté oscil·lant al voltant d'uns valors de referència. Per extensió, la velocitat mitjana al límit també s'aproxima a zero i aquests són els valors que RRDtool emmagatzema. A la figura 3.14 es representa el gràfic que RRDtool mostra si es desa la temperatura com a comptador.

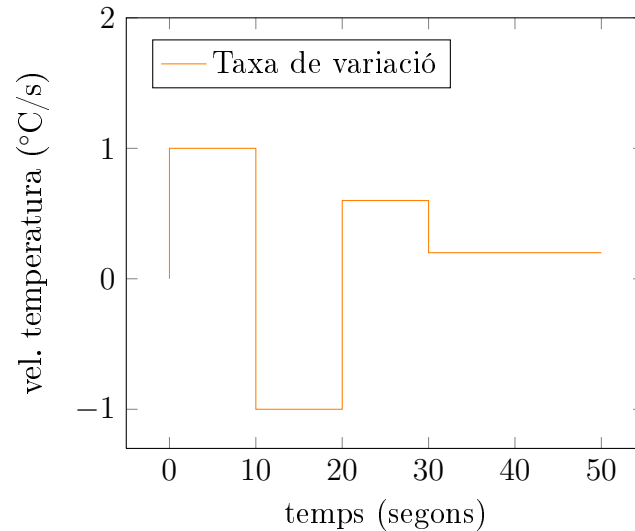


Figura 3.14: Normalització d'increments de temperatura

3.2.2 Emmagatzematge absolut

Així doncs, RRDtool no és capaç de recuperar la referència absoluta si s'emmagatzemen les magnituds com un comptador. La solució que es proposa per part de RRDtool és emmagatzemar les magnituds directament en el domini de velocitats dels comptadors. Aleshores els valors s'emmagatzemen 'tal qual' i s'aplica la normalització seguint l'esquema de RRDtool, de tal manera que els valors de temperatura ara es representen a l'eix de freqüència de l'histograma. Es pot veure un exemple a l'histograma de la figura 3.15 a on es normalitzen els valors de temperatura de la taula 3.7.

A la figura 3.16 es compara la temperatura emmagatzemada en relatiu com a comptador i en absolut com a velocitat de comptador. La primera variable, en ser un comptador, a RRDtool s'ha emmagatzemat com a velocitat i es representaria com s'ha fet anteriorment a la figura 3.14, però en el gràfic actual es representa correctament amb la referència per poder-la comparar en el mateix domini que els valors de temperatura. La segona variable, per RRDtool és tractada com una velocitat mitjana i es dibuixa tal com RRDtool la representaria.

L'emmagatzematge de les magnituds tant de la manera relativa com l'absoluta dona bons resultats d'aproximació. Ara bé, a RRDtool només es poden emmagatzemar com a absoluts ja que no conserva la referència en les relatives.

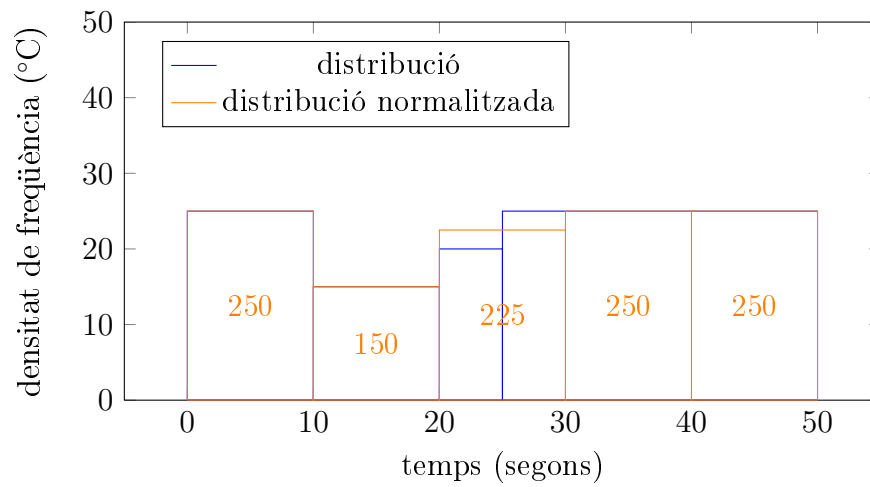


Figura 3.15: Histograma normalitzat per a temperatura

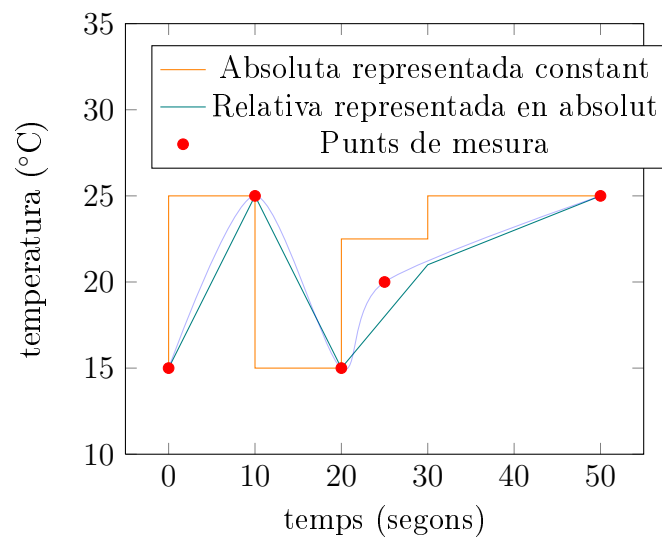


Figura 3.16: Comparació normalitzacions del valor de temperatura

3.3 Emmagatzematge a RRDtool

Com s'ha vist al llarg d'aquest capítol, RRDtool va començar sent una base de dades per a comptadors de tràfic de xarxa i per aquesta raó històrica encara avui dia està molt lligat a l'emmagatzematge de dades que tenen naturalesa de comptadors. A més a més, la representació de les dades també està molt lligada als comptadors. Per tant, representació i emmagatzematge estan massa acoblats per a poder comprendre i expandir RRDtool de manera flexible.

L'emmagatzematge dels comptadors es duu a terme en el domini de velocitat, és a dir que donats unes mesures d'un comptador es transformen a velocitat. Depenent de la manera com aquestes mesures es transformen a velocitat, RRDtool classifica els comptadors en quatre tipus:

- *Counter* és un comptador monòton que es mesura en valor absolut i es transforma segons les equacions 3.1 i 3.2, com el vist a la figura 3.1.
- *Absolute* és un comptador, monòton o no, que es mesura en valor relatiu i es transforma segons l'equació 3.2, com el monòton vist a la figura 3.3 i el no monòton vist a la figura 3.13.
- *Derive* és un comptador no monòton que es mesura en valor absolut i es transforma segons les equacions 3.3 i 3.2, com el vist a la figura 3.12.
- *Gauge* és un comptador, monòton o no, del que es mesura la velocitat mitjana en el darrer interval, com el vist a la figura 3.15. Els valors que es mesuren ja són velocitat i per tant no hi ha transformació.

L'equació 3.2 és l'encarregada de transformar els valors mesurats a velocitat mitjana. Com es pot veure, s'aplica als tres primers tipus i per tant es pot dir que qualsevol comptador acaba emmagatzemat amb el valor de la seva velocitat mitjana tal com passa directament en el darrer cas, els *gauge*. En resum, els *gauge* són el cas bàsic a RRDtool a causa de les raons històriques comentades al llarg d'aquest capítol.

A més a més, la representació que pot fer RRDtool dels comptadors està limitada en aquests valors emmagatzemats; és a dir que només pot mostrar gràfics de l'evolució de la velocitat mitjana, com s'ha vist a les figures 3.6 i 3.14 i en general com s'ha vist als histogrames a on la velocitat mitjana es correspon amb l'eix de densitat de freqüències. RRDtool no és capaç de fer representacions dels comptadors en valor absolut a partir de la velocitat mitjana que té emmagatzemada. Tot i així, per una millor comprensió del capítol, a les figures 3.8, 3.11 i 3.16 s'han representat els comptadors en valor absolut per poder comparar els valors originals del comptador amb els que s'emmagatzemen, ja que no és possible comparar en un mateix gràfic el domini comptador i el domini velocitat.

Finalment, s'ha explicat que RRDtool recomana desar les magnituds físiques com a tipus *gauge*. RRDtool només admet els quatre tipus de dades llistats en aquest

3 Magnituds i comptadors

apartat i els *gauge* són els que més s'avenen a les magnituds ja que emmagatzemen els valors 'tal qual'. En aquest capítol s'ha explicat com es podria desar una magnitud de manera relativa, com si fos un comptador, però que es perdia la referència absoluta i no es podia recuperar correctament. Aquesta pèrdua de referència també es pot produir en els comptadors, però en anomenar-los comptadors s'està suposant que el què interessa són justament els valors relatius; és a dir representar els increments als quals RRDtool es troba més còmode representar en velocitat mitjana.

4 L'SGBD RRDtool

En aquest capítol s'introdueix el sistema de gestió de bases de dades RRDtool, l'únic sistema operatiu conegut actualment que utilitza el model de dades Round Robin (RRD). També s'explica com funciona RRDtool des del punt de vista d'un usuari; per a més detall en el capítol següent s'explica el funcionament intern.

L'estructura d'aquest capítol ofereix incrementalment els conceptes bàsics de RRDtool. Primer s'introdueix el motiu de creació de RRDtool i les solucions que vol aportar.

En segon lloc, es descriu el model que utilitza RRDtool, el model RRD, utilitzant vocabulari habitual en altres bases de dades, el qual potser és més conegut pel lector. Com que RRDtool és una base de dades, no es pot deixar de banda una explicació breu del gestor de bases de dades, en aquest cas no es contempen els aspectes més informàtics i només es fa una equivalència entre vocabulari habitual de les bases de dades i implementació a RRDtool.

Tot seguit, s'explica el que caracteritza a RRDtool: l'estructura Round Robin. A causa d'aquesta estructura es necessita tractar les sèries temporals, el qual s'explica a través de la interpolació i la consolidació de les dades. Per tancar amb l'estructura, s'enumeren els tipus de dades que admet RRDtool.

Finalment, s'explica les operacions que es poden fer amb RRDtool i la sintaxi del llenguatge d'interacció amb la base de dades. Es destaca la curiosa implementació d'operacions per presentar gràficament les dades.

4.1 Introducció

RRDtool (Round Robin Database Tool), [24], és un SGBD per enregistrar i visualitzar dades estructurades com a sèries temporals i, actualment, és l'estàndard *de facto* en l'àmbit del programari lliure. L'objectiu principal de RRDtool és tractar sèries temporals, mantenint la coherència i la lògica que requereix el monitoratge de dades de la manera més ràpida i senzilla possible i ocupant el mínim d'espai.

Inicialment, formava part del programa de monitoratge *Multi Router Traffic Grapher*, (MRTG) [25], però posteriorment ha esdevingut una part separada amb característiques i funcionament independent, segons explica Oetiker a [22]. Durant aquest període de temps, RRDTool ha anat evolucionant fins a tenir una estructura dissenyada específicament per tractar dades en sèries temporals. Finalment,

aquest disseny propi ha estat classificat com un model de dades Round Robin (RRD, *Round Robin Database*).

El nom Round Robin prové principalment de la manera d'emmagatzemar les sèries temporals. Aquest emmagatzematge consisteix en desar cada valor en un arxiu que té un punter, el qual va recorrent els valors tot apuntant al que s'ha de llegir o escriure en cada moment. Això recorda als mètodes que recorren elements d'un grup de manera equitativa i ordenada, començant pel primer i tornant a començar quan s'arriba a l'últim, els quals s'anomenen mètodes *Round Robin*. A més a més, a RRDtool els arxius sempre tenen la mateixa mida d'elements; la mida es defineix en crear la base de dades i així s'eviten els problemes de creixement indeterminat que tenen altres bases de dades.

A banda de tenir un emmagatzematge Round Robin, RRDtool té més tècniques específiques per tractar sèries temporals, les quals han de tenir en compte el model Round Robin.

Una d'aquestes tècniques consisteix en conservar un sèrie temporal amb diferents resolucions de temps, a on cadascuna es pot anomenar 'finestra temporal'. Cada finestra conserva la sèrie temporal fins a un límit de temps, de manera que les dades més recents en el temps es desen amb més resolució però la perden a mesura que es tornen velles. Així es pot consultar la finestra temporal més adient en cada cas, considerant que cal detall en les finestres amb dades més recents però que es pot relaxar la resolució en les finestres que conserven intervals de temps més grans.

Aquesta solució d'emmagatzematge comporta una pèrdua de dades, ja que la informació perd resolució en el temps. Ara bé, aquesta pèrdua permet obtenir avantatges com per exemple un càlcul més àgil amb les dades, que la mida emmagatzemada no creixi en el temps o que la visualització de gràfics sigui més ràpida.

Cal destacar que la generació àgil de gràfics és un altre dels objectius principals de RRDtool, ja que es considera una part important en les interfícies persona-màquina dels sistemes de monitoratge. El benefici de RRDtool és no haver d'estar pendent del sistema de monitoratge ni del d'emmagatzematge i poder-se centrar en l'anàlisi visual de les dades, obtenint el major tractament i anàlisi automàtic possible per part de RRDtool.

Finalment, la implementació d'aquestes tècniques específiques ha comportat que RRDtool tingui un llenguatge propi per operar amb la base de dades. Així doncs, no implementa els llenguatges estàndards de bases de dades, com pot ser l'*Structured Query Language* (SQL, estàndard ISO/IEC 9075) utilitzat en les bases de dades relacionals. Això és degut a que les operacions a RRDtool només tenen sentit per sèries temporals i per exemple, amb referència als gràfics, hi ha una operació que serveix per representar gràficament.

4.2 Comparació amb altres SGBD

Estructuralment i comparant amb altres bases de dades, tenint com a referència bàsica a Date [8], una base de dades RRDtool és una taula que pot tenir més d'un camp però amb el nombre de registres fixos. El primer camp és la clau única i és on es desa la data i hora d'entrada de dades al registre. Els camps poden ser valors d'un mateix recurs o no tenir res a veure però sempre comparteixen el mateix temps de mesura, és a dir que s'han d'inserir tots els camps a la vegada.

Les operacions que permet RRDtool són les d'inserir i consultar dades i augmentar i disminuir els registres. Per la seva pròpia naturalesa no pot eliminar o afegir registres concrets sinó que les operacions d'augmentar i disminuir treballen amb els registres més vells. Tampoc pot canviar dades de registres existents i les dades s'han d'inserir seqüencialment en el temps.

Així doncs, per poder desar valors a RRDtool han de ser una seqüència de sèries temporals. És a dir, el valor s'ha de poder mesurar en temps successius i cada cert interval. A més, han de ser numèrics.

En general, les bases de dades optimitzades per manipular sèries temporals s'anomenen TSDS (*time series database sever*, [32]). Simplifiquen regles i operacions que en bases de dades relacionals són més complicades, com per exemple consultes de dades històriques. Les TSDS imposen un model en concordança amb les sèries temporals i són capaces d'implementar funcions específiques com ara l'organització de les dades, operacions bàsiques amb sèries, combinació de dades per formar sèries noves, filtratge, operacions estadístiques, etc.

La base de dades RRDtool és molt útil per conservar la lògica de temps real utilitzada en el mostreig de les dades (anomenat *system storage footprint*). A RRDtool no té sentit el model de dades relacional ni els enllaços de relació entre entitats (*E/R diagram*) ja que les dades només poden ser sèries temporals.

4.2.1 Gestor de RRDtool

Pel que fa a RRDtool, el sistema de gestió de la base de dades (SGBD), deixa bona part del nivell intern al sistema operatiu. Implementa la base de dades com un fitxer ordinari i per tant a nivell de sistema operatiu es gestionen els permisos d'usuari, si hi ha varies bases de dades RRDtool (més d'un fitxer), l'accés local o remot, redundàncies, etc.

En el nivell extern del SGBD, el d'usuari, hi ha el subllenguatge de dades (DSL) que està referit als objectes de la base de dades i a les operacions (*statements*). RRDtool té un DSL propi com es veu a l'apartat 4.4.

Aquest declaracions estan clarament separades en definicions dels objectes de la base de dades (*data definition language, DDL*) i processament dels objectes (*data*

manipulation language, DML). Això és degut principalment a que l'estructura interna de les RRDtool és fixa i queda configurada en una capçalera de la base de dades on entre altre coses hi ha el nombre de registres, els intervals de temps, el tipus de dades de cada camp (el tractament és diferent per a cada un, vegeu 4.3.2), etc.

Les operacions DDL permeten crear una base de dades RRDtool, augmentar o disminuir el nombre de registres, canviar la capçalera, exportar la base de dades a format XML (*Extensive Markup Language*), etc. Una d'aquestes operacions s'explica més endavant: *create*.

Les operacions DML permeten consultar i inserir valors, crear gràfics a partir de les dades, etc. Algunes d'aquestes operacions s'expliquen més endavant: *update*, *fetch*, *graph*.

4.3 Model Round Robin

El model de dades de RRDtool és Round Robin. Els registres es van recorrent en ordre per anar desant o llegint les dades. Es comença per un registre fins arribar a l'últim i llavors es torna a començar de nou pel primer. Vist de manera lògica, entre cada registre hi ha un interval de temps fix (*step*); és a dir, cada registre emmagatzema el valor que tenia un recurs en un punt temporal determinat.

Les diferències principals amb altres bases de dades són que RRDtool es crea a l'inici amb un nombre fix de registres i que té un punter que va recorrent els registres de manera cíclica cada cop que s'escriuen o llegeixen dades. D'aquesta manera la base de dades no creix en mida i per tant no necessita manteniment.

Aquest comportament té una representació gràfica semblant a la figura 4.1 on la base de dades organitza els registres en una circumferència i els va recorrent. Pel que fa al punt de vista de les dades, també s'assembla molt al model d'una finestra temporal o d'una cua on cada cop que entra una dada nova s'elimina una de vella.

Aquesta estructura Round Robin està repetida varies vegades amb diferents resolucions temporals (*steps*). Com més antigues són les dades menys resolució es necessita i com més properes en el temps es vol tenir més dades però per un temps total més petit. En una base de dades RRDtool hi ha diferents taules amb els diferents intervals, així l'usuari pot seleccionar la resolució que més li convingui i el càlcul és més ràpid. A més a més, aquesta estructura disminueix les dades emmagatzemades.

El concepte de finestres temporals de la figura 4.2 il·lustra molt bé aquest funcionament. En els temps més pròxims hi ha finestres temporals petites (de dos dies) i amb moltes dades (cada 5 minuts). Per temps més antics la finestra és més gran (1 mes) però hi ha menys dades (cada 5 dies).

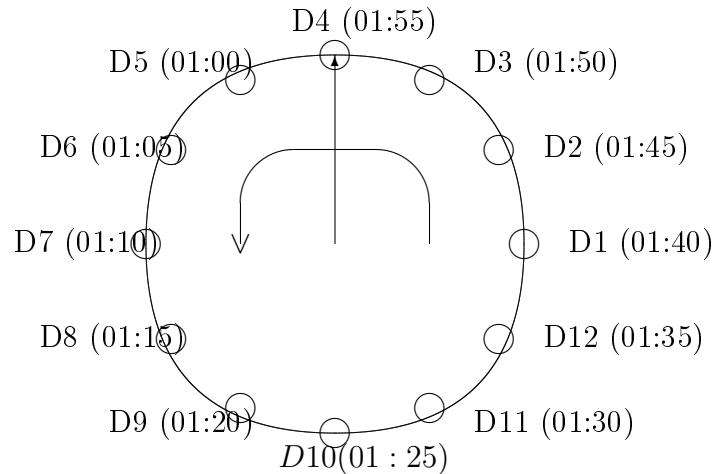


Figura 4.1: Representació d'una RRDtool amb una finestra de 12 registres

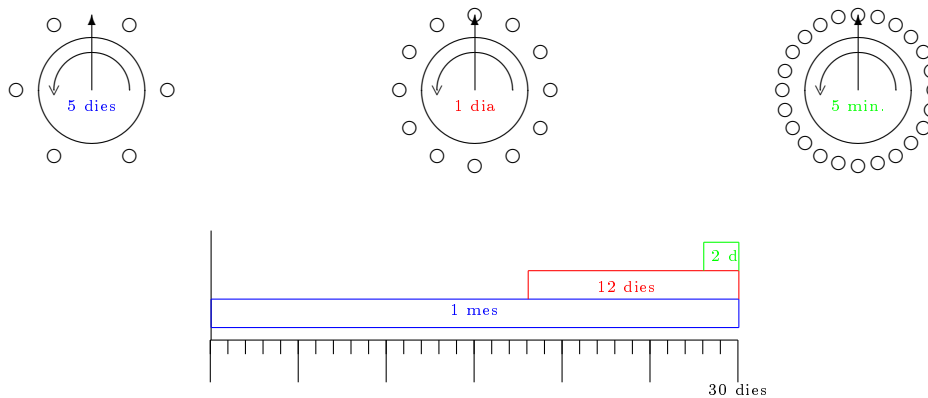


Figura 4.2: Representació d'una RRDtool amb tres finestres temporals

Aquesta estructura de diferents taules es crea al principi, així en una RRDtool sempre hi ha el mateix nombre de registres i no creix en mida. A l'inici cal donar un valor als registres que cauen fora del temps d'inici: se'ls dona el valor desconegut (*unknown*).

4.3.1 Interpolació i consolidació

RRDtool necessita dades en uns temps predefinitos. Si no obté un valor en aquest interval o n'obté un de no vàlid des d'un valor 'U' (*UNKNOWN*) o 'NaN' *Not A Number*.

A la pràctica és molt difícil de recollectar les dades en aquests intervals exactes que RRDtool obliga. Per tant, ha d'interpol·lar i consolidar (*data resampling*) els valors

per saber el que valen i desar-los en el temps prefixat. Sempre es manté coherència en les interpolacions respecte al que s'ha fet a l'interval anterior.

Posem per cas un comptador del qual es disposi d'una mesura cada cinc minuts. Si es fa una mesura en més temps, el comptador ha comptat més però RRDtool desa un valor interpolat al temps que li tocava. Els valors que sobren s'afegeixen al següent interval de temps, de fet tal com els correspon.

Per exemple, es mesura cada cinc minuts el nombre de visites que ha tingut un portal. En la primera mesura s'obté el valor 0. Llavors la següent mesura s'obté al cap de sis minuts amb un valor de 6 i RRDtool interpola als cinc minuts i desa el valor 5. Al cap de quatre minuts es mesura un valor nou de 7 i RRDtool desa el valor 8 perquè afegeix la unitat que havia sobrat en el temps anterior. A la taula 4.1 es compara la mesura del comptador amb els valors desats.

temps (minuts)	mesura (visites)	temps rrd (min.)	valor rrd (visites/min.)	comptatge rrd (visites)
0	0	0	0	0
6	6	5	1	5
10	7	10	1,6	8

Taula 4.1: Emmagatzematge d'un comptador a RRDtool

Així doncs es pot veure que el comptatge total desat ($0 + 5 + 8 = 13$) és el mateix que el mesurat ($0 + 6 + 7 = 13$) i per tant es pot dir que RRDtool sap conservar els comptatges. A RRDtool en comptes de desar el valor del comptatge es desa el valor de velocitat (*rate*) del comptador com es pot veure a la quarta columna de la taula 4.1; és a dir es desa l'increment de comptatge dividit pel temps de l'interval.

En el cas que la variable mesurada sigui una magnitud contínua, per exemple temperatura, RRDtool no està pensada per aquest tipus de dades però pot desar-les 'tal qual', i en cas que faci falta aproximar-les amb la mitjana dels valors mesurats. És a dir, la mesura indica el valor actual de temperatura (no compta la temperatura acumulada) i interpola, com si fos un comptador, en els temps prefixats.

Si es prenen les dades de l'exemple anterior de visites (les mesures de la taula 4.1) com si fossin de temperatura els resultats són diferents com es pot observar a la taula 4.2. Quan el temps de mesura correspon amb el temps RRDtool el valor que es desa és el mateix que el de la mesura, ja que és el valor actual de temperatura. En els cinc minuts que hi ha més d'una mesura, s'ha d'interpol·lar entre el valor 6 durant 1 segon i el valor 7 durant 4 segons: un valor resultant de 6,8.

Per altra banda, hi ha la tasca de la consolidació de valors que té lloc quan s'està calculant el valor per una finestra temporal de resolució més gran. És a dir, que es té un interval de temps amb varis valors i cal resumir-lo amb un valor.

temps (minuts)	mesura (°C)	temps rrd (min.)	valor rrd (°C)
0	0	0	0
6	6	5	6
10	7	10	6,8

Taula 4.2: Emmagatzematge de temperatura a RRDtool

A l'exemple de model Round Robin anterior de la figura 4.2 hi ha una finestra amb dades diàries i una finestra amb dades cada cinc dies. Quan es calcula un valor per la finestra de cinc dies es fa a partir de la mitjana (funció de consolidació) de 5 valors de la finestra diària.

A RRDtool es pot consolidar les dades amb diferents funcions: mitjana, mínim, màxim o últim valor. Aquestes funcions i altres aspectes d'interpolació i consolidació es veuen en més detall al capítol 5.

4.3.2 Tipus de dades

La interpolació que s'ha tractat de manera diferent en els exemples de les taules 4.1 i 4.2 indueixen a que RRDtool ha de diferenciar comptadors i magnituds. És a dir, que les dades de les sèries temporals monitorades són tractades de manera diferent segons la seva naturalesa. Així apareixen els següents tipus de dades que a RRDtool s'anomenen *data source types* (DST):

- counter
- derive
- absolute
- gauge
- compute

Un *counter* és un comptador típic que comença a zero i es va incrementant en relació al paràmetre que està mesurant. Si s'arriba al valor màxim del comptador llavors es torna a començar des de zero. En un moment determinat es pren el valor del comptador i un temps més tard es torna a consultar. Fent la diferència dels dos valors s'obté l'increment. Si a més es divideix per la diferència dels temps el resultat és la mitjana del valor per unitat de temps.

Un *derive* és un comptador però que també pot decrementar. És a dir poden aparèixer increments negatius.

Un *absolute* és un comptador però en aquest cas RRDtool considera que quan es llegeix, el comptador reinicia el seu valor. És a dir que la lectura del comptador dóna directament l'increment del paràmetre mesurat.

Un *gauge* és una mesura d'una magnituds, s'utilitza per a valors que s'han de desartal com s'han mesurat. Les magnituds són funcions contínues i es mesuren en un temps concret i, a diferència dels comptadors, la mesura indica el valor actual i no informa dels valors des de l'última mesura.

Un *compute* calcula els seus valors a partir d'altres tipus de RRDtool. En la terminologia de bases de dades s'anomenen dades virtuals o columnes calculades. Quan s'insereix un valor a una RRDtool no es dóna cap valor als *compute* però es calcula un valor segons una expressió RPN de manera similar als *cdef* (v. apartat 4.4.5).

4.4 Operacions

L'estructura d'una base de dades RRDtool és molt simple ja que està dissenyada només pel monitoratge. Així doncs les operacions també són molt simples. Ara bé, entendre el funcionament intern, sobretot pel que fa al mètode de consolidació, no és tant immediat. El funcionament intern es descriu al capítol 5, a continuació es descriu el funcionament a partir de les operacions.

Hi ha molta terminologia pròpia de RRDtool que s'anirà explicant al llarg d'aquesta secció, a continuació es llista la més important per tal de tenir-ne una primera referència:

DS Data Source - camp d'una RRD

DST Date Source Type - els tipus amb que RRD classifica les dades en sèries temporals (v. apartat 4.3.2)

RRA Round Robin Archive - taula d'una RRD, la qual té l'interval de temps i els registres definits

CF Consolidation Function - funció de consolidació a l'hora de treballar amb les dades

PDP Primary Data Point - valor que s'obté del monitoratge

CDP Consolidated Data Point - conjunt de PDP que passen a ser un valor desat a una taula de la base de dades

Les operacions bàsiques que ofereix RRDtool són les de crear una base de dades, inserir-hi valors i consultar-los. A més a més, el temps és un concepte molt important a RRDtool i treballa amb el format de *Unix time stamps*, a continuació s'explica en detall el treball amb el temps.

Per a més informació de les operacions de RRDtool es pot consultar el manual d'usuari a [24].

4.4.1 Temps: *Unix time stamps*

RRDtool desa dades en sèries temporals. Cada dada té associat un temps de mesura (*time stamp*) que s'expressa de la manera tradicional a Unix: amb *Unix time stamps*, [33]. En resum, el temps s'anota com els segons que han passat des de l'1 de gener de 1970 expressat en UTC (*Coordinated Universal Time*). Quan es treballa amb la base de dades els temps estan desats en aquest format però són senzills de convertir. Per exemple, mitjançant les ordres Unix següents:

- L'ordre `date +%s` retorna el *time stamp* de la data actual: `1274799600`.
- L'ordre `date -d @1274799600` retorna la data que correspon: `dt mai 25 17:00:00 CEST 2010`. A la resposta hi ha fus horari, localització geogràfica de l'hora, en aquest cas s'ha executat l'ordre `date` a la zona *CEST* que correspon a l'horari d'estiu de l'Hora Central Europea (UTC+2).
- L'ordre `date` té molts més paràmetres per consultar temps. Es poden combinar per consultar el *time stamp* d'una hora concreta, per exemple `date -\ d "20100101_00:00:00_UTC" +%s` respon amb: `1262304000`.
- Amb `date` també es tracten les fraccions de segons, la resolució més petita que s'admet és de nanosegons. Per exemple `date -d "20100101_00:00:00.1_UTC" +%s.%N` retorna `1262304000.100000000`. Hi ha formats estàndars de mostrar dades amb nanosegons com per exemple l'Internet RFC 3339, [18] que es pot utilitzar amb `date -d "20100101_00:00:00.1_UTC" --rfc-3339=\ ns` retorna `2010-01-01 01:00:00.100000000+01:00`, un format equivalent a `%Y-%m-%d %H:%M:%S.%N%:z`.

Els temps a RRDtool, per exemple els *stepsize*, tenen resolució de segons. RRDTool només permet utilitzar nanosegons en l'inserció de valors (*update*).

Resumint, els *Unix time stamps* per definició són respecte a UTC però la resta de formats horaris es mostren segons el fus horari. En tots els exemples següents es treballa amb UTC per simplificar la comprensió dels temps.

4.4.2 Creació: *create*

Crear una base de dades RRDtool és crear un fitxer que conté una capçalera amb la informació necessària i unes dades. Hhi ha operacions per exportar aquest fitxer altres formats, com per exemple a XML (*Extensive Markup Language*).

En el moment de creació d'una RRDtool l'estructura ha de quedar plenament definida ja que les operacions d'inserció i consulta no la poden modificar. Els principals camps que s'han de definir són: període de mostreig, temps de termini, variables a mesurar i taules amb diferents temps de mostreig. A continuació es detallen.

El primer que cal és definir el pas de mostreig (*step*). La RRDtool rep valors en temps imprevisibles. AL llarg d'un temps d'*step* forma un PDP (*Primary Data*

Point) que s'obté a partir de la interpolació de les mostres que s'han rebut. Com que normalment les mostres no es reben exactament en el temps d'*step*, es calcula interpolant per trobar el valor de PDP que li correspondria (vegeu 4.3.1).

El temps de termini per rebre una mostra el defineix el *heartbeat*. Si supera aquest termini i no s'ha pogut calcular el PDP perquè no hi ha mostres suficients, el PDP es considera desconegut (*unknown*).

La manera més senzilla de definir l'*step*, en unitats de segons, és que coincideixi amb la freqüència que arriben les dades i que el *heartbeat* sigui una mica més gran. D'aquesta manera hi haurà una mostra per PDP. Aquests temps no cal que siguin els mateixos, els seus valors permeten configurar escenaris diferents. Per exemple per tenir més d'una mostra per cada PDP, el *heartbeat* ha de ser semblant al temps de monitoratge i l'*step* tantes vegades més gran com mostres es vulguin.

El segon que es defineix són els camps, identificats com *Data Source* (DS), que equivalen a les variables que s'estant monitorant. Per cada variable es defineix de quin tipus és (DST, v. apartat 4.3.2), el *heartbeat* i el rang de valors que pot prendre. Si el valor mesurat està fora dels límits es desa un PDP *unknown*, en aquest cas un límit definit com a 'U' indica que no existeix límit.

En tercer lloc es defineix l'estructura de les taules on cada una representa una finestra temporal diferent. Cada taula s'anomena *Round Robin Archive* (RRA) i en cada una es defineix el nombre de registres que té (*rows*) i l'interval de temps que hi ha entre ells (*step*, cal no confondre'l amb l'*step* dels PDP). Aquest interval de temps indica el nombre de PDP que calen per omplir un registre, o fila, d'una taula RRA. Els valors que omplen els registres s'anomenen CDP (*consolidated data point*).

Així bàsicament, a cada taula es defineix les finestres temporals indicant quantes files té i quants PDP li calen per formar un CDP. També es defineix el percentatge (*xff*) de PDP coneguts que calen perquè el CDP es consideri vàlid.

Per formar un CDP s'utilitza una funció de consolidació (CF) als PDP que pot ser *average*, *minimum* (o *min*), *maximum* (o *max*) o *last*. Això encara fa augmentar més el nombre de taules possibles ja que cada funció de consolidació es defineix en una RRA diferent. És a dir, es pot tenir les mateixes finestres temporals però que amb una els CDP es calculin a partir de la mitjana dels PDP i en una altra en la qual el CDP sigui el màxim dels PDP.

Finalment, es decideix el temps exacte d'inici de la base de dades ja que actua com a origen pels intervals de temps dels registres. L'inici es defineix en *Unix time stamp* (vegeu 4.4.1) o també es pot indicar el punt temporal actual amb 'N'. No és un paràmetre molt important perquè el monitoratge es basa en temps relatiu i no cal quadrar la base de dades amb el pensament humà de calendari; el mateix es pot dir per les RRA. Per exemple si es defineixen CDP cada 30 dies, la base de dades no sempre es correspondrà amb els mesos de calendari, però no té més importància ja que el que realment ens interessa és assegurar els períodes de 30 dies.

Un exemple d'ordre de creació d'una base de dades, `temperatura.rrd`, que s'inicia a "20100101 00:00:00 UTC" és:

```
rrdtool create temperatura.rrd --start 1262304000 --step 300 \
    DS:temp1:GAUGE:600:-U:U \
    RRA:AVERAGE:0.5:1:24 \
    RRA:AVERAGE:0.5:12:24
```

Els DS tenen el format `DS:variable_name:DST:heartbeat:min:max` Les RRA tenen el format `RRA:CF:xff:step:rows`

Per exemple, `RRA:AVERAGE:0.5:12:24` fa la mitjana de cada 12 PDP que rebí i desarà les últimes 24 mitjanes calculades i en canvi. Considerant l'*step* de 300 segons, s'obté un CDP cada hora (12×300 s) i en total es desen 24 hores ($24 \times 12 \times 300$ s). En canvi, `RRA:AVERAGE:0.5:1:24` fa la mitjana de cada 1 PDP que rebí i en desa els últims 24 calculats

Es mostra el contingut de la base de dades quan s'acaba de crear, per exemple amb l'ordre `dump` que permet escriure per pantalla el contingut, `rrdtool dump \ temperatura.rrd`. S'observen els *Round Robin Archives*:

- Hi ha emmagatzemats 24 valors per la resolució de 5 minuts, l'equivalent a un *step* de 300 segons:

```
<!-- 2009-12-31 22:05:00 UTC / 1262297100 --> <row><v>NaN</v></row>
->v></row>
<!-- 2009-12-31 22:10:00 UTC / 1262297400 --> <row><v>NaN</v></row>
->v></row>
...
<!-- 2009-12-31 23:55:00 UTC / 1262303700 --> <row><v>NaN</v></row>
->v></row>
<!-- 2010-01-01 00:00:00 UTC / 1262304000 --> <row><v>NaN</v></row>
->v></row>
```

- Hi ha emmagatzemats 24 valors per la resolució de 60 minuts, l'equivalent a 12 *steps* de 300 segons:

```
<!-- 2009-12-31 01:00:00 UTC / 1262221200 --> <row><v>NaN</v></row>
->v></row>
<!-- 2009-12-31 02:00:00 UTC / 1262224800 --> <row><v>NaN</v></row>
->v></row>
....
<!-- 2009-12-31 23:00:00 UTC / 1262300400 --> <row><v>NaN</v></row>
->v></row>
<!-- 2010-01-01 00:00:00 UTC / 1262304000 --> <row><v>NaN</v></row>
->v></row>
```

Els valors mostren NaN (Not a Number) perquè encara no hi ha cap valor emmagatzemat.

Per altra banda, cal notar que els temps són anteriors a la data d'inici: dues hores abans per la resolució de 5 minuts i un dia abans per la resolució de 60 minuts. Això és degut a que RRDtool sempre treballa mirant cap al passat, mai té res guardat que correspongui al futur. D'aquesta forma el primer valor emmagatzemat ocupa la primera posició de la taula i es va desplaçant.

4.4.3 Inserció: *update*

Per desar dades noves a la RRDtool cal fer un *update* del valor de la dada juntament amb el punt temporal que s'han mostrejat. Aquests valors d'entrada seran considerats com a mostres i cada *step* es consolidaran en un PDP. L'únic requisit, indispensable per l'estructura de RRDtool, és que les insercions s'han de fer en ordre temporal. Un cop s'ha entrat un valor amb un temps no es poden entrar valors amb temps anteriors ni tampoc modificar valors anteriors.

Per especificar el temps s'utilitzen *Unix time stamp* (v. apartat 4.4.1) o una 'N' que significa "ara". Una ordre d'inserció a la temperatura.rrd de l'exemple anterior:

```
rrdtool update temperatura.rrd 1262304300:5:15 \  
1262304600:7:30 1262304900:3:2
```

Amb aquesta ordre s'indica que a la 01:05:00 la variable temperatura valia 5°C i s'havien enviat 15 missatges, a les 01:10:00 era de 7 i 30 i a les 01:15:00 prenen 3 i 2 per valor. Com s'observa, es pot actualitzar la base de dades amb un valor o amb més d'un a la vegada.

4.4.4 Consulta: *fetch*

Per extreure dades s'utilitza *fetch*.

S'ha d'especificar l'interval que es vol mostrar definit des del temps d'inici fins al temps final, els quals prenen per valor un *At-Style Time Specification*¹.

Per la consulta també s'ha d'especificar en quina funció de consolidació CF (*average, min, max, last*) es volen llegir les dades. Com s'ha vist a l'apartat 4.4.2, cada CF és una taula diferent.

L'ordre de consulta retorna una resposta amb els valors sol·licitats. Si retorna el valor 'NaN' (*Not A Number*) indica que el valor encara no ha estat introduït o que no s'ha pogut calcular. A vegades també es mostra amb 'U' (*unknown*).

¹La definició d'*At-Style Time Specification* es pot trobar a la documentació de l'ordre *fetch* <http://oss.oetiker.ch/rrdtool/doc/rrdfetch.en.html>

Seguint amb l'exemple, consultem dades a la temperatura.rrd:

```
rrdtool fetch temperatura.rrd AVERAGE \
  --start 1262304300 --end 1262305200
```

Retorna la sortida següent:

```

          temperatura missatges
1262304300: 5 15
1262304600: 7 30
1262304900: 3 2
1262305200: nan nan

```

En el resultat, a la primera fila hi ha els noms de les variables que conté la base de dades. A la primera columna hi ha el temps en *Unix time stamp*, i a les columnes següents hi ha els valors CDP (ja interpolats), a l'exemple la segona columna és de temperatura i la tercera d'increments de missatges.

4.4.5 Gràfics: *graph*

RRDtool permet crear un gràfic de les dades desades en una o múltiples bases de dades. El gràfic és una imatge que pot estar en format PNG, EPS, SVG o PDF, i que també permet extreure les dades per pantalla.

L'ordre que permet generar un gràfic és molt semblant a la de consulta, internament utilitza la funció de consulta, a més a més permet fer càlculs extres amb les dades. Els diferents tipus de definició que es poden trobar en un gràfic i que s'expliquen en els apartats següents:

DEF Data definition

CDEF Data calculation

VDEF Variable definition

GPRINT i **PRINT** treuen dades i informació en forma de text

Aquests quatre són els elements bàsics que té RRDtool per operar amb les sèries temporals. A partir dels DEF que s'utilitzen per referenciar una sèrie temporal d'una base de dades, s'opera als CDEF i VDEF mitjançant expressions RPN i s'acaba mostrant els resultats amb els GPRINT. Aquestes operacions es detallen més endavant en aquesta secció.

També hi ha altres elements gràfics de presentació secundaris (AREA, LINE1, LINE2, LINE3, STACK, HRULE, VRULE, COMMENT, TICK, SHIFT, TEXTALIGN), dels quals només es detallarà els que s'utilitzen per pintar les dades. Es pot trobar més informació al manual d'usuari a [23].

Per exemple, un gràfic que mostra la temperatura:

```
rrdtool graph imatge.png \
  --start 1262304300 --end 1262304900 \
  DEF:temperatura=temperatura.rrd:temperatura:AVERAGE \
  LINE2:temperatura#FF0000
```

A l'exemple, es crea un gràfic *imatge.png* que comença a les 01:05 i acaba a les 01:15. Es defineix que es vol llegir la variable *temperatura* i que la pinti en una línia de 2 píxels. Al final del capítol a la figura 4.3 es mostra un gràfic d'exemple, un cop explicats amb més detall tots els elements que hi intervenen.

Definició de dada: **DEF**

La definició de dades (DEF) és l'encarregada d'obtenir una variable de RRDtool i anomenar-la (amb un *vname*) per poder-la usar a les altres definicions.

Cada declaració DEF només recull valors d'una variable però en una ordre *graph* hi pot haver més d'un DEF i cada un pot recollir variables de bases de dades diferents. Cada una d'aquestes variables conté un conjunt de valors; és a dir conté una sèrie temporal.

La definició DEF és molt semblant a l'operació de consulta i té el format:

```
DEF:vname=rrdfile:ds-name:CF \
  [:step=step] [:start=time] [:end=time]
```

Càlcul amb les dades: **CDEF**

La funció principal dels CDEF és fer càlculs matemàtics amb els valors emmagatzemats a la RRDtool, abans de mostrar-los gràficament, i retorna un conjunt nou de valors. Una operació CDEF sempre ha de contenir com a mínim un DEF o un altre CDEF associat.

Les expressions matemàtiques s'escriuen en RPN (Reverse Polish Notation), molt típic en calculadores científiques HP [10]. La característica bàsica és que s'opera en una pila. Per les instruccions CDEF, la pila es processa per cada valor; és per això que es retorna un conjunt de valors de la mateixa mida que el d'entrada. A continuació es mostra el format del CDEF aplicat a un exemple que converteix velocitat en unitats de km/s a m/s:

```
CDEF:vname=RPN expression
CDEF:vname=DEFoCDEF,1000,*
"CDEF:velocitatMS=velocitatKMS,1000,*"
```

Definició de variable: VDEF

La definició de variable (VDEF) calcula un valor a partir d'un DEF o un CDEF. Retorna un sol valor o un temps dependent de la funció RPN que s'utilitza. Quan s'utilitza el valor del VDEF en una altra expressió, actua com a un valor numèric.

Les expressions matemàtiques s'escriuen en RPN com en el cas CDEF. Ara bé, les instruccions VDEF treballen amb tot el conjunt de dades de cop i per això només retornen un valor; de manera que es pot dir que consoliden les dades. Per a les instruccions RPN de VDEF hi ha una llista limitada de funcions, com per exemple *maximum*, *minimum*, *average*, *total*, *percent*, etc. A continuació es mostra el format del VDEF aplicat a un exemple que calcula la mitjana d'unes dades:

```
VDEF:vname=RPN expression
VDEF:mitjana=dades ,AVERAGE
```

Presentació de text: Gprint

GPRINT pels gràfics i PRINT per la sortida per pantalla, permeten escriure valors de text. Les operacions GPRINT només accepten VDEF. A continuació només es detalla un ús bàsic, es pot trobar al manual, [23], altres casos d'ús.

El format de visualització s'introdueix precedit del signe '%'. Els signes '#' són enters que defineixen l'amplada del camp i la precisió decimal. Alguns exemples de format són els següents: '%#.##le' mostra el número amb format científic 5,4527e+03 o bé '%#.##lf' mostra el número amb decimals 0,0045. Si es mostra un valor VDEF llavors també es pot escriure el temps associat, per fer-ho cal definir el format de temps a *strftime*. A continuació es mostra el format de PRINT aplicat a un exemple que escriu la mitjana calculada amb un VDEF:

```
PRINT:vname:format[:strftime]
PRINT:mitjana:%2.2lf
```

El resultat s'escriu per pantalla:

```
print[0] = "7.500000"
```

Presentació gràfica: Area, Line i Stack

Area, *line* i *stack* són les instruccions encarregades de pintar els valors obtinguts en aplicar DEF, CDEF o VDEF. LINE dibuixa els valors units per una línia i AREA els dibuixa pintant la superfície entre la línia i l'eix X. Si s'utilitza STACK llavors el dibuix es fa acumulant el valor a l'element anterior.

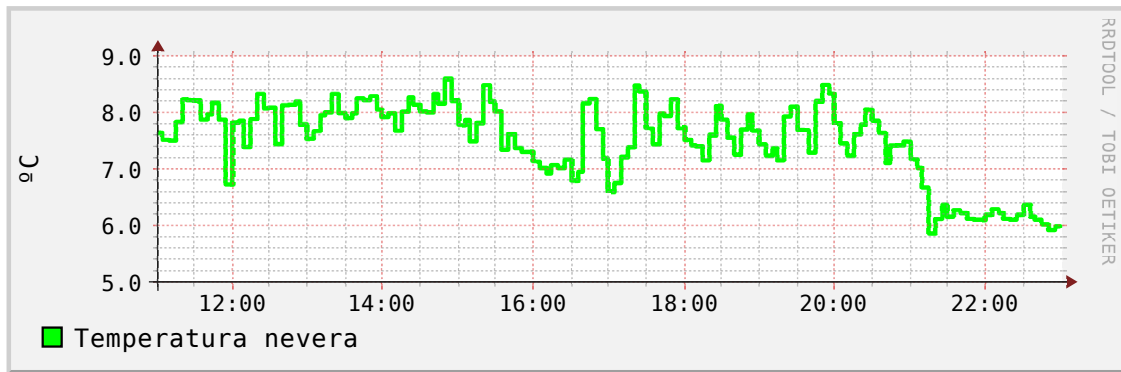


Figura 4.3: Gràfic d'exemple de mesura de temperatura

L'exemple següent mostra com dibuixar una línia d'espessor mitjana al cas de la temperatura. La gràfica obtinguda es visualitza a la figura 4.3:

```
LINE[width]:value[#color][:[legend][:STACK]] ...
LINE2:temperatura#00FF00:"Temperatura nevera"
```

5 Funcionament intern de RRDtool

En aquest capítol s'explica detalladament el funcionament intern de RRDtool des del punt de vista d'una dada; és a dir el recorregut que fa la dada des de que entra fins que queda desada.

Com ja s'ha vist al capítol 4, RRDtool és una base de dades que implementa el model Round Robin (RRD). A partir de l'estudi intern de RRDtool es voldria entendre com és aquest model RRD, tot i així RRDtool conté massa detalls d'implementació i no és possible arribar a la comprensió profunda del model RRD.

L'objectiu d'aquest capítol és recollir les característiques de RRDtool per més endavant extreure la informació rellevant i dissenyar un model de RRD. Amb aquest model, detallat al capítol 6, les bases de dades RRD s'entenen més fàcilment.

L'estructura d'aquest capítol està condicionada pel recorregut que segueixen les dades a RRDtool, el qual es pot entendre en tres etapes, [3, 24]:

1. transformació a velocitat (*transform to a rate*),
2. normalització de l'interval (*normalize the interval*) i
3. consolidació d'interval (*consolidate intervals*).

Durant la lectura del capítol cal tenir present dues vessants de RRDtool. La primera és que només treballa amb sèries temporals en passat; és a dir dades mostrejades cada un cert interval de temps a on el valor indica el què ha passat. La segona és que RRDtool emmagatzema les dades amb dues restriccions: com a ràtio magnitud per segon (velocitat, *rate*) i en intervals prefixats; precisament a través de les tres etapes anteriors es modifica les dades complint aquest esquema.

En la primera etapa es transforma la dada a velocitat, en el cas que no ho sigui. Un cop ja ho és, pot passar a la segona etapa a on s'assegura que les dades s'hagin mostrejat amb temps regular i en cas de no ser així s'interpolen les dades per aconseguir-ho. Un cop les dades són regulars, a la tercera etapa es desen a la base de dades però amb diferents temps de mostreig i diferents funcions d'interpolació, la qual cosa s'anomena consolidació d'interval en altres de menys resolució.

5.1 Transformació a velocitat

RRDtool, internament, només sap manipular variables que siguin velocitats, enteses com s'ha explicat al capítol 3. Per tant, cal assegurar que les dades d'entrada són velocitats o transformar-les en cas que no ho siguin.

Per tal de complir amb aquesta restricció de velocitat, RRDtool només permet l'entrada de dos tipus de dades: velocitats i comptadors. A més a més, distingeix entre tres tipus diferents de comptadors; així doncs, es classifiquen les dades de quatre maneres segons com es transformen a velocitat:

- Gauge
- Counter
- Absolute
- Derive

Particularment, en el cas del primer no hi ha transformacions perquè ja es consideren velocitats. En aquest cas, a RRDtool, també s'hi inclouen les magnituds que no es poden classificar com a comptadors, tot i que de manera aproximada com es veu a l'apartat 5.1.1.

Com que en els *gauge* no hi ha la transformació a velocitat, són un bon punt de partida per entendre com es desen les sèries temporals en velocitat a RRDtool.

5.1.1 Velocitat o magnitud: *Gauge*

Les dades de tipus Gauge són de tipus magnitud. Es considera que aquestes dades ja són velocitats i per tant no s'aplica cap transformació.

Per exemple, és el cas d'un tacòmetre. Aquest aparell mesura velocitats angulars directament en rad/s (ràtio d'una magnitud per segon) i per tant no cal fer transformacions d'aquestes dades.

Però no totes les magnituds són velocitats, per exemple la temperatura. RRDtool també classifica aquestes dades com a tipus Gauge ja que així es desen 'tal qual'. Com a conseqüència, les operacions que es defineixen en etapes posteriors no tenen sentit físic per aquestes magnituds; tot i que per magnituds són aproximacions vàlides segurament no són les millors aproximacions que s'hi poden fer. Alguns exemples de variables magnitud són la temperatura, el nivell d'un dipòsit, l'espai de disc lliure, memòria utilitzada, etc.

En la tradició de sèries temporals, com per exemple a [1], se suposa que la variable descrita és contínua i que els valors mostrejats representen el valor instantani del senyal. Aquestes sèries temporals es representen i s'interpreten amb interpolacions dels valors.

Però a RRDtool les sèries temporals s'interpreten en passat; és a dir se suposa que els valors mostrejats representen el valor del senyal en tot l'interval de temps que comprèn el temps actual fins al temps del darrer mostreig. Així doncs, es representen i s'interpreten amb intervals constants dels valors.

Aquest concepte de passat pren més sentit en els comptadors. En les magnituds s'ha d'entendre que el valor mostreat ofereix informació de la velocitat de la variable en tot l'interval anterior. En l'exemple següent es pot observar aquest comportament en passat. En els valors del gràfic 5.1, es mesura en el temps 20 una velocitat de 1m/s i RRDtool entén que entre el temps 10 i el temps 20 s'ha anat a una velocitat de 1m/s.

Exemple Es mesura cada 10 segons la velocitat d'un mòbil que parteix del repòs, té una acceleració de $0,1 \text{ m/s}^2$ i s'acaba parant, com es veu a la figura 5.1.

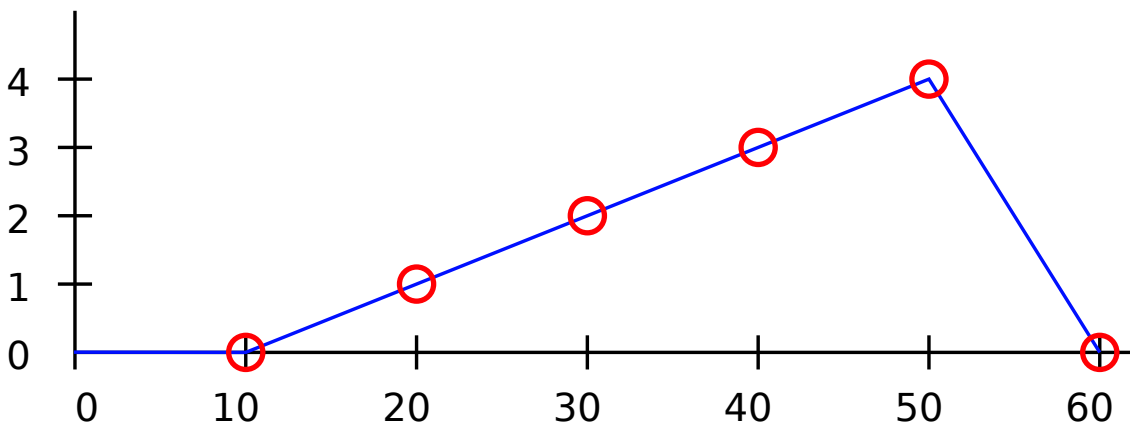


Figura 5.1: Moviment d'un mòbil: perfil de velocitat en blau, punts de mostreig en vermell

En primer lloc, es crea una base de dades de nom *velocitat.rrd* amb un temps de mostreig de 10 segons i un registre anomenat *mps* que emmagatzema la velocitat del mòbil.

```
export TZ=UTC
rrdtool create velocitat.rrd --start 1262304000 --step 10 \
    DS:mps:GAUGE:600:-U:U \
    RRA:AVERAGE:0.5:1:24
```

En segon lloc, s'actualitza la base de dades amb els valors $x = [0, 1, 2, 3, 4, 0]$ que s'han mesurat als temps $t = [10, 20, 30, 40, 50, 60]$.

```
rrdtool update velocitat.rrd 1262304010:0 1262304020:1 1262304030:2 \
    1262304040:3 1262304050:4 1262304060:0
```

Si es mostra el contingut de la base de dades, amb `rrdtool dump`, es pot veure com els valors han entrat correctament:

```

00:00:00 UTC --> <row><v>NaN</v></row>
00:00:10 UTC --> <row><v>0.0000000000e+00</v></row>
00:00:20 UTC --> <row><v>1.0000000000e+00</v></row>
00:00:30 UTC --> <row><v>2.0000000000e+00</v></row>
00:00:40 UTC --> <row><v>3.0000000000e+00</v></row>
00:00:50 UTC --> <row><v>4.0000000000e+00</v></row>
00:01:00 UTC --> <row><v>0.0000000000e+00</v></row>

```

Finalment, es demana a RRDtool que faci un gràfic dels valors anteriors, el resultat es pot veure a la figura 5.2:

```

rrdtool graph velocitat.eps -a EPS --start 1262303999 --end \
1262304060 DEF:vel=velocitat.rrd:mps:AVERAGE LINE1:vel#0000FF:"\
velocitat en m/s\l" -v "m/s" --x-grid SECOND:10:SECOND:10:SECOND\
:10:0:%X

```

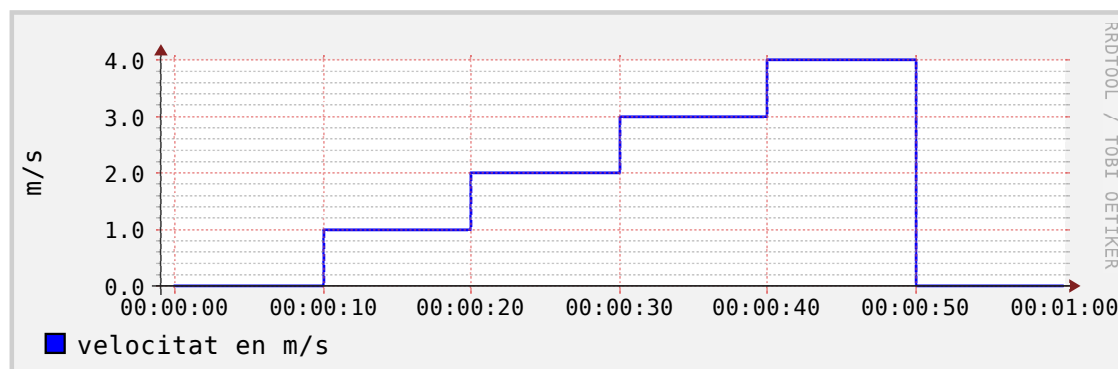


Figura 5.2: Velocitats emmagatzemades a RRDtool d'un tipus Gauge

D'aquest gràfic es desprèn com RRDtool interpreta les sèries temporals en sentit de passat. Els valors desats es representen constants en l'interval anterior al temps que els correspon. En l'exemple, a l'interval de temps $(0,10]$ hi ha el valor 0, en l'interval $(10,20]$ hi ha el valor 2, etc. En el temps 0, el valor és desconegut ja que no hi ha informació per l'interval anterior a zero.

Concloent, aquest és un exemple mostrejat de manera poc adequada segons la idea de RRDtool. El mòbil de la figura 5.1 té un perfil de velocitats mostrejat amb el valor instantani i aquest correspondria al mateix cas que mesurar una magnitud com la temperatura. Per mostrejar amb la idea de velocitat de RRDtool caldria que la velocitat fos constant; de fet, aleshores el perfil de velocitats original seria el de la figura 5.2.

Si la velocitat no és constant, com és el cas de l'exemple, aleshores per complir amb la idea de RRDtool el valor mostrejat hauria d'informar de la velocitat en el darrer interval. Per exemple, amb la velocitat mitjana de cada interval ($x =$

[0, 0,5, 1,5, 2,5, 3,5, 2]) ja s'aconsegueix una millor aproximació a la idea de RRDtool. Encara és més, en el concepte de RRDtool aquest exemple s'hauria de passar al domini de comptadors; és a dir mesurar l'espai recorregut en comptes de la velocitat. En els apartats següents es tractarà amb més detall els comptadors.

5.1.2 Comptador monòton: *Counter*

Les dades de tipus Counter són dades provinents de comptadors. Aquestes dades no són velocitats i per tant cal aplicar transformació.

Un comptador és un aparell de mesura que comença a zero i es va incrementant de manera contínua i monòtona en relació al paràmetre que està mesurant. A diferència d'una magnitud, un comptador no només indica el valor instantani de la variable sinó que informa dels valors des de l'última mesura. En anglès hi ha dues paraules que es tradueixen com a comptador: *counter*, utilitzat en general, i *meter*, utilitzat per exemple en energia elèctrica, en gas, en odometria, etc.

Per a transformar els valors d'un comptador a velocitat, a RRDtool se segueix el procediment següent. En un moment determinat es pren el valor del comptador i al cap d'un temps definit es torna a consultar. Fent la diferència dels dos valors s'obté l'increment. Si a més es divideix per la diferència dels temps, el resultat és la mitjana del valor per unitat de temps; és a dir, una velocitat. A RRDtool el temps es mesura en segons, per tant les velocitats sempre es desen amb unitats de *magnitud/s*.

$$c_2 \geq c_1 \longrightarrow \text{mitjana per } u.t. = \frac{c_2 - c_1}{t_2 - t_1} \quad (5.1)$$

on c_2 i t_2 són el valor del comptador i el temps actual i c_1 i t_1 són en el temps anterior.

RRDtool només emmagatzema velocitats. Per tant en les dades tipus comptador sempre calcula aquesta proporció mitjana del valor en cada període. Per exemple si el comptador mesura quilòmetres (km), el valor desat són velocitats (km/s). Per aquesta mateixa raó, el primer valor que es desa a la base de dades no es pot calcular, ja que l'anterior no existeix, i es considera desconegut.

Els comptadors no poden decreïxer però tenen un valor màxim que quan hi arriben tornen a començar des de zero. És a dir, els comptadors es poden desbordar (*counter wraps*). RRDtool té en compte aquest cas i calcula l'increment real que s'ha produït.

$$c_2 < c_1 \longrightarrow \text{mitjana per } u.t. = \frac{(\text{fons d'escala} - c_1) + c_2}{t_2 - t_1} \quad (5.2)$$

on ara, a diferència de l'equació 5.1, cal tenir en compte el fons d'escala del comptador.

Ara bé, de moment RRDtool només reconeix comptadors de 32 o 64 bits, els quals tenen rangs hexadecimals de 0 a FFFFFFFF_h i de 0 a FFFFFFFFFFFFFFFF_h

respectivament. Si el rang del comptador és diferent d'aquests dos, el valor calculat és erroni ¹.

Alguns exemples de comptador són el comptaquilòmetres d'un cotxe, els bytes transferits d'un router, la quantitat de pàgines impreses, el nombre de visitants, etc.

Exemple Ara en el mòbil anterior de la figura 5.2, en comptes de mesurar la velocitat es mesura l'espai recorregut com es veu a la figura 5.3, també cada 10 segons.

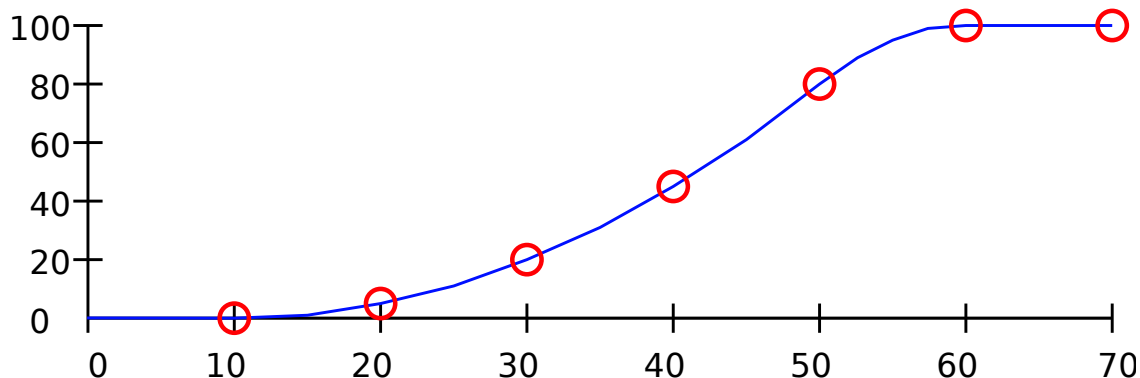


Figura 5.3: Moviment d'un mòbil: perfil d'espai recorregut en blau, punts de mesura en vermell

En primer lloc, es crea una base de dades de nom *comptador.rrd* amb un temps de mostreig de 10 segons i un registre anomenat *mps* que emmagatzema la velocitat del mòbil però les dades d'entrada són l'espai recorregut.

```
export TZ=UTC
rrdtool create comptador.rrd --start 1262304000 --step 10 \
    DS:mps:COUNTER:600:-U:U \
    RRA:AVERAGE:0.5:1:24
```

En segon lloc, s'actualitza la base de dades amb els valors $x = [0, 5, 20, 45, 80, 100, 100]$ que s'han mesurat als temps $t = [10, 20, 30, 40, 50, 60, 70]$

```
rrdtool update comptador.rrd 1262304010:0 1262304020:5 \
1262304030:20 1262304040:45 1262304050:80 1262304060:100 \
1262304070:100
```

¹En comptes del tipus *counter* es pot utilitzar el tipus *derive* amb el límit de velocitat mínima a zero. Llavors no es calculen valors erronis en els desbordaments sinó que es consideren desconeguts. Sigui quina sigui la manera, RRDtool no sap tractar aquests comptadors de manera adequada.

Si es mostra el contingut de la base de dades es pot veure com no hi ha els mateixos valors que s'han inserit; el gràfic de la figura 5.4 que ha generat RRDtool també corrobora aquestes dades:

```
00:00:00 UTC --> <row><v>NaN</v></row>
00:00:10 UTC --> <row><v>NaN</v></row>
00:00:20 UTC --> <row><v>5.0000000000e-01</v></row>
00:00:30 UTC --> <row><v>1.5000000000e+00</v></row>
00:00:40 UTC --> <row><v>2.5000000000e+00</v></row>
00:00:50 UTC --> <row><v>3.5000000000e+00</v></row>
00:01:00 UTC --> <row><v>2.0000000000e+00</v></row>
00:01:10 UTC --> <row><v>0.0000000000e+00</v></row>
```

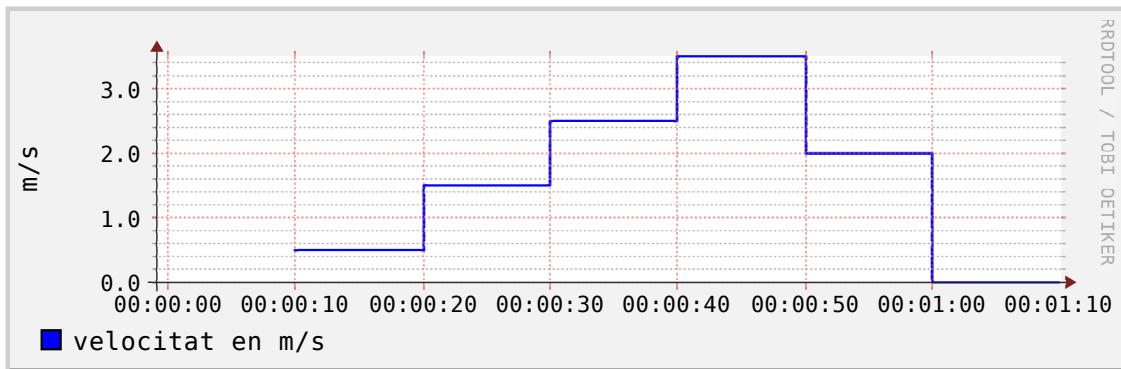


Figura 5.4: Velocitats emmagatzemades a RRDtool d'un tipus Counter

En aquest gràfic, on les unitats són de metres per segon, es veu l'efecte de desar les dades en velocitat. Com que l'entrada de dades és de tipus comptador, RRDtool s'encarrega de transformar els valor a velocitat segons les equacions 5.1 i 5.2 i també es compleix que el primer valor inserit s'emmagatzema com a desconegut ja que no es pot calcular.

En referència a l'espai recorregut, es pot calcular multiplicant cada valor emmagatzemat pel temps de mostreig (10 segons). Aleshores s'obté $[u, 5, 15, 25, 35, 20, 0]$ a on cada valor és l'increment d'espai recorregut en cada interval de temps corresponent; és a dir en els intervals $[(0, 10], (10, 20], \dots, (60, 70]$. Per calcular l'espai recorregut total entre el temps 10 i el temps 70 s'ha de sumar aquests increments, aleshores s'obté el comptatge total de 100 metres que es correspon correctament amb el de la figura 5.3.

5.1.3 Comptador doble: *Derive*

Les dades de tipus derive són de tipus comptador que tant pot incrementar com decrementar el valor; per això l'anomenem comptador doble. Aquestes dades no són velocitats i per tant cal aplicar transformació.

La transformació a velocitat es calcula amb l'equació 5.1, de la mateixa manera que els comptadors de tipus *counter*. Però en els *derive* no es té en compte el desbordament, sinó que un decrement del comptador es transforma a una velocitat negativa:

$$\text{mitjana per } u.t. = \frac{c_2 - c_1}{t_2 - t_1}$$

Com que es desactiva el detector de desbordaments, el tipus *derive* també s'utilitza per als *counter* que no són ni de 32 ni de 64 bits. En aquests casos es configura el límit mínim de velocitat a zero i quan hi ha el desbordament² el valor transformat es considera desconegut, ja que la velocitat calculada és inferior al límit imposat de valor zero.

Alguns exemples de comptador *derive* són una bomba de cabal que pot bombejar o succionar, un comptador elèctric que mesura el total d'energia consumida menys la produïda, el balanç de població que emigra menys la que immigra, el saldo d'una llibreta que té ingressos i despeses, la quantitat de productes d'una màquina de *vending*, etc.

Exemple Es mesura el saldo d'una llibreta d'estalvi que inicialment està a 0€, al cap de 10 segons rep un ingrés de 10€, al cap de 10 segons rep 20€ i al cap de 10 segons té una despesa de 15€.

Els valors que va prenent la llibreta (el comptador) es poden veure a la figura 5.5. Les mostres es prenen a l'instant abans que es faci l'operació d'ingrés o despesa; és a dir que la llibreta mostra el valor que ha tingut a l'interval anterior, la qual és la manera com s'entenen les sèries temporals a RRDtool.

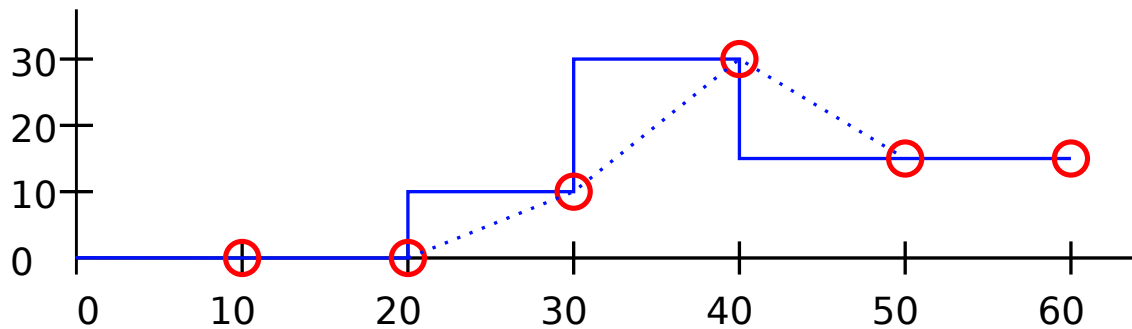


Figura 5.5: Llibreta d'estalvi: valor del saldo en blau, punts de mesura en vermell

A més, RRDtool entén que les variables mesurades són contínues. Per tant amb les mesures fetes de la llibreta entén que el valor s'ha incrementat de manera contínua en tot l'interval. Això es pot observar a la línia discontinua de la figura 5.5, que és l'aproximació que fa RRDtool ja que només es guarda la velocitat mitjana a l'interval.

²Això no es pot considerar una bona solució, vegeu nota 1.

Es crea una base de dades de nom *derive.rrd* amb un temps de mostreig de 10 segons i un registre anomenat *eurps* que emmagatzema la velocitat del comptador (del saldo de la llibreta) a partir de les lectures del saldo.

```
export TZ=UTC
rrdtool create derive.rrd --start 1262304000 --step 10 \
    DS:eurps:DERIVE:600:-U:U \
    RRA:AVERAGE:0.5:1:24
```

S'actualitza la base de dades amb els valors $x = [0, 0, 10, 30, 15, 15]$ que s'han mesurat als temps $t = [10, 20, 30, 40, 50, 60]$

```
rrdtool update derive.rrd 1262304010:0 1262304020:0 1262304030:10 \
1262304040:30 1262304050:15 1262304060:15
```

A continuació es mostren els valors emmagatzemats i a la figura 5.6 es mostra el gràfic que genera RRDtool:

```
00:00:00 UTC --> <row><v>NaN</v></row>
00:00:10 UTC --> <row><v>NaN</v></row>
00:00:20 UTC --> <row><v>0.0000000000e+00</v></row>
00:00:30 UTC --> <row><v>1.0000000000e+00</v></row>
00:00:40 UTC --> <row><v>2.0000000000e+00</v></row>
00:00:50 UTC --> <row><v>-1.5000000000e+00</v></row>
00:01:00 UTC --> <row><v>0.0000000000e+00</v></row>
```

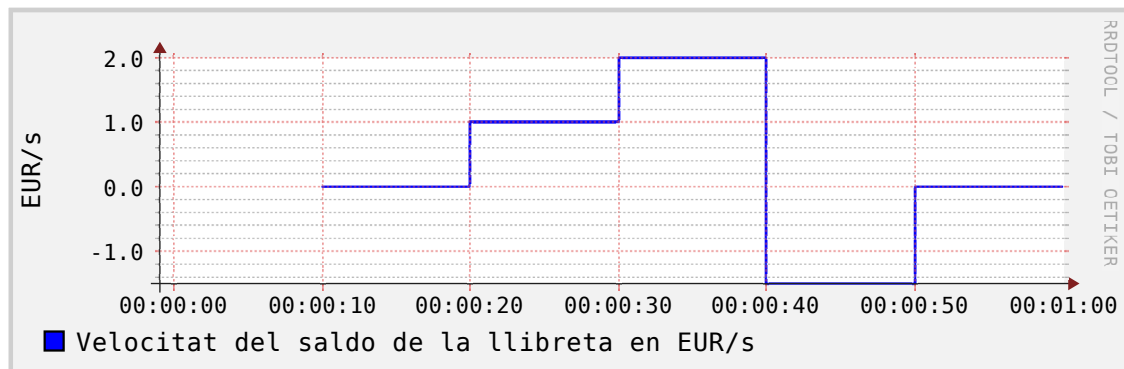


Figura 5.6: Velocitats emmagatzemades a RRDtool d'un tipus Derive

Aquest gràfic és com l'analitzat en l'exemple dels tipus *counter* (figura 5.4), amb la diferència que ara en els *derive* hi apareixen velocitats negatives. Com s'ha fet en els *counter*, es pot calcular els increments de saldo de cada interval multiplicant les velocitats pel temps de mostreig. Aleshores els increments de saldo són $[u, 0, 10, 20, -15, 0]$ que sumats donen correctament els 15€ de saldo final de la figura 5.5.

5.1.4 Comptador relatiu: *Absolute*

Les dades de tipus *Absolute* són de tipus comptador d'increments; per això l'anomenen comptador relatiu. Aquestes dades no són velocitats i per tant cal aplicar transformació.

Un comptador d'increments mesura directament la diferència des de l'última lectura, es pot veure com un comptador que es posa a zero a cada lectura. Per tant, per transformar les mesures a velocitat es divideix l'increment per la diferència de temps

$$\text{mitjana per } u.t. = \frac{\Delta c}{t_2 - t_1}$$

on ara, a diferència de l'equació 5.1, es mesura directament el comptatge (Δc) des de la mesura anterior.

Els increments mesurats poden ser positius o negatius; és a dir com en el cas *derive* les velocitats resultants poden ser positives o negatives. Per tant, el tipus *absolute* és el mateix que el *derive* amb la diferència que el valor mesurat, el qual és el valor actual del comptador, és un increment.

El primer valor que es desa a la base de dades ara es pot calcular, a diferència dels comptadors *counter* i *derive*. En aquest cas cal calcular l'increment de comptatge i en el primer valor és desconegut, ja que podria ser erroni suposar-lo com a zero. En el cas del comptador *absolute* també cal indicar quin és el temps inicial, però aquest es considera que és el mateix que el d'inici de la base de dades.

Alguns exemples de comptador *absolute* són els mateixos que en el *derive* però mesurats de manera relativa: increments de bombeig o succió d'una bomba, la mesura de l'energia elèctrica en un cert temps, els increments de població degut a la migració, els registres a una llibreta, la venda o repostatge de productes, etc.

Exemple Com en l'exemple del tipus *derive* (figura 5.5), es mesura el saldo d'una llibreta d'estalvi. Ara, però, les mesures de comptador són les operacions d'ingrés o de despesa. El saldo pren els mateixos valors que en l'exemple anterior però ara les mesures són els increments, positius o negatius, que es poden veure a la figura 5.7.

Les línies discontinües segueixen mostrant, com a la figura 5.5, l'aproximació que RRDtool fa del comptador com a variable contínua.

Es crea una base de dades de nom *absolute.rrd* amb un temps de mostreig de 10 segons i un registre anomenat *eurps* que emmagatzema la velocitat del comptador (el saldo de la llibreta) a partir dels increments de saldo.

```
export TZ=UTC
rrdtool create absolute.rrd --start 1262304000 --step 10 \
    DS:eurps:ABSOLUTE:600:-U:U \
    RRA:AVERAGE:0.5:1:24
```

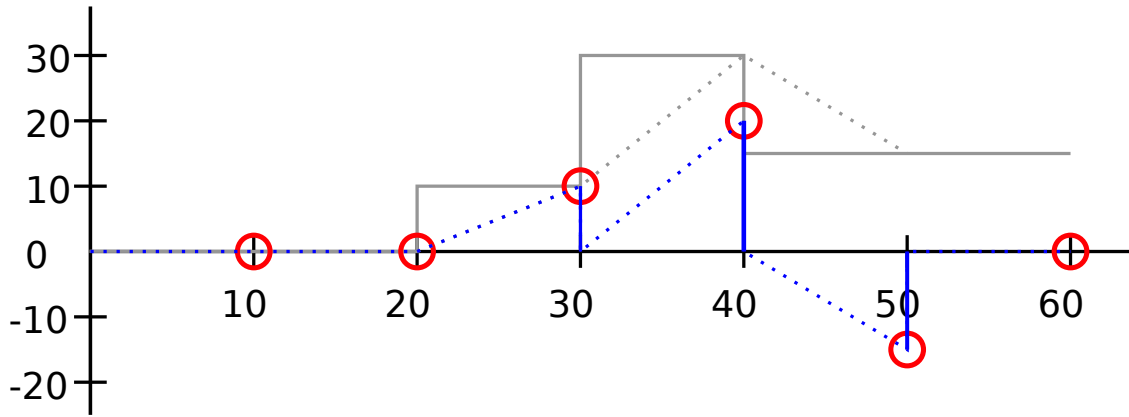


Figura 5.7: Llibreta d'estalvi: increments de saldo en blau, punts de mesura en vermell, saldo en gris

S'actualitza la base de dades amb els valors $x = [0, 0, 10, 20, -15, 0]$ que s'han mesurat als temps $t = [10, 20, 30, 40, 50, 60]$

```
rrdtool update absolute.rrd 1262304010:0 1262304020:0 1262304030:10\
1262304040:20 1262304050:-15 1262304060:0
```

A continuació es mostren els valors emmagatzemats i a la figura 5.8 es mostra el gràfic que genera RRDtool; és idèntic a la figura 5.6 però ara el primer valor és conegut.:

```
00:00:00 UTC --> <row><v>NaN</v></row>
00:00:10 UTC --> <row><v>0.0000000000e+00</v></row>
00:00:20 UTC --> <row><v>0.0000000000e+00</v></row>
00:00:30 UTC --> <row><v>1.0000000000e+00</v></row>
00:00:40 UTC --> <row><v>2.0000000000e+00</v></row>
00:00:50 UTC --> <row><v>-1.5000000000e+00</v></row>
00:01:00 UTC --> <row><v>0.0000000000e+00</v></row>
```

D'aquest gràfic, com en el casos *derive* i *counter*, es pot obtenir els increments de comptatge a cada interval si es multiplica cada valor pel temps de mostreig (10 segons). Aquests valors es podrien observar en un gràfic de barres, tot i que RRDtool només treballa amb variables velocitat contínues i per tant no pot mostrar exactament aquests gràfics de barres del comptador. A la figura 5.9 s'utilitza RRDtool perquè dibuixi els valors pintant l'àrea i multiplicant l'escala de l'eix Y per 10; perquè fos un gràfic de barres cal imaginar que a l'eix X hi ha les etiquetes corresponents als intervals: $(0, 10]$, $(10, 20]$, \dots , $(50, 60]$ i que cada un d'aquests intervals és una barra.

Ara bé, RRDtool sí que té funcions per tractar amb càlculs típics dels comptadors, per exemple calcular el total del comptador en un interval de temps donat. En aquest cas, el total del comptador des de l'inici fins a una data correspon al saldo

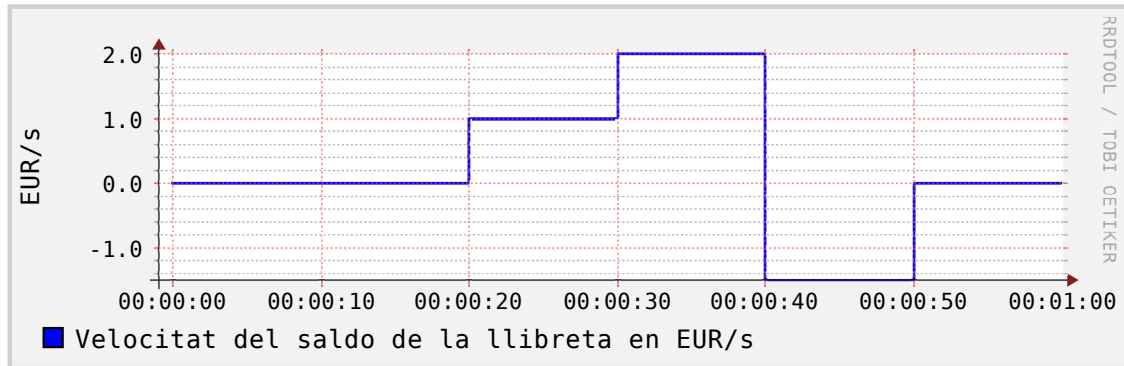


Figura 5.8: Velocitats emmagatzemades a RRDtool d'un tipus Absolute

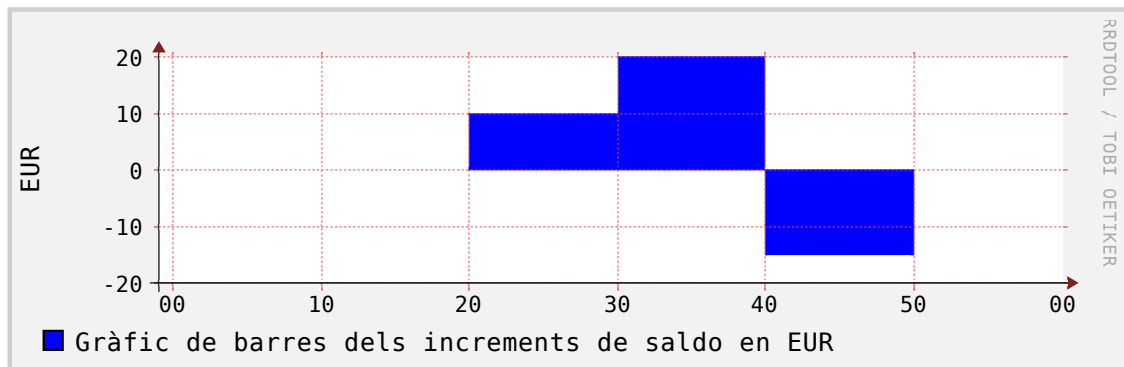


Figura 5.9: Gràfic de barres a RRDtool d'un comptador

de la llibreta en aquella data. A continuació es calcula el saldo des del temps 0 fins al temps 60, el qual correspon als 15€ de la figura 5.7.:

```
rrdtool graphv absolute.eps -a EPS --start 1262303999 --end \
1262304060 DEF:saldo=absolute.rrd:eurps:AVERAGE VDEF:tot=saldo,\
TOTAL PRINT:tot:%lf\EUR/s
```

```
print[0] = "15.000000EUR/s"
```

5.2 Normalització de l'interval

RRDtool emmagatzema les dades a intervals regulars, és a dir en el temps de mostreig desitjat (anomenat *step*). Com que a la pràctica és difícil complir-lo amb exactitud, cal modificar les dades perquè encaixin en aquest temps; és a dir s'ha de normalitzar l'interval de la sèrie temporal.

Per a modificar els valors de la sèrie temporal, en el context de velocitat vist a l'apartat anterior es pot plantejar una modificació de dades que tracti de mantenir l'àrea de sota la corba. La interpretació física és que es manté la quantitat total de magnitud. Per exemple si les dades són velocitats lineals en m/s es pot entendre el problema com que es modifica la velocitat però es manté l'espai recorregut. O bé si les dades són comptadors, com que s'ha desat la velocitat s'entén que s'està mantenint la quantitat total.

Així doncs, el problema a resoldre és el següent. Donada una sèrie temporal acotada a un temps final T_F

$$S = \{\mathbf{X}(\mathbf{t}); t = t_0, t_1, t_2, \dots, t_F\}$$

que no té un període de mesura regular

$$t_{i+1} \neq t_i + T$$

es vol convertir a una sèrie temporal normalitzada amb interval regulars tal que ara T correspongui al període de mostreig t_m :

$$S^N = \{\mathbf{X}^N(\mathbf{t}^N); t^N = t_0, t_1, t_2, \dots, t_k\}$$

$$t_{k+1}^N = t_k^N + t_m$$

$$t_k^N < T_F$$

on N indica que els valors i els temps ara contenen els valors normalitzats:

$$\mathbf{X}^N = [x^N(t_0^N), x^N(t_1^N), \dots, x^N(t_{T_i}^N)]$$

En resum, el vector de mesures té la forma

$$\mathbf{X}(\mathbf{t}) = [x_0, x_1, \dots, x_{F-1}, x_F]$$

i amb la mateixa mida el vector de temps de mesures

$$\mathbf{t} = [t_0, t_1 \dots, t_{T_f-1}, t_{T_f}]$$

Però el vector de valors normalitzats té una mida diferent

$$\mathbf{X}^N(\mathbf{t}^N) = [x_0^N, x_1^N, \dots, x_k^N]$$

on el vector de temps de mostreig normalitzats té la forma

$$\mathbf{t}^N = [0, t_m, 2t_m, 3t_m, \dots, kt_m]$$

A l'hora de convertir la sèrie temporal RRDtool utilitza el criteri de mantenir constant l'àrea sota la corba; a l'apartat 5.1.1 s'ha vist com RRDtool representa les sèries temporals en velocitat i en passat. En els exemples d'aquest apartat, però, les mesures es feien en el temps de mostreig exacte, a continuació es mostrarà com RRDtool gestiona els casos en els que les mostres no coincideixen amb el temps de mostreig.

5.2.1 Mostreig en temps real

En un primer cas, la sèrie temporal original està mostrejada amb un únic valor a cada interval, tal com es faria en un sistema en temps real

$$\mathbf{X}(\mathbf{t}) = [x(t_0) \dots, x(t_{T_f-1}), x(t_{T_f})]$$

$$0 < t_0 < t_1^N < \dots < t_{i-1}^N < t_{T_f-1} < t_i^N < t_{T_f}$$

En aquest cas, la sèrie temporal no té intervals regulars però sí que hi ha garantits uns terminis temporals en els que es disposa d'un valor i només hi ha un valor a cada interval:

$$t_{i+1} \neq t_i + T$$

$$t_{i+1} - t_i \leq t_m$$

A continuació es detalla com RRDtool normalitza la sèrie temporal en l'interval $[t_{i-1}^N, t_i^N]$, es pot veure una interpretació gràfica a la figura 5.10.

Per una banda, es calcula l'àrea en l'interval seleccionat a partir dels valors mesurats

$$A([t_{i-1}^N, t_i^N]) = A([t_{i-1}^N, t_{i-1}]) + A([t_{i-1}, t_i^N]) = (t_{i-1} - t_{i-1}^N)x_{i-1} + (t_i^N - t_{i-1})x_i$$

i per altra banda, la normalització de l'interval s'expressa amb un valor de velocitat X_i^N que es representa en el temps t_i^N degut a que RRDtool interpreta els valors en passat (constants en l'interval anterior)

$$A^N = (t_i^N - t_{i-1}^N)x_i^N = t_m x_i^N$$

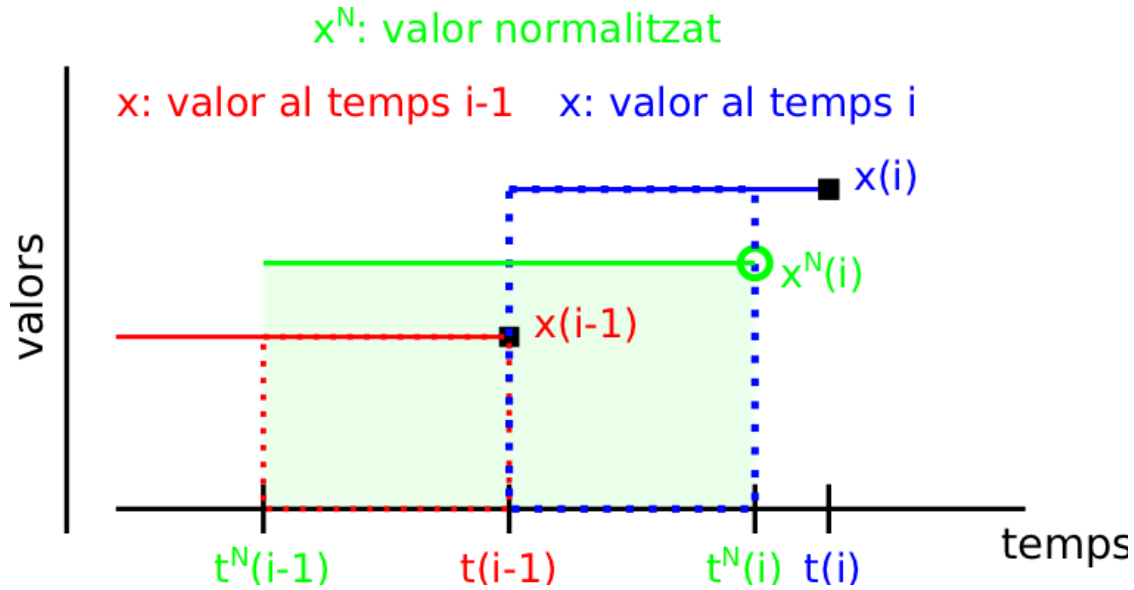


Figura 5.10: Normalització d'un interval

Aleshores, s'igualen les dues expressions per tal que en normalitzar es conservi l'àrea en l'interval

$$A^N = A([t_{i-1}^N, t_i^N])$$

d'on aïllant s'obté l'equació del valor normalitzat

$$x_i^N = \frac{(t_{i-1} - t_{i-1}^N)x_{i-1} + (t_i^N - t_{i-1})x_i}{t_m} \quad (5.3)$$

El mateix es pot aplicar per tots els intervals de \mathbf{X} per obtenir \mathbf{X}^N .

Exemple Agafem una nova base de dades `velocitat.rrd` i ara l'actualitzem amb el mateix perfil de velocitat de la figura 5.1 però mostrejat de manera irregular com es veu a la figura 5.11. A cada interval de mostreig hi segueix havent una mesura, i per tant compleixen el període de mostreig, però el temps de mostreig és irregular. Així ara els valors de velocitat mesurats són $x = [0, 0,5, 1, 2,8, 3,2, 2, 0]$ en els temps $t = [10, 15, 20, 38, 42, 55, 65]$.

S'insereixen els valors a la base de dades:

```
rrdtool update velocitat.rrd 1262304010:0 1262304015:0.5 \
1262304020:1 1262304038:2.8 1262304042:3.2 1262304055:2 \
1262304065:0
```

A continuació es mostren els valors emmagatzemats i el gràfic que genera RRDtool (figura 5.12):

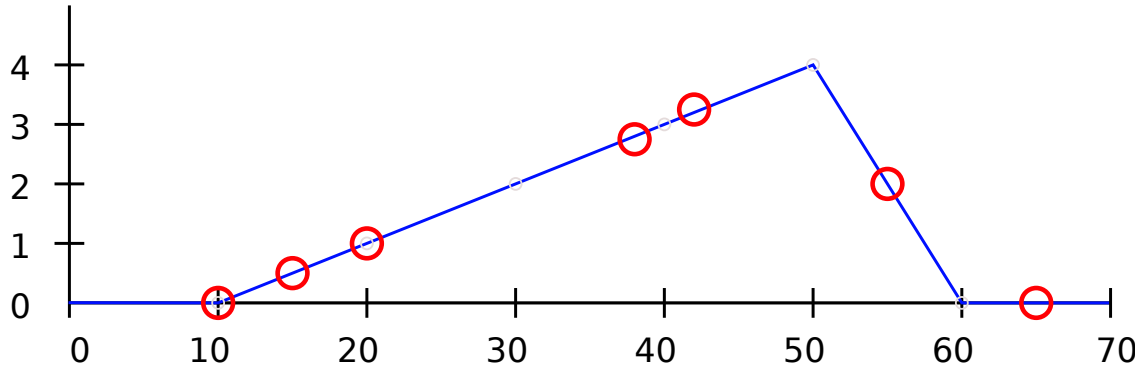


Figura 5.11: Mostreig en temps real: perfil de velocitat en blau, punts de mesura irregulars en vermell, període de mostreig de 10 segons

```
00:00:00 UTC --> <row><v>NaN</v></row>
00:00:10 UTC --> <row><v>0.0000000000e+00</v></row>
00:00:20 UTC --> <row><v>7.5000000000e-01</v></row>
00:00:30 UTC --> <row><v>2.8000000000e+00</v></row>
00:00:40 UTC --> <row><v>2.8800000000e+00</v></row>
00:00:50 UTC --> <row><v>2.2400000000e+00</v></row>
00:01:00 UTC --> <row><v>1.0000000000e+00</v></row>
```

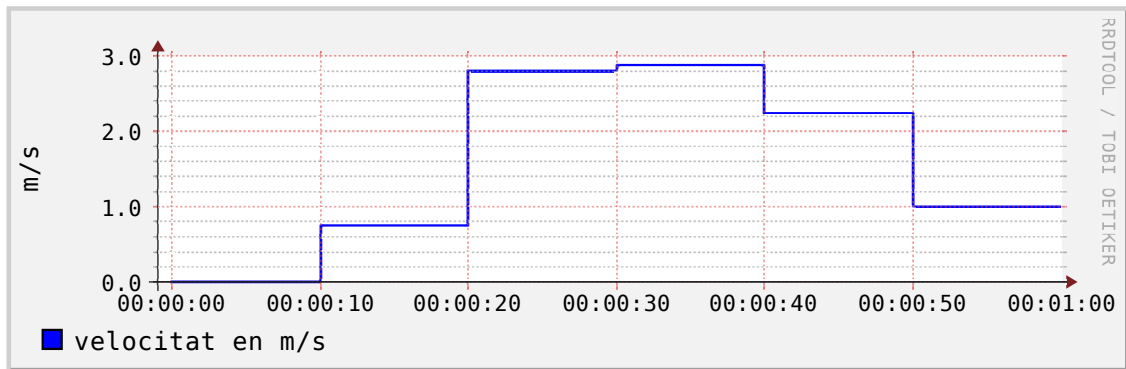


Figura 5.12: Valors emmagatzemats a RRDtool d'un mostreig en temps real

Aplicant l'equació 5.3 als valors mesurats es pot comprovar que s'obtenen els valors normalitzats calculats per RRDtool $\mathbf{X}^N = [u, 0, 0,75, 2,8, 2,88, 2,24, 1]$. Cal notar que a RRDtool el primer valor normalitzat, el qual correspon al temps t_0^N , és desconegut ja que no s'admeten insercions en el temps t_0 .

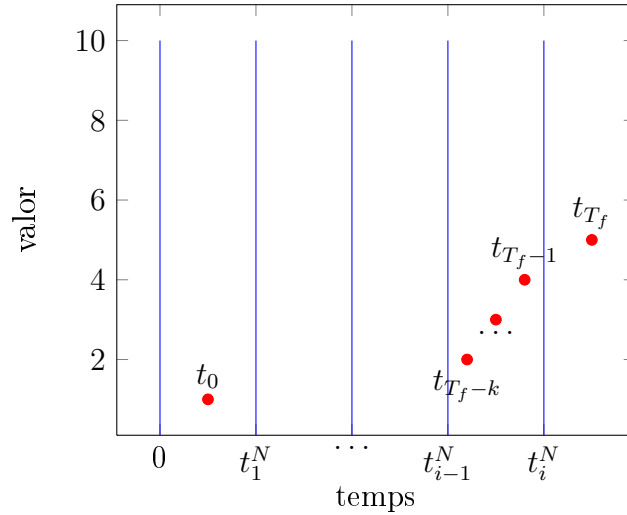


Figura 5.13: Representació d'ultramostreig

5.2.2 Ultramostreig

En la normalització anterior de mostreig en temps real, en cada període de mostreig només hi havia una mesura. Ara bé, RRDtool permet que en cada interval hi hagi més d'una mesura i per tant hi pot haver més d'un valor a l'hora de normalitzar; anomenem aquest mostreig com ultramostreig (*upsampling*), representat a la figura 5.13.

$$\mathbf{X}(\mathbf{t}) = [x(t_0) \dots, x(t_{T_f-k}), \dots, x(t_{T_f-2}), x(t_{T_f-1}), x(t_{T_f})]$$

$$0 < t_0 < t_1^N < \dots < t_{i-1}^N < t_{T_f-k} < \dots < t_{T_f-1} < t_i^N < t_{T_f}$$

Aleshores, el valor normalitzat es calcula de la mateixa manera que a l'equació 5.3, però ara el càlcul és una ponderació pel temps de tots els valors que cauen a dins de l'interval de normalització

$$x_i^N = \frac{(t_{T_f-k} - t_{i-1}^N)x_{f-k} + \dots + (t_{T_f-1} - t_{T_f-2})x_{f-1} + (t_i^N - t_{T_f-1})x_f}{t_m} \quad (5.4)$$

Seguint aquesta equació de manera iterativa, es pot calcular el vector de valors normalitzats \mathbf{X}^N per tots els $\mathbf{X}(\mathbf{t})$ amb l'algoritme següent:

Normalització d'una sèrie temporal amb ultramostreig
a períodes de mostreig regulars

INPUT:

vector de valors $\mathbf{X} = [x_0, x_1, \dots, x_f]$

vector de temps $\mathbf{t} = [t_0, t_1, \dots, t_f]$

període de mostreig regular t_m

OUTPUT:

```

vector de valors normalitzats  $\mathbf{X}^N = [x_0^N, x_1^N, \dots, x_k^N]$ 

 $x_0^N := \text{unknown}$ 
 $t_0^N := 0$ 
 $t_1^N := t_0^N + t_m$ 
 $i := 1$ 

 $A := x_0 t_0 <-- \text{\`a}rea acumulada inicial$ 
 $k := 1$ 

mentre  $k \leq \text{dim}(x)$  fes

  si  $t_k < t_i^N$  llavors
     $A := A + x_k(t_k - t_{k-1}) <-- \text{acumulaci\`o}$ 
     $k := k+1$ 
  sino
     $x_i^N := \frac{A + x_k(t_i^N - t_{k-1})}{t_m}$ 

     $t_{i+1}^N := t_i^N + t_m$ 
     $A := x_k * (t_k - t_i^N) <-- \text{\`a}rea acumulada inicial$ 
     $k := k+1$ 
     $i := i+1$ 
  fsi

fmentre

```

Exemple Agafem una nova base de dades `velocitat.rrd` i ara l'actualitzem amb el mateix perfil de velocitat de la figura 5.1 però amb més mostres en algun interval com es veu a la figura 5.14. A cada interval de mostreig hi segueix havent com a mínim una mesura i compleixen el període de mostreig però en algun interval hi ha més d'una mostra. Així ara el valors de velocitat mesurats són $x = [0, 0,5, 0,8, 1, 2,8, 3,2, 3,5, 3,8, 2, 0]$ en els temps $t = [10, 15, 18, 20, 38, 42, 45, 48, 55, 65]$.

S'insereixen els valors a la base de dades:

```

rrdtool update velocitat.rrd 1262304010:0 1262304015:0.5 \
1262304018:0.8 1262304020:1 1262304038:2.8 1262304042:3.2 \
1262304045:3.5 1262304048:3.8 1262304055:2 1262304065:0

```

A continuació es mostren els valors emmagatzemats i el gràfic que genera RRDtool (figura 5.15):

```

00:00:00 UTC --> <row><v>NaN</v></row>
00:00:10 UTC --> <row><v>0.0000000000e+00</v></row>

```

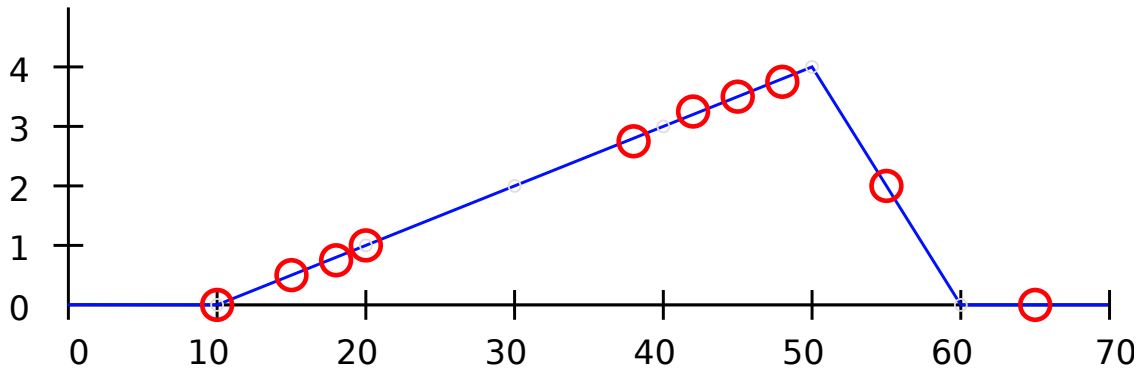


Figura 5.14: Ultramostreig: perfil de velocitat en blau, punts de mesura (alguns ultramostrejats) en vermell, període de mostreig de 10 segons

```
00:00:20 UTC --> <row><v>6.9000000000e-01</v></row>
00:00:30 UTC --> <row><v>2.8000000000e+00</v></row>
00:00:40 UTC --> <row><v>2.8800000000e+00</v></row>
00:00:50 UTC --> <row><v>3.2300000000e+00</v></row>
00:01:00 UTC --> <row><v>1.0000000000e+00</v></row>
```

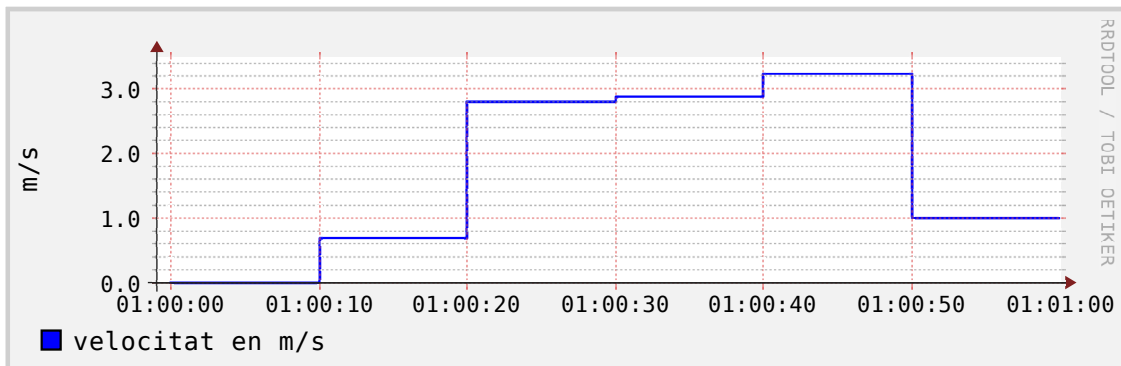


Figura 5.15: Valors emmagatzemats a RRDtool d'un ultramostreig

Respecte a la figura 5.12 del cas de mostreig en temps real, ara només han canviat els valors dels intervals ultramostrejats mentre que els altres intervals continuen valent el mateix: $\mathbf{X}^N = [u, 0, \underline{0,69}, 2,8, 2,88, \underline{2,24}, 1]$. Els valors que canvien són molt semblants als anteriors, però es pot dir que es té una millor aproximació a la velocitat real ja que en aquests intervals s'ha mostrejat més.

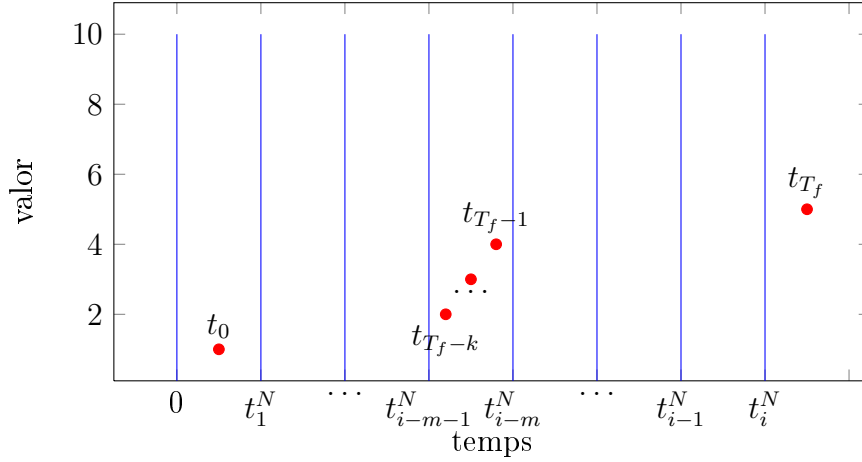


Figura 5.16: Representació d'inframostreig

5.2.3 Inframostreig

Per altra banda, quan en un interval no hi ha cap mesura, és a dir que el temps de mostreig supera al període de mostreig, s'anomenem inframostreig (*downsampling*).

$$\mathbf{X}(\mathbf{t}) = [x(t_0) \dots, x(t_{T_f-1}), x(t_{T_f})]$$

$$t_{i-m-1}^N < t_{T_f-1} < t_{i-m}^N < \dots < t_{i-1}^N < t_i^N < t_{T_f}$$

Llavors per normalitzar amb inframostreig RRDtool utilitza el valor del següent interval. En aquest cas se suposa que les mesures no tenen termini, més endavant s'afegeix el problema del termini (vegeu apartat 5.2.4).

És a dir, quan entre dues mesures hi ha més temps que el temps de mostreig llavors el valor de l'última mesura $\mathbf{x}(t_{T_f})$ s'expandeix enrere fins arribar a l'altra $\mathbf{x}(t_{T_f-1})$ i es calcula el valor normalitzat d'aquest gran interval. Si hi ha inframostreig però no hi ha ultramostreig

$$t_{i-m-1}^N < t_{T_f-1} < t_{i-m}^N$$

el valor normalitzat és:

$$x^N = \frac{(t_{T_f-1} - t_{i-m-1}^N)x_{f-1} + (t_i^N - t_{T_f-1})x_f}{t_i^N - t_{i-m-1}^N}$$

Generalitzant, si ahora hi ha inframostreig i ultramostreig, representat a la figura 5.16,

$$t_{i-m-1}^N < t_{T_f-k} < \dots < t_{T_f-1} < t_{i-m}^N$$

el valor normalitzat es calcula de la mateixa manera que a l'equació 5.4, però en l'interval $[t_{i-m-1}^N, t_i^N]$:

$$x^N = \frac{(t_{T_f-k} - t_{i-m-1}^N)x_{f-k} + \dots + (t_{T_f-1} - t_{T_f-2})x_{f-1} + (t_i^N - t_{T_f-1})x_f}{t_i^N - t_{i-m-1}^N} \quad (5.5)$$

Finalment, aquest valor normalitzat per inframostreig, hi hagi ultramostreig o no, és utilitzat en tots els intervals afectats:

$$x^N = x_{i-m}^N = \dots = x_i^N$$

Es modifica l'algoritme de càlcul iteratiu anterior segons aquest inframostreig:

Normalització d'una sèrie temporal amb inframostreig
a períodes de mostreig regulars

INPUT:

vector de valors $\mathbf{X} = [x_0, x_1, \dots, x_f]$

vector de temps $\mathbf{t} = [t_0, t_1, \dots, t_f]$

període de mostreig regular t_m

OUTPUT:

vector de valors normalitzats $\mathbf{X}^N = [x_0^N, x_1^N, \dots, x_k^N]$

$x_0^N := \text{unknown}$

$t_0^N := 0$

$t_1^N := t_0^N + t_m$

$i := 1$

$A := x_0 t_0$

$k := 1$

mentre $k \leq \text{dim}(x)$ fes

si $t_k < t_i^N$ llavors

$A := A + x_k(t_k - t_{k-1})$

$k := k+1$

sino

$N_{inf} := (t_k - t_i^N) \text{ div } t_m <-- \text{n. d'intervals amb inframostreig}$

$$x_i^N := \frac{A + x_k(t_i^N - t_{k-1}) + x_k \cdot N_{inf} \cdot t_m}{t_m(1 + N_{inf})}$$

mateix valor per cada interval amb inframostreig

$x_{i+N_{inf}}^N := \dots := x_{i+1}^N := x_i^N$

$t_{i+1}^N := t_i^N + (N_{inf} + 1)t_m$

$i := i + (N_{inf} + 1)$

$A := x_k(t_k - t_{i-1}^N)$

$k := k+1$

fsi

fmentre

Exemple Agafem una nova base de dades `velocitat.rrd` i ara l'actualitzem amb el mateix perfil de velocitat de la figura 5.1 però en algun interval no hi ha mostres, com es veu a la figura 5.17. És important ressaltar que aquests intervals no compleixen el període de mostreig Així ara el valors de velocitat mesurats són $x = [0, 0,5, 2,5, 3,8, 2, 0]$ en els temps $t = [10, 15, 35, 48, 55, 65]$.

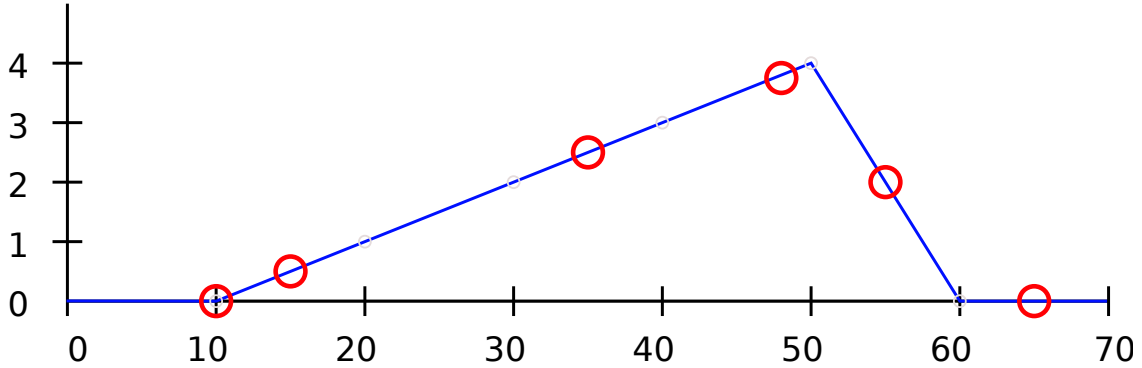


Figura 5.17: Inframostrat: perfil de velocitat en blau, punts de mesura (alguns inframostrats) en vermell, període de mostreig de 10 segons

S'insereixen els valors a la base de dades:

```
rrdtool update velocitat.rrd 1262304010:0 1262304015:0.5 \
1262304035:2.5 1262304048:3.8 1262304055:2 1262304065:0
```

A continuació es mostren els valors emmagatzemats i el gràfic que genera RRDtool (figura 5.18):

```
00:00:00 UTC --> <row><v>NaN</v></row>
00:00:10 UTC --> <row><v>0.0000000000e+00</v></row>
00:00:20 UTC --> <row><v>2.0000000000e+00</v></row>
00:00:30 UTC --> <row><v>2.0000000000e+00</v></row>
00:00:40 UTC --> <row><v>3.1500000000e+00</v></row>
00:00:50 UTC --> <row><v>3.4400000000e+00</v></row>
00:01:00 UTC --> <row><v>1.0000000000e+00</v></row>
```

Respecte a la figura 5.12 del cas de mostreig en temps real, aquest gràfic només es diferencia en els intervals inframostrats. Ara els intervals afectats per l'inframostrat (10, 20] i (20, 30] tenen el mateix valor de 2, el qual resulta d'aplicar l'equació 5.5: $x_{20}^N = x_{30}^N = \frac{(15-10)0,5+(30-15)2,5}{30-10} = 2$. En aquests dos intervals hi ha una pitjor aproximació a la velocitat real ja que s'ha mostreat molt poc.

5.2.4 Tractament de dades desconegudes

En les seccions anteriors, les mesures de les sèries temporals sempre tenen un valor numèric. En concret són números reals (\mathbb{R}), els quals es representen informàtica-

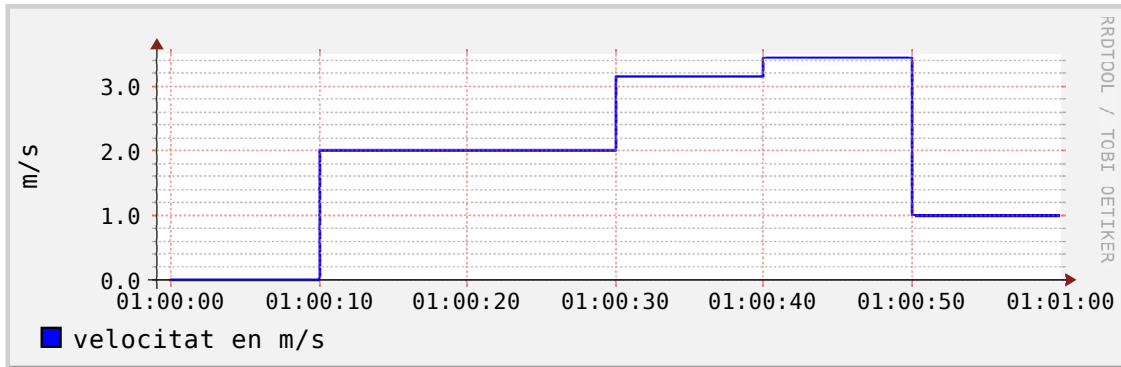


Figura 5.18: Valors emmagatzemats a RRDtool d'un inframostreig

ment amb el format numèric de precisió doble seguint l'estàndard IEEE 754, [30].

Però les mesures, a més a més de tenir un valor, també poden ser desconegudes (*unknown data*); és a dir, que no existeixin o que s'ignorin (es descartin). Aquests valors desconeguts també estan contemplats a l'estàndard IEEE 754 i es representen amb el valor numèric especial NaN (*Not a Number*).

L'algoritme de normalització d'interval ha de saber manipular les mostres amb mesures desconegudes. En el cas de RRDtool, les dades desconegudes poden venir per tres vies:

- la mesura no existeix,
- el temps de mesura ha superat el termini o
- el valor de mesura està fora dels límits.

A continuació es detallen aquestes tres vies i al final es modifica l'algoritme de normalització en concordança amb el tractament d'aquestes dades desconegudes.

Mesura inexistent

Una mesura no existeix quan no s'ha pogut establir contacte amb el sensor o quan aquest retorna valors que no són numèrics. Quan una mesura no existeix, es considera desconeguda i s'insereix a la base de dades amb el valor 'U' (*unknown*) tot i que queda representat amb el valor NaN segons l'estàndard IEEE 754.

Les mesures també es desconeixen quan s'inicialitza la base de dades. A l'inici d'una base de dades RRDtool es creen tots els registres fins al temps actual però aquestes mesures no han existit mai; per tant prenen 'desconegut' com a valor.

Exemple S'insereix una dada amb valor conegut i l'altra amb valor desconegut

```
rrdtool update velocitat.rrd 1262304010:1 1262304020:U
```

```

23:59:30 UTC --> <row><v>NaN</v></row>
23:59:40 UTC --> <row><v>NaN</v></row>
23:59:50 UTC --> <row><v>NaN</v></row>
00:00:00 UTC --> <row><v>NaN</v></row>
00:00:10 UTC --> <row><v>1.0000000000e+00</v></row>
00:00:20 UTC --> <row><v>NaN</v></row>

```

Per una banda, tots els valors anteriors al temps 0 (aquest inclòs) són desconeguts perquè no es coneix res abans que s'inicialitzi la base de dades. Per altra banda, s'ha inserit un valor numèric en el temps 10 i el valor 'desconegut' en el temps 20.

Temps de termini

El temps de termini, el qual RRDtool anomena *heartbeat* (t_h), és el temps màxim que s'admet entre dues mesures. La mesura actual s'ignora si ha passat més temps que el termini des de la mesura anterior.

$$t_i - t_{i-1} > t_h \longrightarrow x(t_i) = \text{unknown}$$

A RRDtool el termini pot ser més gran, igual o més petit que el temps de mostreig. És a dir, que el termini no només afecta en els casos d'inframosteig sinó que també al casos d'ultramosteig.

Cal no confondre el termini de RRDtool (*heartbeat*) amb el termini utilitzat pels sistemes de temps real (*deadline*). El *heartbeat* es mesura entre mostres amb l'objectiu de tenir dades 'fresques' i en canvi el *deadline* es mesura a partir del temps d'activació amb l'objectiu de complir temps de càlcul. En el cas de tasques periòdiques, com és el cas del temps de mostreig a RRDtool, en temps real no té sentit parlar de terminis més grans que el període de mostreig, en canvi en els RRDtool ja s'ha vist que poden tractar aquestes situacions, anomenades inframosteig.

A continuació s'estudien quatre casos diferents segons els valors que pot adoptar el temps de termini respecte del temps de mostreig. Els quatre casos es poden veure a l'eix vertical de la figura 5.19 representant el termini amb t_h , el període de mostreig amb t_m i exemples de temps de mesura amb una circumferència vermella a l'eix horitzontal.

Primer de tot, del concepte de RRDtool es desprèn que en qualsevol cas sempre hi pot haver ultramosteig. Això és degut a que els terminis són els temps màxims entre mesures però no s'especifica un temps mínim. Aquestes situacions es poden veure a la figura 5.19: a l'eix horitzontal hi ha marcats els temps de mostreig exactes però el temps de mesura pot prendre qualsevol valor mentre sigui inferior al termini prefixat.

Un termini més gran que el temps de mostreig ($t_h > t_m$) indica que s'accepten els casos d'inframosteig i el temps del termini limita el temps durant el qual es

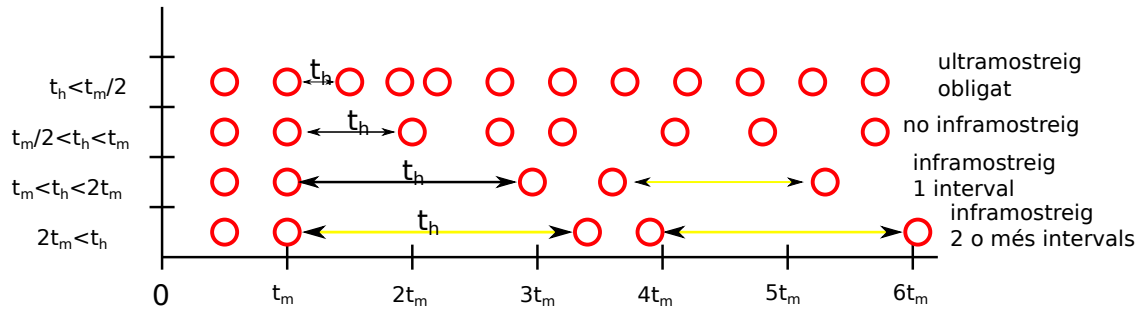


Figura 5.19: Comparació de mostrejigs amb terminis t_h diferents pel mateix període de mostreig t_m , punts de mesura en vermell i intervals inframostrejats en groc

permet no disposar de mesures. A més, si el termini és més petit que el doble del temps de mostreig ($t_m < t_h < 2t_m$), com a màxim només hi pot haver un interval amb inframostreig. Però si el termini supera el doble del temps de mostreig, llavors l'inframostreig es pot donar en dos intervals, en tres intervals si és el triple, etc.

Un termini més petit que el temps de mostreig ($t_h < t_m$) indica que no es vol inframostreig; és a dir que en cada interval com a mínim hi ha d'haver una mostra. Però, al mateix temps, un termini més petit que el temps de mostreig fa que a la llarga s'obligui a ultramostrejigs en alguns intervals. Encara més, si el termini és inferior a la meitat del temps de mostreig, llavors hi ha d'haver ultramostreig en tots els intervals.

Normalment, en els usos de RRDtool, el valor del termini s'assigna a un valor més gran que el doble del temps de mostreig ($t_h > 2t_m$) per tal de poder tenir folgança en les mesures. Cal observar que el cas del mostreig únic es representa amb el termini doblant el temps de mostreig ($t_h = 2t_m$); això permet que la mesura següent se situï en qualsevol temps de l'interval de mostreig següent.

El cas del mostreig únic és comparable als sistemes en temps real. En els sistemes en temps real el temps entre mostres com a màxim pot doblar el temps de mostreig, mentre es compleixi que en cada interval de mostreig hi ha una mostra. En el cas de RRDtool, quan $t_h = 2t_m$, hi pot haver inframostreig i per tant es pot no complir el temps real requerit pel mostreig únic. En conclusió, el temps de termini a RRDtool només s'utilitza per avaluar si les dades són 'fresques', la responsabilitat de les mesures i del temps real recau a la part d'adquisició de dades dels sistemes de monitoratge.

A RRD el termini es defineix a l'hora de crear una base de dades però es pot tornar a configurar. Cada variable mesurada (cada registre) té el seu termini.

Exemple Ara es crea la base de dades `velocitat.rrd` amb un temps de mostreig de 10 segons i un termini de 6 segons.

```
rrdtool create velocitat.rrd --start 1262304000 --step 10 \
    DS:mps:GAUGE:6:-U:U \
    RRA:AVERAGE:0.5:1:24
```

S'actualitza la base de dades amb els valors $x = [1, 1, 1]$ en els temps $t = [6, 10, 20]$.

```
rrdtool update velocitat.rrd 1262304006:1 1262304010:1 1262304020:1
```

A continuació es mostren els valors emmagatzemats:

```
00:00:00 UTC --> <row><v>NaN</v></row>
00:00:10 UTC --> <row><v>1.0000000000e+00</v></row>
00:00:20 UTC --> <row><v>NaN</v></row>
```

Les mesures implicades en el primer interval $[0,10]$ han complert els terminis ja que el temps entre mesures és igual o inferior a 6 segons. Però en el segon interval $[10,20]$, l'última mesura dista 10 segons respecte de l'anterior i per tant s'ignora i es considera desconeguda.

Límits

Una altra comprovació que es fa a les mesures és si estan dins d'un rang. A RRDtool es comprova que el valor mesurat és més gran que un límit inferior i que és més petit que un límit superior. Si la mesura està fora de rang, s'ignora i es considera desconeguda.

$$x(t_i) > L_{\max} \longrightarrow x(t_i) = \textit{unknown}$$

$$x(t_i) < L_{\min} \longrightarrow x(t_i) = \textit{unknown}$$

El límits sempre es calculen a les mesures un cop ja han passat l'etapa de transformació a velocitat. En el cas dels comptadors es comprova que la velocitat calculada estigui dins del rang però no es comprova la magnitud comptada, és a dir el valor que realment s'ha mesurat.

A RRDtool els límits es defineixen a l'hora de crear una base de dades però es poden tornar a configurar. Cada variable mesurada, això és cada registre, té el seus límits. Si no es vol comprovar el rang de les mesures, els límits es poden configurar amb el valor desconegut (U) per tal que no es tinguin en compte.

Exemple Ara es crea la base de dades velocitat.rrd amb un límit inferior de 0 i un límit superior desconegut.

```
rrdtool create velocitat.rrd --start 1262304000 --step 10 \
    DS:mps:GAUGE:600:0:U \
    RRA:AVERAGE:0.5:1:24
```

Ara s'actualitza la base de dades amb els valors $x = [0, 1000, -1]$ en els temps $t = [10, 20, 30]$.

```
rrdtool update velocitat.rrd 1262304006:1 1262304010:1 1262304020:1
```

A continuació es mostren els valors emmagatzemats:

```
00:00:00 UTC --> <row><v>NaN</v></row>
00:00:10 UTC --> <row><v>0.0000000000e+00</v></row>
00:00:20 UTC --> <row><v>1.0000000000e+03</v></row>
00:00:30 UTC --> <row><v>NaN</v></row>
```

No hi ha límit superior però sí que n'hi ha d'inferior. En el temps 30 el valor és més petit que el límit i per tant s'ignora la mesura i es considera desconeguda.

Càlcul amb dades desconegudes

Fins ara s'ha vist les tres vies per on es poden obtenir dades desconegudes en la normalització d'interval. Cal notar que les tres vies només serveixen per considerar si una mesura és desconeguda o no, però una mesura desconeguda no és suficient per considerar tot l'interval normalitzat desconegut.

A RRDtool, l'interval es considera desconegut³ quan el temps de mesures desconegudes (T_u) és superior a la meitat del període de mostreig.

$$T_u = \sum_{\forall x_f = \text{unknown}} (t_f - t_{f-1})$$

$$\text{si } T_u > t_m/2 \longrightarrow x_{i^N}^N = \text{unknown}$$

sent T_u el temps total que en un interval els valors \mathbf{X} prenen valor desconegut.

En cas contrari es calcula l'interval amb els valors coneguts. És a dir, l'algoritme de normalització d'interval ha de saber manipular les mostres amb mesures desconegudes. Per normalitzar, les dades desconegudes es considera que valen la mitjana de l'interval ($x_i^N = x_u$). D'aquesta manera no afecta a la velocitat mitjana de l'interval. Tot seguit es demostra.

Per una banda, la mitjana de l'interval sense tenir en compte els valors desconeguts es calcula utilitzant l'equació (5.4) només en els valors coneguts. Per facilitar la comprensió, es reescriu l'equació utilitzant t_a , t_b i t_j , representats a la figura 5.20:

$$t_a = t_{T_f-k} - t_{i-1}^N \quad \text{i} \quad t_b = t_i^N - t_{T_f-1}$$

$$t_j = t_{T_f-1-j} - t_{T_f-2-j} \quad \forall j = [0, k-2]$$

³A l'etapa de normalització d'interval no es pot triar la quantitat de desconeguts que són acceptables, ni fins i tot poder decidir si un valor desconegut ja fa calcular tot l'interval com desconegut. A l'etapa de consolidació sí que és possible (v. secció 5.3.1).

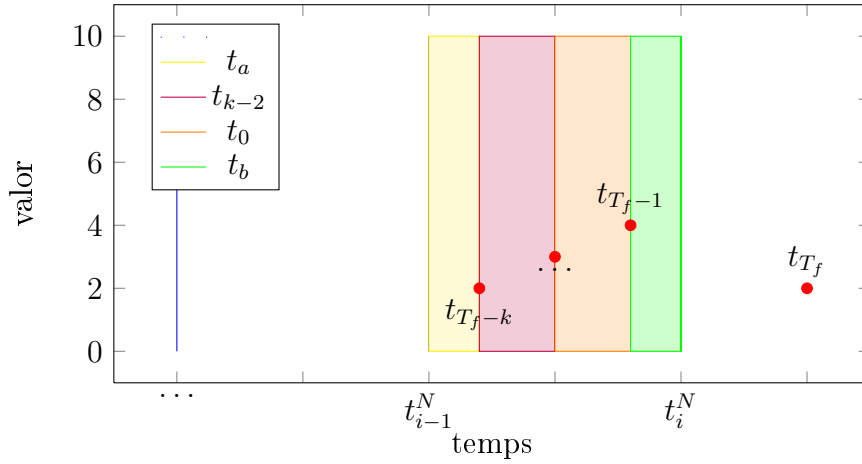


Figura 5.20: Representació d'ultramostreig simplificada

$$x_i^N = \frac{t_a x_a + \sum_{j=0}^{k-2} (t_j x_j) + t_b x_b}{t_a + \sum_{j=0}^{k-2} (t_j) + t_b} \quad (5.6)$$

Per altra banda, l'equació tenint en compte els valors desconeguts (x_u) és

$$x_i^N = \frac{t_a x_a + \sum_0^{k-2} (t_j x_j) + t_b x_b + T_u x_u}{t_a + \sum_0^{k-2} (t_j) + t_b + T_u} \quad (5.7)$$

Substituint (5.6) a (5.7)

$$x_i^N = \frac{x_i^N (t_a + \sum_0^{k-2} (t_j) + t_b) + T_u x_u}{t_a + \sum_0^{k-2} (t_j) + t_b + T_u}$$

operant amb aquesta expressió

$$x_i^N - \frac{x_i^N(t_a + \sum_0^{k-2}(t_j) + t_b) + T_u x_u}{t_a + \sum_0^{k-2}(t_j) + t_b + T_u} = 0$$

$$\frac{x_i^N(t_a + \sum_0^{k-2}(t_j) + t_b + T_u) - x_i^N(t_a + \sum_0^{k-2}(t_j) + t_b) - T_u x_u}{t_a + \sum_0^{k-2}(t_j) + t_b + T_u} = 0$$

$$\frac{T_u x_i^N + T_u x_u}{t_a + \sum_0^{k-2}(t_j) + t_b + T_u} = 0$$

Aleshores

$$x_i^N = x_u$$

Per tant si els valors desconeguts valen el mateix que la mitjana ($x_u = x_i^N$), aquesta mitjana es pot calcular sense tenir-los en compte com a l'equació 5.6. En aquesta equació el temps de l'interval queda reduït, ja que el temps total de l'interval equival al temps de mostreig

$$t_m = t_a + \sum_0^{k-2}(t_j) + t_b + T_u$$

i per tant es pot reexpressar a l'eq. 5.6 com el temps de mostreig descomptant-hi els temps en valors desconeguts

$$t_a + \sum_0^{k-2}(t_j) + t_b = t_m - T_u$$

En resum, la normalització de l'interval es pot calcular amb l'àrea acumulada dels valors sense comptar-hi els desconeguts i al final dividida entre el temps de mostreig descomptant-hi els temps en valors desconeguts

$$x_i^N = \frac{t_a x_a + \sum_0^{k-2}(t_j x_j) + t_b x_b}{t_m - T_u} \quad \forall x(t) \neq \text{unknown}$$

Com que es dona un valor al desconegut, això fa que la magnitud comptada augmenti. És a dir, s'està suposant que el comptador ha seguit mesurant a una velocitat semblant a les altres de l'interval.

Aquest augment s'observa seguint l'equació $e = x(t)$

$$e_{mesurat} = x_a t_a + \sum_0^{k-2}(x_j t_j) + x_b t_b$$

$$e_{emmagatzemat} = x_a t_a + \sum_0^{k-2} (x_j t_j) + x_b t_b + x_i^N T_u$$

5.2.5 Algoritme de normalització

Ara es modifica l'algoritme tenint compte les dades desconegudes i també els ultramostrejos i inframostrejos dels intervals. Aquest és l'algoritme complet de normalització d'intervals.

Normalització d'una sèrie temporal

a períodes de mostreig regulars

INPUT:

vector de valors $\mathbf{X} = [x_0, x_1, \dots, x_f]$

vector de temps $\mathbf{T} = [t_0, t_1, \dots, t_f]$

període de mostreig regular t_m

termini H

llindars màxim X^{MAX} i mínim X^{MIN}

OUTPUT:

vector de valors normalitzats $\mathbf{X}^N = [x_0^N, x_1^N, \dots, x_k^N]$

$x_0^N := \text{unknown} <--$ primer valor normalitzat desconegut

$t^N := t_m <--$ següent interval de normalització

$i := 1$

$A := 0 <---$ àrea acumulada

$t_{ant} := 0 <--$ temps de la mostra anterior

$T_u := 0 <--$ temps en valor desconegut

per cada parella (x, t) en \mathbf{X} i \mathbf{T} fes:

avalua termini i llindar:

si $t - t_{ant} > H$ o no $X^{MIN} \leq x \leq X^{MAX}$ llavors $x := \text{unknown}$ fsi

si $t < t^N$ llavors

si $x = \text{unknown}$ llavors

$T_u := T_u + t - t_{ant} <--$ acumula temps desconegut

sino

$A := A + x(t - t_{ant}) <--$ acumula valor

fsi

sino $<--$ càlcul d'un valor normalitzat

$N_{inf} := (t - t^N) \text{div } t_m <--$ n. d'intervals amb inframostreig

```

si x = unknown llavors
     $T_u := T_u + t^N - t_{ant} + N_{inf} \cdot t_m$  <-- acumula temps desconegut
sino
     $x_i^N := A + x(t^N - t_{ant}) + x \cdot N_{inf} \cdot t_m$  <-- acumula valor
fsi

si  $T_u > \frac{(1+N_{inf})t_m}{2}$  llavors
     $x_i^N := \text{unknown}$  <-- normalitza desconegut
sino
     $x_i^N := \frac{A}{t_m(1 + N_{inf}) - T_u}$  <-- normalitza valor
fsi

mateix valor per cada interval amb inframostrig
 $x_{i+N_{inf}}^N := \dots := x_{i+1}^N := x_i^N$ 

si x = unknown llavors <-- acumula desconegut, reset àrea
     $T_u := t - t_{ant}$ 
     $A := 0$ 
sino <-- acumula àrea, reset desconeguts
     $T_u := 0$ 
     $A := x(t - t^N)$ 
fsi

 $t^N := t^N + (N_{inf} + 1)t_m$  <-- següent interval de normalització
i := i + (Ninf+1)

fsi

 $t_{ant} := t$ 

repeteix

```

5.3 Consolidació d'interval

Un cop s'ha normalitzat l'interval, a la base de dades hi ha els valors amb temps de mostreig regular (*step*). Aquest valors ja normalitzats a RRDtool s'anomenen *Primary Data Points* (PDP).

Però aquest interval de mostreig inicial no es pot mantenir durant gaire temps ja que cada cop és més difícil gestionar la quantitat de dades. Per afitar la quantitat de

dades emmagatzemades, RRDtool desa les dades en diferents resolucions temporals.

És a dir, cal consolidar l'interval normalitzat inicial a altres temps de mostreig desitjats. A partir dels PDP es tornen a mostrejar (*resampling*) nous intervals amb menys resolució.

A RRDtool, aquest valors nous consolidats s'anomenen *Consolidated Data Points* (CDP) i cada resolució s'emmagatzema en les taules anomenades *Round Robin Archive* (RRA). Cada CDP es calcula a partir de n PDP i cada RRA defineix aquest n necessari, el qual a RRDtool s'anomena *step* però que cal no confondre'l amb l'*step* de normalització d'interval. Es poden distingir com a *step de CDP* i *step de PDP*, respectivament.

En aquest càlcul pot interessar conservar diferent informació de les variables mesurades. El canvi de resolució implica una pèrdua de precisió i informació, cal trobar un càlcul que mantingui les propietats que interessin de les dades

Per un RRA que aplica una funció f de consolidació a n PDP (valors normalitzats V^N) per obtenir un CDP:

$$CDP_i = f(x_{i-n}^N, \dots, x_i^N) \quad i \in \{n, 2n, 3n \dots\}$$

És a dir que la resolució dels CDP sempre és múltiple del temps de mostreig base dels PDP. Actualment RRDTool permet calcular quatre funcions de consolidació:

- mitjana (AVERAGE)
- màxim (MAX)
- mínim (MIN)
- últim (LAST)

Calcular la mitjana és equivalent al que es fa a l'etapa de normalització de l'interval. En aquell cas cal fer una mitjana ponderada per la durada de la mesura però ara els intervals són iguals per tots els valors i per tant ja no cal considerar el temps.

$$CDP_i(\text{AVERAGE}) = \frac{\sum_{k=1}^N t_m \text{PDP}_k}{\sum_{k=1}^N t_m} = \frac{t_m \sum_{k=1}^N \text{PDP}_k}{nt_m} = \frac{\sum_{k=1}^N \text{PDP}_k}{n}$$

on n és el total de valors normalitzats que s'agafen, el qual s'ha anomenat *step de CDP*.

Per als càlculs de màxim, mínim i últim cal recordar que s'apliquen als PDP, és a dir un cop els valors mesurats han estat normalitzats. Per exemple en el cas del màxim no es conserva el valor més alt que s'ha mesurat sinó el valor més alt dels PDP; un cop s'ha fet la mitjana ponderada de les mesures.

Una altra lectura possible és que en un RRA que calculi cada CDP amb un PDP tant és quina funció s'apliqui; els CDP seran iguals als PDP.

5.3.1 Dades desconegudes

A la sortida de l'etapa de normalització, alguns intervals poden prendre el valor 'desconegut' (*unknown*). Les funcions de consolidació han de calcular amb aquests valors i per tant han de decidir com tractar-los.

Per una banda, quan l'etapa de normalització es troba valors desconeguts en un interval de mostreig, aproximació la normalització de manera que els valors desconeguts no alterin la mitjana dels valors coneguts. Ara a l'etapa de consolidació també s'aplica el mateix criteri per a la funció de consolidació de mitjana: com s'ha fet per $x_u = x_i^N$ a (5.6) a (5.7) ara s'aplica el mateix a $x_U = \text{CDP}_i$ i aleshores es pot escriure

$$\text{CDP}_i(\text{AVERAGE}) = \frac{\sum_{k=1}^N \text{PDP}_k}{n_{\text{coneguts}}} = \frac{x_u + \sum_{k=1}^N \text{PDP}_k}{n}$$

Però hi ha altres funcions i cal generalitzar el criteri. Per a les funcions de mínim i màxim es considera que els valors desconeguts no afecten en el càlcul i per a la funció d'últim el valor consolidat pren el que calgui, valor o 'desconegut'. Per tant, aquestes funcions no es veuen afectades per n .

Per altra banda, a l'etapa de normalització es considera un interval desconegut quan més de la meitat de l'interval de mostreig té valors desconeguts. Però ara, a l'etapa de consolidació, hi ha un paràmetre que permet decidir el percentatge de valors desconeguts que calen perquè el valor consolidat sigui considerat desconegut. Aquest paràmetre s'anomena xff (*xfiles factor*⁴).

$$N_{\text{desconeguts}}/N > \text{xff} \longrightarrow \text{CDP}_i = \text{unknown}$$

on l' xff s'expressa en tant per u.

A continuació es resumeix el tractament de desconeguts en un algoritme

Consolidació d'una sèrie temporal

cada període de consolidació segons els valors normalitzats

INPUT:

vector de valors normalitzats $\mathbf{PDP} = \mathbf{X}^N = [x_0^N, x_1^N, \dots, x_k^N]$

funció d'interpolació f

període de consolidació N en n . de PDP

llindar de desconeguts xff en tant per u

OUTPUT:

vector de valors consolidats $\mathbf{CDP} = [cdp_0, cdp_1, \dots, cdp_l]$

⁴RRDtool l'anomena el factor d'expedients X (*X-Files Factor*) perquè si té un valor diferent de zero el resultat està fora dels límits científics, [3]

```

i := 0
cdpi := unknown

a := 0 <-- acumulador pels interpoladors
Tu := 0 <-- Temps en valor desconegut
n := 1 <-- quantitat de PDP calculats

per cada p a PDP fes:

    si p = unknow llavors
        TU := Tu + 1 <-- acumula temps desconegut

    sino <-- aplica l'interpolador corresponent
        cas f és
            AVERAGE → a :=  $\frac{a(n-1) + p}{n}$ 
            MAX → a := max(a, p)
            MIN → a := min(a, p)
            LAST → a := p
        fcas
    fsi

    si n = N llavors <-- consolida

        si Tu / N > xff llavors
            cdpi := unknown <-- consolida desconegut
        sino
            cdpi ::= acumulat <-- consolida el valor
        fsi
        i := i+1
        n := 0
        a := 0
        Tu := 0
    fsi
    n := n + 1
repeteix

```

6 Model dels SGBD Round Robin

En aquest capítol es dissenya un model matemàtic per a les bases de dades Round Robin (RRD). Aquest model està inspirat en el sistema de gestió de bases de dades RRDtool i és el resultat de l'abstracció dels conceptes que s'han exposat als capítols 4 i 5.

A l'inici del capítol, s'introdueix el concepte de model matemàtic per a les bases de dades. A continuació, es defineixen els conceptes necessaris per a finalment definir una base de dades Round Robin. Al final del capítol, es resumeix breument la informació exposada.

6.1 Introducció

Segons Date, [8], “una base de dades és un contenidor informàtic per a una col·lecció de dades”. El sistema informàtic que tracten amb bases de dades s'anomenen sistema de gestió de bases de dades (SGBD) i tenen l'objectiu d'emmagatzemar informació i permetre consultar i afegir aquesta informació per part dels usuaris. Per complir aquests objectius, els SGBD ofereixen a l'usuari diferents operacions a fer amb la base de dades, com per exemple crear-la, afegir dades o consultar informació a partir de les dades emmagatzemades.

Els SGBD es basen en teories matemàtiques que reben el nom de model de dades, un SGBD és una implementació d'un model de dades. Segons Date, “un model de dades és una definició abstracta, auto continguda i lògica dels objectes, de les operacions i de la resta que conjuntament constitueixen la màquina abstracta amb la que els usuaris interactuen. Els objectes permeten modelar l'estructura de les dades. Les operacions permeten modelar el comportament”.

Les bases de dades Round Robin són bases de dades que contenen sèries temporals. Les sèries temporals són una col·lecció de dades mesurades en diferents instants de temps i necessiten un tractament adequat per part de la base de dades. El model de dades Round Robin és una solució d'emmagatzematge per a les sèries temporals que, resumint, consisteix a repartir la informació d'una sèrie temporal en intervals de temps diferents.

El model de dades Round Robin es dissenya per primera vegada en aquest capítol. Actualment existeix un SGBD, RRDtool, que implementa conceptes susceptibles d'esdevenir un model de dades però enlloc s'ha recollit com a tal. A partir de

l'anàlisi i l'abstracció profunda dels conceptes de RRDtool s'ha dissenyat el model de dades Round Robin, batejat segons el nom que té a RRDtool (*Round Robin Database tool*). Els conceptes de RRDtool s'han exposat detalladament als capítols 4 i 5.

Així doncs, a continuació es presenta el model de dades Round Robin. Es defineixen els objectes principals d'estudi: les *mesures* i les *sèries temporals*. Les *mesures* són dades mesurades en un instant de temps i les *sèries temporals* són col·leccions de mesures, a cada base de dades Round Robin hi ha emmagatzemada una sèrie temporal.

El model de dades Round Robin s'estructura a partir de *discs Round Robin*, els quals acumulen temporalment les *mesures* en un *buffer* per tal de tractar-les abans d'emmagatzemar-les a un *disc*. El tractament principal consisteix en canviar els intervals de temps entre *mesures* amb l'objectiu de compactar la informació de la sèrie temporal. Així doncs, la sèrie temporal queda emmagatzemada en intervals de temps diferents, repartits en els *discs Round Robin*.

Pel que fa a les operacions, és indispensable que el model Round Robin pugui fer aquests canvis d'intervals de temps, els quals s'aconsegueixen amb les operacions d'*interpolació* i *consolidació*. En el model de dades Round Robin d'aquest capítol també es defineixen operacions per crear una base de dades Round Robin, per inserir-hi mesures i per representar sèries temporals.

6.2 Mesura

Definició 1 (Mesura). *Definim mesura com el tuple (v, t) , en el que $v \in \mathbb{R}$ és el valor de la mesura i t és l'instant de temps en que s'ha pres aquesta mesura.*

Donada una mesura $m = (v, t)$ escriurem $V(m)$ per referir-nos a v i $T(m)$ per referir-nos a t .

Definició 2 (Temps). *Siguin t_i^a i t_j^a dos temps absoluts, observem un temps relatiu t^r com un valor $t^r \in \mathbb{R}$ que marca la distància en unitats de temps entre dos temps absoluts $t^r = t_i^a - t_j^a$.*

Sigui t^r un temps relatiu i t^R un temps absolut de referència, observem un temps absolut t^a com un valor $t^a \in \mathbb{R}$ que marca el temps relatiu respecte al temps de referència $t^a = t^R + t^r$. Aquest valor de referència $t^R \in \mathbb{R}$ és també un temps absolut però que permet definir unívocament la posició de qualsevol altre temps absolut.

Aquesta definició és coherent amb la d'un sistema de coordenades.

El temps absolut també s'anomena instant de temps i el temps relatiu, durada. En resum, els temps absoluts es poden veure com una seqüència de valors reals

que indiquen esdeveniments amb ordre clarament definit i entre dos temps absoluts sempre hi ha un temps relatiu que indica durada. Tant els temps absoluts com els temps relatius s'expressen amb un real que té unitats de temps, segons en sistema internacional.

Donades dues mesures és fàcil establir la relació d'ordre induïda pel temps.

Definició 3. *Sigui $m = (v_m, t_m)$ i $n = (v_n, t_n)$. Direm que $m \geq n$ si i solament si $t_m \geq t_n$.*

En la definició 1, el valor d'una mesura és un escalar. Això no obstant, és fàcil estendre el concepte a mesures multivaluades, que representen una col·lecció de valors mesurats en el mateix instant de temps tal i com es fa, per exemple a Åkfalg, [1].

Definició 4 (Mesura multivaluada). *Definim mesura com el tuple (v, t) , en el que $v \in \mathbb{R}^n$ és el valor de la mesura i t és l'instant de temps en que s'ha pres aquesta mesura.*

6.3 Sèrie temporal

Les sèries temporals són seqüències de mesures ordenades en el temps. Tradicionalment s'anomenen sèries temporals tot i que en algun lloc també s'anomenen seqüències temporals, per exemple al capítol 2 de Last, [19].

Definició 5 (Sèrie temporal). *Una sèrie temporal S és un conjunt de mesures $S = \{m_0, \dots, m_k\}$ tal que $\forall i, j : i \leq k, j \leq k, i \neq j : T(m_i) \neq T(m_j)$.*

La relació definida a 3 indueix sobre una sèrie temporal una relació d'ordre total. Com que la sèrie temporal s'ha considerat finita, això comporta l'existència d'un màxim i d'un mínim. Si S és una sèrie temporal, $\max(S)$ i $\min(S)$ són respectivament la mesura màxima i mínima d' S .

Definició 6. *Sigui $S = \{m_0, \dots, m_k\}$ una sèrie temporal i $n \in S$ una mesura. Direm que $n = \max(S)$ és el màxim de la sèrie temporal si $\forall m \in S : n \geq m$. Direm que $n = \min(S)$ és el mínim de la sèrie temporal si $\forall m \in S : n \leq m$.*

Atesa la relació d'ordre induïda pel temps en una sèrie temporal és possible definir el concepte d'interval sobre la seqüència, semblant a com es fa als capítols 1 i 2 de Last, [19].

Definició 7. *Sigui $S = \{m_0, \dots, m_k\}$ una sèrie temporal. Definirem la subsèrie $S(r, t] \subseteq S$ com la sèrie temporal $S(r, t] = \{m \in S | r < T(m) \leq t\}$, a on r i t són dos instants de temps.*

També es defineix la subsèrie $S(r, \infty) \subseteq S$ com la sèrie temporal $S(r, \infty) = S(r, T(\max(S)))$ i la subsèrie $S(-\infty, t) \subseteq S$ com la sèrie temporal $S(-\infty, t) = \{m \in S \mid T(\min(S)) \leq T(m) < t\}$.

Atesa la relació d'ordre induïda pel temps en una sèrie temporal es defineix el concepte de mesura següent i mesura anterior en una seqüència.

Definició 8. *Sigui $S = \{m_0, \dots, m_k\}$ una sèrie temporal i $n, l \in S$ dues mesures. Direm que l és el successor de m en S i ho notarem com $l = \text{seg}(m)$ si i només si $l = \min(S(T(m), \infty))$. Direm que l és el predecessor de m en S i ho notarem com $l = \text{ant}(m)$ si i només si $l = \max(S(-\infty, T(m)))$.*

Sigui $o = (\infty, \infty)$ i $p = (\infty, -\infty)$ dues mesures indefinides, en els casos que la mesura següent o anterior no existeix es defineixen següent i anterior corresponentment com $\text{seg}(\max S) = o$ i $\text{ant}(\min S) = p$.

Quan no hi hagi dubte de la sèrie temporal que marca l'ordre, escriurem $\text{seg}(m)$ i $\text{ant}(m)$.

6.3.1 Regularitat de les sèries temporals

Sigui $S = \{m_0, \dots, m_k\}$ una sèrie temporal, t un instant de temps i δ una durada de temps, en l'interval de temps $i_0 = [t, t + \delta]$ i els seus múltiples $i_j = [t + j\delta, t + (j + 1)\delta] : j = 0, 1, 2, \dots$, la sèrie temporal S és de naturalesa diferent segons la situació dels temps $T(m_i)$ en els intervals de temps i_j . En aquest context, aquests intervals de temps s'anomenen intervals de mostreig, δ s'anomena període de mostreig i t s'anomena temps d'inici del mostreig.

Una sèrie temporal és regular quan les mesures són equidistants en el temps, tal com ho anomenen al capítol 2 de Last, [19].

Definició 9 (Sèrie temporal regular). *Sigui $S = \{m_0, \dots, m_k\}$ una sèrie temporal, t un instant de temps i δ una durada de temps. Direm que S és regular si i només si $\forall m \in S(T(\min(S), \infty) : T(m) - T(\text{ant}(m)) = \delta$ i $T(\min(S)) = t$.*

Si una sèrie temporal és regular, l'anomenem sèrie temporal mostrejada regularment amb període de mostreig δ .

Una sèrie temporal és no regular quan no és regular.

6.3.2 Representació de sèries temporals

Keogh al capítol 1 de Last, [19], cita diverses representacions per les sèries temporals com per exemple *Fourier Transforms*, *Wavelets*, *Symbolic Mappings* o *Piecewise*

Linear Representation (PLR), però assenyala aquesta última com la representació més utilitzada. La PLR, funció definida a trossos lineal, és l'aproximació d'una sèrie temporal S , de llargada n , amb K segments rectes. Els segments podrien ser polinomis de qualsevol grau, però la manera més comuna de representar sèries temporals és amb funcions lineals, segons Keogh, [12]. Per aproximar el segment $S(t_a : t_b]$ d'una sèrie S , Keogh defineix dues tècniques: interpolació lineal, la recta que connecta t_a i t_b , i regressió lineal, la millor recta que aproxima per mínims quadrats el segment entre t_a i t_b .

Però també es pot representar una sèrie temporal amb una funció esglaó (*step* o *staircase function*); és a dir, amb una funció definida a trossos constant (*piecewise constant representation*). La representació a trossos constant és utilitzada en electrònica als convertidors digital-analògic (DAC, *digital-to-analog converter*). En aquest cas, un senyal discret es considera una sèrie temporal i per reconstruir el senyal continu típicament s'aplica el model de *zero-order hold*, equivalent a la representació a trossos constant, o el de *first-order hold*, equivalent a la representació a trossos lineal. El model de *zero-order hold* consisteix en mantenir constant cada valor fins al proper. S'obté una representació a trossos constant que en electrònica s'anomena seqüència de pulsos rectangulars (*rectangular pulses*).

A continuació, la representació d'una sèrie temporal segons el model de *zero-order hold* s'estén per diferents continuïtats en els intervals de temps de representació.

Sigui S una sèrie temporal, es defineix $S(t)$ com la representació de la sèrie temporal contínuament al llarg del temps t . En primer lloc, es representa amb *zero-order hold* a partir de funcions graó contínues per la dreta (*right-continuous*).

Definició 10 (Representació amb *zero-order hold*). *Sigui $S = \{m_0, \dots, m_k\}$ una sèrie temporal, la representació $S(t)$ amb zero-order hold es defineix*

$$\forall t \in \mathbb{R}, \forall m \in S : S(t) = \begin{cases} V(\min S) & \text{si } t < T(\min S) \\ V(m) & \text{si } t \in [T(m), T(\text{seg } m)) \end{cases}$$

En segon lloc, es representa $S(t)$ amb *zero-order hold* centrada en l'interval, definit també a partir de funcions graó contínues per la dreta.

Definició 11 (Representació amb *zero-order hold* centrada en l'interval). *Sigui $S = \{m_0, \dots, m_k\}$ una sèrie temporal, la representació $S(t)$ amb zero-order hold centrada en l'interval es defineix*

$$\forall t \in \mathbb{R}, \forall m \in S : S(t) = \begin{cases} V(m) & \text{si } t = \frac{T(\text{ant } m) + T(m)}{2} \\ V(m) & \text{si } t \in \left(\frac{T(\text{ant } m) + T(m)}{2}, \frac{T(m) + T(\text{seg } m)}{2} \right) \end{cases}$$

En tercer lloc, es representa $S(t)$ amb *zero-order hold* cap enrere, ara definit a partir de funcions graó contínues per l'esquerra.

Definició 12 (Representació en *zero-order hold* cap enrere). *Sigui $S = \{m_0, \dots, m_k\}$ una sèrie temporal, la representació $S(t)$ amb zero-order hold cap enrere es defineix*

$$\forall t \in \mathbb{R}, \forall m \in S : S(t) = \begin{cases} V(\max S) & \text{si } t > T(\max S) \\ V(m) & \text{si } t \in (T(\text{ant } m), T(m)] \end{cases}$$

Si S és una sèrie temporal regular i δ una durada de temps, aleshores la representació de $S(t)$ amb *zero-order hold* és la mateixa que la de $S(t - \delta)$ amb *zero-order hold* cap enrere i és la mateixa que la de $S(t - \frac{\delta}{2})$ centrada en l'interval.

6.4 Buffer

Un buffer és un contenidor d'una sèrie temporal, regular o no regular, que mitjançant una funció permet regularitzar aquesta sèrie temporal amb un període de mostreig constant. A l'acció de regularitzar un interval d'una sèrie temporal l'anomenarem consolidació, al període de mostreig contant l'anomenarem pas de consolidació i a la funció de regularització l'anomenarem interpolador.

Definició 13 (Buffer). *Definim buffer com el tuple (S, τ, δ, f) , en el que S és una sèrie temporal, τ és el darrer instant de temps de consolidació, δ és la durada del pas de consolidació i f és un interpolador.*

La consolidació d'una sèrie temporal s'inicia en un instant de temps concret i té lloc a cada pas de consolidació. Amb la finalitat d'establir els intervals de consolidació de la sèrie temporal, es defineix un buffer inicial.

Definició 14. *Definim buffer inicial o buffer buit com el buffer $B_0 = (\emptyset, t_0, \delta_0, f)$, el qual conté una sèrie temporal buida, l'instant de temps inicial de consolidació, una durada que indica el pas de consolidació i un interpolador.*

A partir del buffer buit es poden conèixer tots els instants de temps de consolidació del buffer, els quals seran $t_0 + k\delta, k \in \mathbb{N}$. Abans de consolidar, però, cal que la sèrie temporal contingui mesures. L'operació *afegeix* permet afegir una mesura a un buffer.

Definició 15. *L'operació afegeix afegeix una mesura a la sèrie temporal del buffer:*

$$\text{afegeix} : \text{Buffer} \times \text{Mesura} \longrightarrow \text{Buffer}$$

$$B \times m \longrightarrow B' = B \cup \{m\}$$

Cada cop que s'afegeix una mesura a un buffer es pot comprovar si el buffer ja és consolidable mitjançant un predicat que ens retorna un booleà: cert o fals.

Definició 16. *Un buffer és consolidable quan el temps d'una mesura de la sèrie temporal és més gran que el proper instant de temps de consolidació:*

$$\text{consolidable?} : \text{Buffer} \longrightarrow \text{Booleà}$$

Si sigui $B = (S, \tau, \delta, f)$ un buffer i $m = \max(S)$ la mesura màxima, B és consolidable si i només si $T(m) \geq \tau + \delta$

6.4.1 Interpolació

Si sigui S una sèrie temporal i T_0 i T_f dos instants de temps, un interpolador f calcula la mesura que resumeix a S en un interval de temps $i = [T_0, T_f]$.

$$f : \text{Sèrie temporal} \times \text{interval de temps} \longrightarrow \text{Mesura}$$

Hi poden haver diferents interpoladors depenent de com es vol resumir la sèrie temporal. A més, la representació de la sèrie temporal influeix en la manera d'interpolador. A continuació es defineixen alguns interpoladors per a sèries temporals representades amb *zero-order hold* cap enrere, a on l'interval de temps d'interpolació i s'interpreta continu per l'esquerra $(T_0, T_f]$.

Definició 17 (Interpolador mitjana aritmètica). *Si sigui $S = \{m_0, \dots, m_k\}$ una sèrie temporal, $S(t)$ la representació de la sèrie temporal amb zero-order hold cap enrere i $i = [T_0, T_f]$ un interval de temps, l'interpolador mitjana aritmètica (MA) resumeix $S(t)$ amb una mesura que és la mitjana dels valors de les mesures al conjunt $S(T_0, T_f]$.*

$$\text{MA} : \text{Sèrie temporal} \times \text{interval de temps} \longrightarrow \text{Mesura}$$

$$S = \{m_0, \dots, m_k\} \times i = [T_0, T_f] \longrightarrow m' = (v', T_f)$$

$$S' = S(T_0 : T_f] : v' = \frac{1}{|S'|} \sum_{\forall m \in S'} V(m)$$

Definició 18 (Interpolador màxim). *Si sigui $S = \{m_0, \dots, m_k\}$ una sèrie temporal, $S(t)$ la representació de la sèrie temporal amb zero-order hold cap enrere i $i = (T_0, T_f]$ un interval de temps, l'interpolador màxim (MAX) resumeix $S(t)$ amb una mesura que és el màxim dels valors de les mesures al conjunt $S(T_0, T_f]$.*

$$\text{MAX} : \text{Sèrie temporal} \times \text{interval de temps} \longrightarrow \text{Mesura}$$

$$S = \{m_0, \dots, m_k\} \times i = [T_0, T_f] \longrightarrow m' = (v', T_f)$$

$$S' = S(T_0 : T_f] : v' = \max_{\forall m \in S'} (V(m))$$

De manera dual es pot definir l'interpolador mínim (MIN) com el que resumeix $S(t)$ amb una mesura que és el mínim dels valors de les mesures al conjunt $S(T_0, T_f]$.

Definició 19 (Interpolador últim). Sigui $S = \{m_0, \dots, m_k\}$ una sèrie temporal, $S(t)$ la representació de la sèrie temporal amb zero-order hold cap enrere i $i = (T_0, T_f]$ un interval de temps, l'interpolador últim (LAST) resumeix $S(t)$ amb una mesura que és la mesura màxima del conjunt $S(T_0, T_f]$.

LAST : Sèrie temporal \times interval de temps \longrightarrow Mesura

$$S = \{m_0, \dots, m_k\} \times i = [T_0, T_f] \longrightarrow m' = (\max(S(T_0 : T_f]), T_f)$$

Definició 20 (Interpolador àrea). Sigui $S = \{m_0, \dots, m_k\}$ una sèrie temporal, $S(t)$ la representació de la sèrie temporal amb zero-order hold cap enrere i $i = (T_0, T_f]$ un interval de temps, l'interpolador àrea (AREA) resumeix $S(t)$ amb una mesura que conserva l'àrea sota la corba de la representació zero-order hold cap enrere del conjunt $S(T_0, T_f]$.

AREA : Sèrie temporal \times interval de temps \longrightarrow Mesura

$$S = \{m_0, \dots, m_k\} \times i = [T_0, T_f] \longrightarrow m' = (v', T_f)$$

$$v' = \frac{\int_{T_0}^{T_f} S(t) dt}{T_f - T_0}$$

per $S(t)$ estar definida a trossos, v' es pot expressar com

$$n = \min(S(T_f, \infty), o = \min(S(T_0 : T_f]), S' = S(T_0 : T_f] - \{o\} :$$

$$: v' = \frac{(T(o) - T_0)V(o) + (T_f - T(\text{ant}_S n))V(n) + \sum_{\forall m \in S'} (T(m) - T(\text{ant}_S m))V(m)}{T_f - T_0}$$

6.4.2 Consolidació

Quan un buffer és consolidable, es pot calcular una mesura de consolidació de la sèrie temporal per cada interval de temps consolidable. De manera simplificada, a cada consolidació només es té en compte l'interval que comença al darrer temps de consolidació del buffer.

Sigui $B = (S, \tau, \delta, f)$ un buffer consolidable, la mesura de consolidació de B en l'interval de temps $i = [\tau, \tau + \delta]$ és $m' = (v, \tau + \delta)$ on $m' = f(S, i)$ i f és un interpolador. L'operació *consolida* permet consolidar la sèrie temporal del buffer calculant-ne la mesura de consolidació.

Definició 21. L'operació *consolida* calcula la mesura de consolidació i treu les mesures consolidades de la sèrie temporal del buffer, en l'interval de consolidació actual:

consolida : Buffer \longrightarrow Buffer \times Mesura

$$B = (S, \tau, \delta, f) \longrightarrow B' \times m'$$

$$B' = (S', \tau + \delta, \delta, f)$$

$$S' = S(\tau + \delta, \infty)$$

$$m' = f(S, [\tau, \tau + \delta]) : f \text{ és un interpolador}$$

6.5 Disc

Un disc és un contenidor d'una sèrie temporal regular amb un nombre acotat de mesures. En arribar al nombre màxim de mesures permeses, cada cop que s'afegeix una mesura nova s'elimina la mesura mínima de la sèrie temporal. Així doncs, un disc és semblant a una cua FIFO (*First In First Out*), a on el primer d'arribar és el primer de sortir.

Definició 22 (Disc). *Definim disc com el tuple (S, k) , en el que S és una sèrie temporal i $k \in \mathbb{N}$ és el cardinal màxim de S .*

A l'inici, un disc no conté mesures però cal que estigui caracteritzat pel cardinal màxim. Amb aquesta finalitat es defineix un disc inicial.

Definició 23. *Definim disc inicial o disc buit com el disc $D_\emptyset = (\emptyset, k)$, el qual conté una sèrie temporal buida i el cardinal màxim que podrà prendre S .*

L'operació *afegeix* permet afegir una mesura a un disc, controlant-ne el cardinal màxim.

Definició 24. *L'operació afegeix afegeix una mesura a la sèrie temporal del disc:*

$$\text{afegeix} : \text{Disc} \times \text{Mesura} \longrightarrow \text{Disc}$$

$$D = (S, k) \times m \longrightarrow D' = (S', k)$$

$$S' = \begin{cases} S \cup \{m\} & \text{si } |S| < k \\ (S - \{\min(S)\}) \cup \{m\} & \end{cases}$$

6.6 Disc Round Robin

Un disc Round Robin és un disc amb buffer. En el buffer hi ha la part d'una sèrie temporal a regularitzar i en el disc hi ha l'altra part ja regularitzada, amb un nombre acotat de mesures.

Definició 25 (Disc Round Robin). *Definim disc Round Robin com el tuple (B, D) , en el que B és un buffer i D és un disc.*

Per una banda, la definició de buffer buit (def. 14) i de disc buit (def. 23) indueixen a una definició de Disc Round Robin buit.

Definició 26. *Definim disc Round Robin buit com el disc Round Robin $R_\emptyset = (B_\emptyset, D_\emptyset)$, el qual conté un buffer buit i un disc buit.*

Per altra banda, les operacions dels buffers i dels discs estan relacionades amb les operacions dels discs Round Robin.

L'operació *afegeix* permet afegir una mesura a un disc Round Robin.

Definició 27. *L'operació afegeix afegeix una mesura al buffer del disc Round Robin:*

$$\text{afegir} : \text{Disc Round Robin} \times \text{Mesura} \longrightarrow \text{Disc Round Robin}$$

$$R = (B, D) \times m \longrightarrow R' = (B', D)$$

$$B' = B \text{ afegeix } m$$

Cada cop que s'afegeix una mesura a un disc Round Robin es pot comprovar si ja és consolidable.

Definició 28. *Un disc Round Robin és consolidable quan el seu buffer és consolidable:*

$$\text{consolidable?} : \text{Disc Round Robin} \longrightarrow \text{Booleà}$$

Sigui $R = (B, D)$ un disc Round Robin, R és consolidable si i només si B és consolidable.

Quan un disc Round Robin és consolidable, es pot consolidar amb l'operació *consolida*.

Definició 29. *L'operació consolida calcula una mesura de consolidació del buffer, en l'interval de consolidació actual, i la desa al disc.*

$$\text{consolida} : \text{Disc Round Robin} \longrightarrow \text{Disc Round Robin}$$

$$R = (B, D) \longrightarrow R' = (B', D')$$

$$B' \times m' = \text{consolida } B$$

$$D' = D \text{ afegeix } m'$$

6.7 Base de dades Round Robin

Una base de dades Round Robin és un arxiu de discs Round Robin que comparteixen l'entrada de mesures, les quals provenen d'una sèrie temporal. En aquesta entrada de mesures hi pot haver un buffer d'entrada comú per tal que els discs Round Robin rebin la sèrie temporal regularitzada. Amb buffer d'entrada o sense, la sèrie temporal queda regularitzada i distribuïda en els diferents discs Round Robin. A la figura 6.1 es representa esquemàticament una base de dades Round Robin.

Definició 30 (Base de dades Round Robin). *Definim base de dades Round Robin com el tuple (B, A) , en el que B és un buffer i A és un conjunt de discs Round Robin $A = \{R_0, \dots, R_d\}$.*

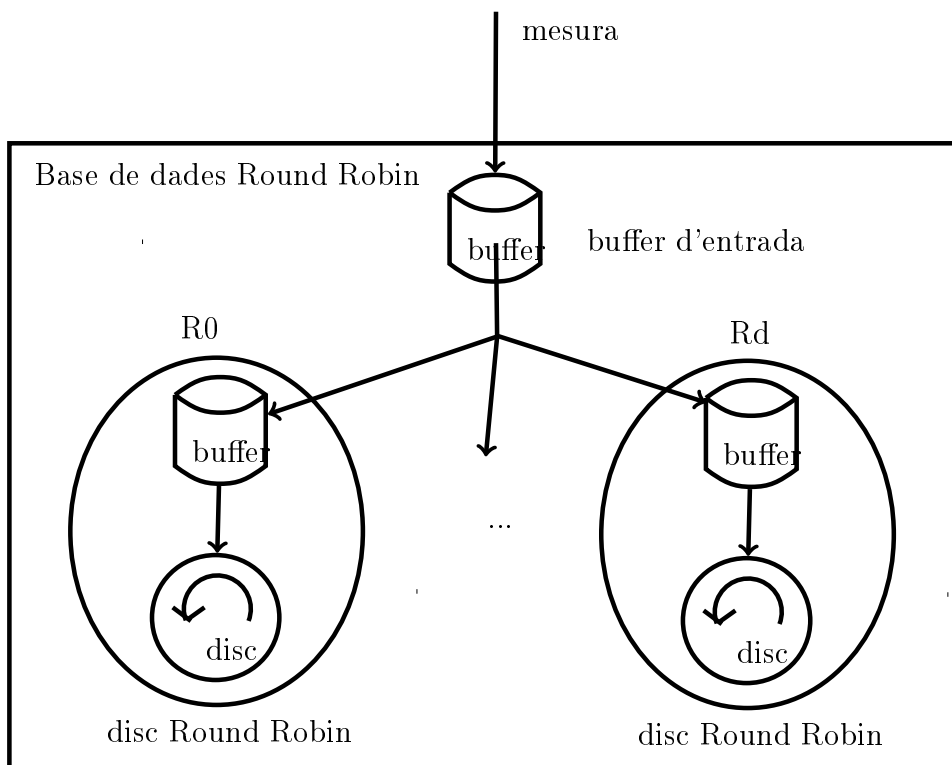


Figura 6.1: Esquema del model de dades Round Robin

Per una banda, aquest conjunt de discs Round Robin cadascun amb la seva definició de buit (def 26) i la definició de buffer buit (def 14) pel buffer comú d'entrada, indueixen a una definició de base de dades Round Robin buida.

Definició 31. *Definim base de dades Round Robin buida com la base de dades Round Robin $M_0 = (B_0, A_0)$, la qual conté un buffer buit i un arxiu amb quantitat d de discs Round Robin $A_0 = \{R_0, \dots, R_d\}$ on cada disc Round Robin és buit $R_i = R_0$.*

Normalment, en el conjunt de discs Round Robin A_0 no hi ha elements repetits amb la mateixa informació. És a dir, donats dos discs Round Robin $R_a = (B_a, D_a)$ i $R_b = (B_b, D_b)$, els seus respectius buffers $B_a = (S_a, \tau_a, \delta_a, f_a)$ i $B_b = (S_b, \tau_b, \delta_b, f_b)$ no tenen el mateix interval de consolidació i interpolador: $\delta_a \neq \delta_b \wedge f_a \neq f_b$.

Per altra banda, les operacions del buffer d'entrada i dels discs Round Robin estan relacionades amb les operacions de la base de dades Round Robin.

L'operació *afegeix* permet afegir una mesura a una base de dades Round Robin. Concretament s'afegeix al buffer d'entrada comú.

Definició 32. *L'operació afegeix afegeix una mesura a la base de dades Round Robin:*

afegeix : Base de dades Round Robin \times Mesura \longrightarrow Base de dades Round Robin

$$M = (B, A) \times m \longrightarrow M' = (B', A)$$

$$B' = B \text{ afegeix } m$$

Cada cop que s'afegeix una mesura a una base de dades Round Robin es pot comprovar si ja és consolidable.

Definició 33. *Una base de dades Round Robin és consolidable quan el seu buffer és consolidable:*

$$\text{consolidable?} : \text{Base de dades Round Robin} \longrightarrow \text{Booleà}$$

Si sigui $M = (B, A)$ una base de dades Round Robin, M és consolidable si i només si B és consolidable.

Quan una base de dades Round Robin és consolidable, es pot consolidar amb l'operació *consolida*.

Definició 34. *L'operació consolida calcula una mesura de consolidació del buffer, en l'interval de consolidació actual, i la desca a tots els disc Round Robin de l'arxiu.*

consolida : Base de dades Round Robin \longrightarrow Base de dades Round Robin

$$M = (B, A), A = \{R_0, \dots, R_d\} \longrightarrow M' = (B', A')$$

$$B' \times m' = \text{consolida } B$$

$$A' = \forall R_i \in A : R_i \text{ afegeix } m'$$

Finalment, les operacions de consolidació dels disc Round Robin també tenen la relació corresponent a la base de dades Round Robin. Aquestes consolidacions es poden veure com que els discs 'roden' i per tant les operacions s'anomenen amb aquesta arrel.

Cada cop que es consolida una base de dades Round Robin es pot comprovar si ja és rodable.

Definició 35. Una base de dades Round Robin és rodable quan hi ha un disc Round Robin consolidable.

$$\text{rodable?} : \text{Base de dades Round Robin} \longrightarrow \text{Booleà}$$

Sigui $M = (B, A)$ una base de dades Round Robin amb l'arxiu A com un conjunt de discs Round Robin $A = \{R_0, \dots, R_d\}$, M és rodable si i només si $\exists R_i \in A$ tal que R_i és consolidable.

Quan una base de dades Round Robin és rodable, es pot rodar amb l'operació *roda*.

Definició 36. L'operació *roda* calcula una mesura de consolidació del buffer, en l'interval de consolidació actual, i la desa a tots els disc Round Robin de l'arxiu.

$$\text{roda} : \text{Base de dades Round Robin} \longrightarrow \text{Base de dades Round Robin}$$

$$M = (B, A), A = \{R_0, \dots, R_d\} \longrightarrow M' = (B, A')$$

$$A' = \{R'_0, \dots, R'_{|A|}\} : \forall R_i \in A : R'_i = \begin{cases} \text{consolida } R_i & \text{si } R_i \text{ és consolidable} \\ R_i & \end{cases}$$

6.8 Resum

Aquest capítol s'acaba amb un resum dels conceptes exposats en el model de dades Round Robin. Una base de dades Round Robin és un sistema informàtic d'emmagatzematge d'una sèrie temporal entesa com una col·lecció de dades mesurades en diferents instants de temps.

A la base de dades, la sèrie temporal queda estructurada com s'ha esquematitzat a la figura 6.1. És una forma compacta d'emmagatzemar la sèrie temporal de manera que queda repartida segons diferents funcions d'interpolació i períodes de mostreig. Aquest repartiment té lloc en els diferents discs Round Robin, els quals fan ús del seu buffer per interpolar les mesures i fan ús del seu disc per consolidar-les.

El conjunt de discs Round Robin constitueixen la part principal d'una base de dades Round Robin tot i que, com a l'esquema, hi pot haver un buffer d'entrada de mesures comú per tal de regularitzar la sèrie temporal des d'un principi, ja que el pas de no regular a regular requereix interpoladors més complicats.

En el capítol 7, utilitzant el llenguatge de programació Python es dissenya, a nivell acadèmic, un sistema de gestió de bases de dades que implementa el model de dades Round Robin tal com s'ha definit en aquest capítol.

En resum, a partir del model de dades Round Robin descrit en aquest capítol per una banda es poden estudiar quin efecte té una configuració determinada de paràmetres i per altra banda es poden dissenyar sistemes de gestió de bases de dades assegurant que implementen el model i per tant que tenen el funcionament desitjat.

7 Implementació de referència del model RRD

En aquest capítol es dissenya, a nivell acadèmic, amb llenguatge de programació Python un sistema de gestió de bases de dades (SGBD) que implementa el model de dades Round Robin (RRD) tal com s'ha definit al capítol 6.

Aquest capítol s'estructura de manera semblant al capítol anterior, amb les mateixes seccions de la 6.2 a la 6.7, de manera que una secció d'aquest capítol és la implementació en Python de la definició del model en la mateixa secció del capítol anterior.

Al final s'experimenta amb un exemple de funcionament de la base de dades RRD completa i s'extreuen unes conclusions breus d'aquesta implementació en Python.

En aquest capítol se suposen coneixements bàsics de Python per a poder entendre la implementació en aquest llenguatge.

7.1 Introducció

Python és un llenguatge d'alt nivell i això permet que la implementació sigui quasi bé idèntica a les definicions del model. Python té una sintaxi senzilla que fa que sigui molt llegible, fins i tot per a qui no coneix Python. Tot i així, calen uns coneixements bàsics de programació.

Python és un llenguatge orientat a objectes. El model s'implementa utilitzant el paradigma de programació d'orientació a objectes. És a dir, cada objecte del model s'implementa en una classe, amb els atributs i mètodes corresponents. Hi ha un ús intens de mètodes especials de Python, com a referència es pot consultar la documentació de Python [27]. A les classes s'implementa el mètode especial de representació per poder obtenir una visualització en text de les instàncies, no s'ha de confondre amb la representació de sèries temporals descrita al model.

El temps està implementat com a temps relatiu, per a més completesa s'hauria de dissenyar un objecte `Temps`. Com a conseqüència d'aquesta simplificació, el SGBD resultant inicialitza la base de dades al temps zero i els temps es consideren relatius a aquest instant d'inicialització.

7.2 Mesura

La mesura s'ha implementat a la classe `Mesura` (Listing 7.1).

Una `Mesura` té els atributs `v` pel valor de la mesura i `t` pel temps. Donada una mesura `m` es pot accedir als atributs de manera que `m.v` correspon a la notació $V(m)$ i `m.t` correspon a $T(m)$, com es pot veure a l'exemple següent:

```
>>> m = Mesura(1,2)
>>> m.v
1
>>> m.t
2
```

Per implementar la relació d'ordre induïda pel temps s'utilitzen els mètodes especials pels operadors d'igualtat i desigualtat. Concretament, per tenir la relació d'ordre completa cal implementar la igualtat, la no igualtat i el major.

Aleshores es poden comparar dues mesures:

```
>>> m1 = Mesura(1,2)
>>> m2 = Mesura(1,3)
>>> m2 > m1
True
```

7.3 Sèrie temporal

La sèrie temporal s'ha implementat a la classe `SerieTemporal` (Listing 7.2). Aquesta classe només té un mètode ja que es defineix com una subclasse de `set`, d'on hereta la resta d'atributs i mètodes.

Els `set` són un dels tipus de dades bàsics que té Python. A la documentació de `set`, [27], es defineixen com una col·lecció desordenada d'elements no repetits.

Les sèries temporals s'implementen com a subclasse de `set` perquè aquests ja tenen implementades les operacions matemàtiques habituals de conjunts: pertinença, unió, intersecció, diferència o diferència simètrica.

No obstant, els `set` no tenen ordre i en canvi les sèries temporals sí que en tenen i per tant es voldrien implementar com a seqüències. Els tipus amb ordre a Python s'anomenen *sequence* però aquests tipus no són conjunts.

En resum, es voldria un `OrderedSet` però oficialment a Python (v2.7) no existeix. Així doncs, s'ha optat per mantenir les sèries temporals com a subclasse de `set` i implementar els mètodes que tenen les seqüències. Pel que fa a l'ordre, tot i que els `set` no en tenen les mesures sí i per tant les operacions de màxim i mínim

Listing 7.1: Implementació de la classe Mesura

```

1 class Mesura(object):
2     """
3     Mesura m = (v,t) on v és el valor en el temps t.
4     """
5     def __init__(self,v,t):
6         """
7         Constructor d'una mesura
8         """
9         self.v = v
10        self.t = t
11
12    def __gt__(self,other):
13        """
14        Relació d'ordre induïda pel temps
15        """
16        return self.t > other.t
17
18    def __eq__(self,other):
19        """
20        Relació d'ordre induïda pel temps
21        """
22        return isinstance(other,Mesura) and self.t == other.t
23
24    def __ne__(self,other):
25        """
26        Relació d'ordre induïda pel temps
27        """
28        return not self == other
29
30    def __hash__(self):
31        """
32        Retorna el hash d'una mesura, necessari per poder pertànyer a objectes ↘
33        → col·leccions com per exemple els sets
34        """
35        return hash(self.t)
36
37    def __repr__(self):
38        return 'm({0},{1})'.format(self.v,self.t)

```

Listing 7.2: Implementació de la classe SerieTemporal

```

1 class SerieTemporal(set):
2     """
3     Sèrie temporal  $s = \{m_0, \dots, m_k\}$  com una seqüència de mesures ordenades en
4     → el temps
5     """
6     def __getitem__(self, key):
7         """
8         Definició d'element i interval d'elements en una sèrie temporal.
9         """
10        #mesura màxima i mínima de la sèrie temporal
11        minm = min(self)
12        maxm = max(self)
13
14        #interval d'elements
15        if isinstance(key, slice):
16            l = key.start
17            g = key.stop
18            p = key.step
19
20            #Per definició,  $S(r, \text{infinit})$  és  $S(r, T(\max(S)))$ 
21            if g is None:
22                g = maxm.t
23
24            s = SerieTemporal()
25            for m in self:
26                if m.t > l and m.t <= g:
27                    s.add(m)
28            return s
29
30        #element
31        else:
32            l = key
33            g = key
34
35            if l < minm.t or g > maxm.t:
36                raise IndexError
37            for m in self:
38                if m.t == key:
39                    return m
40            raise KeyError

```

retornaran el valor correcte del conjunt.

Així doncs, l'única definició que cal implementar és la d'interval sobre la seqüència. S'utilitza el mètode especial de les seqüències d'accés a elements, en el que també es pot sol·licitar intervals de la seqüència (anomenats *slices* a Python).

Donada una sèrie temporal s , es pot escriure l'*slice* $s[r:t]$ que correspon a la definició $S(r, t]$ on r i t són dos temps.

Els *slice* també permeten les sintaxis $s[r:]$, $s[:t]$ i $s[:]$ que respectivament es fan correspondre a les definicions $S(r, \infty)$, $S(\infty, t]$ i $S(\infty, \infty)$.

Per altra banda, per a la implementació a Python de l'accés a element cal definir com s'obté un element en concret, el qual s'escriu amb la sintaxi $s[i]$ on i és un enter. Aquest accés a element s'ha fet correspondre amb donada una sèrie temporal S i un temps i , $s(i) = m \in S : T(m) = i$; tot i que no encaixa bé amb el concepte de Python on i és un enter que indica la posició dins la seqüència.

7.4 Buffer

Un buffer s'ha implementat a la classe `Buffer` (Listing 7.3). Aquesta classe té quatre atributs que es corresponen amb s és una sèrie temporal, τ és el darrer temps de consolidació, δ és el pas de consolidació i f és un interpolador. Tot seguit es detalla l'ús d'aquests atributs.

Per a construir un buffer buit es necessita com a mínim dos paràmetres: δ , el pas de consolidació, i f , l'interpolador associat al buffer. És a dir, aquests atributs són els que caracteritzen a un buffer i estan pensats per a ser constants un cop s'ha construït un buffer.

Pel que fa als dos altres atributs, la sèrie temporal s s'inicia com un conjunt buit i el darrer temps de consolidació τ correspon al temps inicial, de moment implementat com a relatiu i per tant temps zero.

Aleshores es defineixen els tres mètodes que es corresponen amb les operacions que tenen els buffers: *afegeix*, *consolidable* i *consolida*.

El mètode *afegeix* necessita una mesura com a paràmetre per tal d'afegir-la a la sèrie temporal.

En el mètode *consolidable* primer es comprova que s'hagi afegit algun valor a la sèrie temporal i en cas afirmatiu es cerca si la mesura més recent ha superat l'interval de consolidació, calculat com el pas de consolidació des del darrer temps de consolidació del buffer. En aquest cas es considera que el buffer pot ser consolidat.

Aleshores es pot cridar el mètode *consolida* per a consolidar el buffer. Aquest extreu de la sèrie temporal les mesures que cauen dins de l'interval de consolidació, situa el nou interval de consolidació i retorna una mesura que resumeix l'interval

Listing 7.3: Implementació de la classe Buffer

```

1 class Buffer(object):
2     """
3     Buffer B = (S,tau,delta,f) on S és una sèrie temporal, tau el darrer temps
4     de consolidació, delta el pas de consolidació i f un interpolador
5     """
6     def __init__(self,delta,f):
7         """
8         Constructor d'un Buffer buit
9         """
10        self.s = SerieTemporal(); self.tau = 0; self.delta = delta; self.f = f
11
12    def afegeix(self,m):
13        """
14        Definició de l'operació afegeix
15        """
16        self.s.add(m)
17
18    def consolidable(self):
19        """
20        Definició del predicat consolidable
21        """
22        if self.s:
23            m = max(self.s)
24            return m.t >= (self.tau + self.delta)
25        return False
26
27    def consolida(self):
28        """
29        Definició de l'operació consolida
30        """
31        noutau = self.tau + self.delta
32        interval = (self.tau,noutau)
33        interpola = self.f(self.s, interval)
34
35        self.s = self.s[noutau:]
36        self.tau = noutau
37        self.delta = self.delta
38        return interpola
39
40    def __repr__(self):
41        return 'Buffer({0},{1},{2})'.format(self.s,self.tau,self.delta)

```

Listing 7.4: Implementació de l'interpolador mitjana

```

1 def mitjana(s,i):
2     """
3     Interpolador mitjana d'una sèrie temporal s en l'interval i.
4     """
5     sp = s[i[0]:i[1]]
6     v = 0
7     for m in sp:
8         v += m.v
9     v /= float(len(sp))
10    return Mesura(v,i[1])

```

consolidat calculada per l'interpolador associat al buffer.

7.4.1 Interpolador mitjana

Un interpolador s'implementa com una funció que com a paràmetres necessita una sèrie temporal i l'interval en el que es vol interpolat. Aquest interval s'implementa com una tupla de dos elements.

L'interpolador opera amb aquests dos paràmetres, de la manera que creu convenient, i retorna un objecte `Mesura` que resumeix la sèrie temporal a l'interval demanat.

És a dir, hi pot haver més d'un interpolador diferent on cadascun es caracteritza per la manera d'operar amb la sèrie temporal. Com a exemple s'implementa l'interpolador mitjana, el qual resumeix la sèrie temporal amb la mesura mitjana en l'interval demanat.

L'interpolador mitjana (Listing 7.4) s'ha implementat com una funció que calcula les mesures que cauen dins de l'interval `i` i en fa la mitjana dels seus valors.

En aquesta implementació no s'ha tingut en compte que les mesures siguin multivaluades, però es podria modificar per tal que s'admetessin. Aleshores la funció retornaria una mesura multivaluada amb una mitjana per cada multivalor.

7.5 Disc

Un disc s'ha implementat a la classe `Disc` (Listing 7.5). Aquesta classe té dos atributs: `s` és una sèrie temporal i `k` és el cardinal màxim que pot prendre la sèrie temporal.

Així doncs, el cardinal màxim és el paràmetre que caracteritza cada disc i es ne-

Listing 7.5: Implementació de la classe Disc

```

1 class Disc(object):
2     """
3     Disc D = (S,k) on S és una sèrie temporal i k és el cardinal màxim de S
4     """
5     def __init__(self,k):
6         """
7         Constructor d'un Disc buit
8         """
9         self.s = SerieTemporal()
10        self.k = k
11
12    def afegeix(self,m):
13        """
14        Definició de l'operació `afegeix`
15        """
16        if len(self.s) < self.k:
17            self.s.add(m)
18        else:
19            smin = SerieTemporal()
20            smin.add( min(self.s) )
21
22            self.s -= smin
23            self.s.add(m)
24
25    def __repr__(self):
26        return 'Disc({0},_{1})'.format(self.s,self.k)

```

cessita en construir un disc buit. La sèrie temporal s'inicialitza com un conjunt buit.

Un disc només té el mètode *afegeix* que necessita una mesura com a paràmetre per tal d'afegir-la a la sèrie temporal. En aquest mètode es té en compte que si el cardinal de la sèrie temporal és igual a l'atribut *k* llavors per afegir una nova mesura abans cal treure la mesura mínima.

Per a operar amb la sèrie temporal s'utilitzen les operacions de conjunt. Concretament s'utilitza el mètode *add* per afegir objectes al conjunt i l'operador diferència, el qual necessita que la mesura que es vol treure formi part d'un conjunt que en aquest cas s'anomena *smin*.

7.6 Disc Round Robin

Un disc Round Robin s'ha implementat a la classe `DiscRoundRobin` (Listing 7.6). Aquesta classe té dos atributs: *B* és un buffer i *D* és un disc.

A l'hora de construir un disc round robin cal crear un buffer i un disc buit, per tant aquests dos últims marquen els paràmetres necessaris pel constructor. Són tres paràmetres: el pas de consolidació *delta* i la funció d'interpolació *f* del buffer, i el cardinal màxim *k* del disc. Així doncs, els disc round robin estan caracteritzats per aquests tres paràmetres, els quals estan pensats per ser constants des de la creació.

Els mètodes implementats per un `DiscRoundRobin` també recorden als buffers i discs: *afegeix* que afegeix una mesura al buffer, *consolidable* que consulta si el buffer és consolidable i *consolida* que consolida el buffer i afegeix la mesura interpolada del buffer al disc.

7.7 Base de dades Round Robin (RRD)

Una base de dades Round Robin (RRD) s'ha implementat a la classe `RRD` (Listing 7.7). Aquesta classe té dos atributs: *B* és un buffer i *A* és un arxiu dels `DiscRoundRobin` implementat com una llista (collecció d'elements).

Una RRD es crea amb un buffer i amb un arxiu buit de discs Round Robin. Per tant, els paràmetres necessaris del constructor corresponen a a aquest buffer i són el pas de consolidació *delta* i la funció d'interpolació *f*.

Per tal de poder afegir els `DiscRoundRobin` a la RRD, s'implementa el mètode *afegeix_disc* que donats els paràmetres que caracteritzen a un `DiscRoundRobin` en crea un i l'afegeix a l'arxiu.

Aleshores, la RRD es caracteritza pels paràmetres del buffer i dels `DiscRoundRobin` que té. Per a treballar amb aquests buffers i disc, la `RRD` implementa mètodes

Listing 7.6: Implementació de la classe DiscRoundRobin

```

1 class DiscRoundRobin(object):
2     """
3     Disc Round Robin R = (B,D) on B és un buffer i D és un disc
4     """
5     def __init__(self,delta,k,f):
6         """
7         Constructor d'un Disc Round Robin buit
8         """
9         self.B = Buffer(delta,f)
10        self.D = Disc(k)
11
12    def afegeix(self,m):
13        """
14        Definició de l'operació `afegeix`
15        """
16        self.B.afegeix(m)
17
18    def consolidable(self):
19        """
20        Definició del predicat `consolidable`
21        """
22        return self.B.consolidable()
23
24    def consolida(self):
25        """
26        Definició de l'operació `consolida`
27        """
28        m = self.B.consolida()
29        self.D.afegeix(m)
30
31    def __repr__(self):
32        return 'Disc({0},_{1})'.format(self.s,self.k)

```

Listing 7.7: Implementació de la classe RRD

```

1 class RRD(object):
2     """
3     M = (B,A) una base de dades Round Robin on B és un buffer i A és un
4     →conjunt de discs Round Robin
5     """
6     def __init__(self,delta,f):
7         self.B = Buffer(delta,f)
8         self.A = []
9
10    def afegeix_disc(self,delta,k,f):
11        self.A.append( DiscRoundRobin(delta,k,f) )
12
13    def update(self,m):
14        self.B.afegeix(m)
15
16    def consolidable(self):
17        return self.B.consolidable()
18
19    def consolidar(self):
20        m = self.B.consolidada()
21        for R in self.A:
22            R.afegeix(m)
23
24    def rodable(self):
25        for R in self.A:
26            if R.consolidable():
27                return True
28        return False
29
30    def roda(self):
31        for R in self.A:
32            if R.consolidable():
33                R.consolidada()

```

semblants als de buffers i discs.

Per una banda, `consolidable` consulta si el buffer de la RRD és consolidable i en cas afirmatiu es pot cridar el mètode `consolidar`, el qual consolida aquest buffer i afegeix la mesura resultant de la consolidació del buffer a tots els `DiscRoundRobin` que hi ha a l'arxiu.

Per altra banda, `rodable` consulta si hi ha algun `DiscRoundRobin` que sigui consolidable i en cas afirmatiu es pot cridar el mètode `roda`, el qual consolida tots els `DiscRoundRobin` de l'arxiu que siguin consolidables.

7.8 Experiment de funcionament

A continuació es detalla un exemple de funcionament del SGBD Round Robin dissenyat. En un intèrpret de Python es treballa amb una base de dades RRD.

A l'exemple del Listing 7.8 es pot veure un exemple d'ús de tots els mètodes d'una RRD.

En primer lloc es crea una base de dades RRD, amb el buffer d'entrada amb període de mostreig de 5 i interpolador mitjana. En segon lloc es crea un disc Round Robin amb període de mostreig 10, cardinal màxim 2 i interpolador mitjana. En tercer lloc es creen quatre mesures que s'utilitzaran més endavant, aquestes mesures simulen valors en els temps 1, 2, 5 i 10.

En quart lloc, s'insereix la primera mesura i s'observa que la base de dades encara no és consolidable. A continuació, s'insereix la segona mesura i la base de dades encara no és consolidable. Quan s'insereix la tercera mesura, la base de dades ja és consolidable perquè s'ha completat el primer interval de consolidació (0, 5]. S'insereix la quarta mesura i la base de dades segueix sent consolidable ja que encara hi ha pendent de consolidar l'interval (0, 5].

En cinquè lloc, es consolida la base de dades amb la mesura calculada per l'interpolador mitjana: $\frac{10+10+40}{3} = 20$. La base de dades no és rodable però segueix sent consolidable, ara en l'interval (5, 10] degut a la quarta mesura inserida.

En sisè lloc, es torna a consolidar la base de dades, ara en l'interval (5, 10] amb la mesura calculada per l'interpolador mitjana: $\frac{50}{1} = 50$. La base de dades ja no és consolidable però sí que és rodable perquè s'ha completat el primer interval de consolidació (0, 10] del disc Round Robin. La mesura de consolidació que en resulta és calculada per l'interpolador mitjana: $\frac{20+50}{2} = 35$.

En setè lloc i finalment, es mostre el contingut de l'arxiu de discs Round Robin, només n'hi ha un, de la base de dades. Es pot veure com el buffer està buit i com el disc conté una mesura consolidada en el temps 10 i amb valor 35, el calculat a l'operació roda anterior.

Listing 7.8: Exemple de funcionament de la implementació en Python de la RRD

```

1 >>> M = RRD(5,mitjana)
2 >>> M.afegix_disc(10,2,mitjana)
3 >>>
4 >>> m1 = Mesura(10,1)
5 >>> m2 = Mesura(10,2)
6 >>> m3 = Mesura(40,5)
7 >>> m4 = Mesura(50,10)
8 >>>
9 >>> M.update(m1)
10 >>> M.consolidable()
11 False
12 >>> M.update(m2)
13 >>> M.consolidable()
14 False
15 >>> M.update(m3)
16 >>> M.consolidable()
17 True
18 >>> M.update(m4)
19 >>> M.consolidable()
20 True
21 >>> M.consolidar()
22 >>> M.rodable()
23 False
24 >>> M.consolidable()
25 True
26 >>> M.consolidar()
27 >>> M.rodable()
28 True
29 >>> M.consolidable()
30 False
31 >>> M.roda()
32 >>> M.rodable()
33 False
34 >>> M.consolidable()
35 False
36 >>> M.A
37 [DRR:Buffer(SerieTemporal([],10,10),Disc(SerieTemporal([m(35.0,10)]), |2|)]

```

7.9 Conclusions de la implementació

Amb aquesta implementació amb Python del model RRD s'ha provat que el model dissenyat té un funcionament correcte. En la implementació hi ha detalls del llenguatge que dificulten la comprensió de l'estructura i funcionament dels SGBD RRD, i això que Python permet definir el programa amb sintaxi molt propera a la del model matemàtic.

Per treballar amb una RRD, l'usuari només ha de conèixer la classe RRD i la classe Mesura. No li cal usar SerieTemporal, Buffer, Disc ni tampoc DiscRoundRobin, ja que aquestes classes s'usen internament.

Aquesta, és una implementació bàsica i no hi ha tota la potencialitat del model RRD. Només s'ha provat la implementació amb l'interpolador mitjana, els temps es consideren relatius, no hi ha les operacions de representació de sèries temporals, etc. En aquesta primera implementació s'ha implementat un temps de manera senzilla, de la manera relativa. En una segona implementació caldrà millorar aquest aspecte ja que ara no és possible comparar dues bases de dades RRD.

Pel que fa a les mesures no s'ha definit si són multivaluades o només admeten un valor per mesura. Podrien ser perfectament un objecte collecció, per exemple una llista de mesures, i només caldria que els interpoladors en fossin conscients per tal de tenir l'estructura i el comportament per mesures multivaluades.

8 Conclusions

En aquest capítol es resumeix l'exposat en el cos principal del document, s'extreuen conclusions del model dissenyat i es proposen treballs futurs a partir d'aquest model. Amb aquest el document s'han complert els objectius plantejats:

- En el capítol 2 s'ha situat l'estat actual de l'emmagatzematge i del tractament de sèries temporals.
- En els capítols 3, 4, i 5 s'ha estudiat profundament el sistema de gestió de bases de dades RRDtool.
- En el capítol 6 s'ha dissenyat un model de dades que descriu l'estructura i el comportament dels SGBD per a sèries temporals. Aquest model s'ha anomenat model de dades Round Robin (RRD).
- En el capítol 7 s'ha proposat una implementació de referència del model dissenyat. Per aquesta implementació s'ha utilitzat el llenguatge Python.
- En aquest capítol 8 de conclusions, a l'apartat 8.1, es proposen millores i treballs futurs al voltant del model dissenyat.

En la primera part del document (caps. 3, 4 i 5) s'ha detallat el sistema de gestió de bases de dades (SGBD) RRDtool. És un SGBD que permet emmagatzemar i tractar sèries temporals distribuint-les en diferents intervals de temps. En tres capítols s'ha analitzat profundament el comportament de RRDtool, del qual cal destacar-ne un seguit de particularitats.

En primer lloc, les dades es desen normalitzades en el temps, és a dir que entre dues mesures emmagatzemades sempre hi ha el mateix temps, el qual es considera el temps de mostreig bàsic d'una base de dades RRDtool. Per a normalitzar, les dades s'interpolen seguint un criteri que conservar l'àrea de sota la corba. Com a conseqüència de la interpolació, no es desen els valors 'tal qual' s'han inserit sinó que es modifiquen segons el criteri descrit, per tant en la consulta cal tenir present de cercar-hi informació i no les dades mesurades.

En segon lloc, les dades es consoliden, és a dir que es desen amb diferents intervals de temps. La consolidació és també una interpolació però de valors ja normalitzats a uns altres valors normalitzats amb temps de mostreig diferent. En la consolidació RRDtool permet quatre criteris: mitjana, màxim, mínim i últim.

En tercer lloc, RRDtool té un tractament adequat de valors desconeguts. Primer, sap detectar quan cal considerar un valor desconegut, ja sigui perquè no s'ha me-

surat, perquè sobrepassa un llindar o perquè s'ha exhaurit el termini de mesurar. Segon, es pot qualificar quan un interval de temps s'ha d'emmagatzemar com a desconegut. Tercer, sap calcular amb valors desconeguts.

En quart lloc, RRDtool es va iniciar en l'àmbit de les sèries temporals per a comptadors i encara conserva aquest context. Les dades que s'insereixen s'han de classificar en els quatre tipus *gauge*, *counter*, *absolute* o *derive*, els quals estan pensats per a comptadors. Si bé és cert que es permet inserir magnituds com a *gauge*, aquesta diferenciació especial no ajuda a comprendre que el tractament i la representació posterior sigui la mateixa per tots els tipus.

En cinquè lloc, i com a curiositat, RRDtool directament sap representar les dades gràficament. És un avantatge gran a l'hora de considerar la rapidesa del sistema, ja que sap aprofitar adequadament l'emmagatzematge en diferents temps de mostreig. Tot i així, només sap representar els valors de manera constant, molt pensat per a velocitats mitjanes de comptadors, però no fa interpolacions lineals dels valors, la manera habitual de representar sèries temporals.

En sisè lloc, la mida de la base de dades és fixa des de la creació. És una conseqüència del model Round Robin afegit a que els interpoladors només treballen amb un valor. És un gran avantatge ja que la mida de les bases de dades per sèries temporals és un factor crític. Com a desavantatge, els interpoladors queden restringits als que pugin fer càlculs acumulatius en un valor.

En setè lloc, una base de dades RRDtool no té un comptador de temps com a rellotge sinó que les operacions indiquen en quina hora i data s'efectuen. A partir d'aquests valors, RRDtool executa el pas del temps que sigui necessari.

En vuitè lloc i com a resum dels altres, si la base de dades forma part d'un sistema de monitoratge complex, l'usuari no ha d'estar pendent del sistema d'emmagatzematge i tractament. La base de dades no pot esgotar l'espai de disc, no pot desincronitzar-se en el temps, els valors desconeguts estan sota control i els gràfics són representats de manera coherent.

El capítol 5 és on es detalla com és RRDtool a nivell d'usuari avançat. Aquest tipus de manual actualment no es pot trobar en la documentació de RRDtool, només hi ha el manual d'usuari bàsic [24], el qual parla en gran part de sintaxi i només anomena alguns aspectes singulars del model RRD, i un tutorial de van den Bogaerdt [3] que introdueix els conceptes de RRDtool en tres etapes però sense detallar els efectes que tenen cadascuna. D'aquestes dues documentacions s'han llegit els conceptes de RRDtool i els exemples d'ús que proposen, a partir dels quals han sorgit les idees per experimentar amb RRDtool i per reflexionar sobre les implicacions que té cada paràmetre; tot i que en alguns punts el manual i el tutorial s'arriben a contradir. Com a resultat s'han obtingut els capítols 3 i 5 de desenvolupament, els quals poden esdevenir una bona documentació avançada de RRDtool.

A partir del desenvolupament d'aquesta primera part del document, en la segona part s'ha dissenyat un model de dades per a sèries temporals que s'ha anomenat model Round Robin (RRD).

En el capítol 6, s'ha definit un SGBD RRD com un sistema informàtic d'emmagatzematge de sèries temporals, enteses com una una col·lecció de dades mesurades en diferents instants de temps. Segons el model RRD, a la base de dades una sèrie temporal queda estructurada de forma compacta i repartida segons diferents funcions d'interpolació i períodes de mostreig. Aquest repartiment té lloc en els diferents discs Round Robin, els quals fan ús del seu buffer per interpoliar les mesures i fan ús del seu disc per consolidar-les. El conjunt de discs Round Robin constitueixen la part principal del model RRD tot i que hi pot haver un buffer d'entrada de mesures comú per tal de regularitzar la sèrie temporal des d'un principi, ja que el pas de no regular a regular requereix interpoladors més complicats.

El model RRD resumeix, clarifica i posa en valor els conceptes principals desenvolupats a la part de RRDtool d'aquest document. Els SGBD, com RRDtool, solen tenir massa detalls d'implementació que dificulten la comprensió de l'estructura i el comportament de les bases de dades. A partir d'ara, en disposar del model RRD ja no serà necessari haver d'interpretar els conceptes Round Robin a partir de RRDtool; el model matemàtic en descriu exactament l'estructura i el comportament així com de tots els SGBD que l'implementin.

En el capítol 7, s'ha implementat un SGBD RRD utilitzant el llenguatge de programació Python. S'ha implementat a nivell acadèmic per visualitzar el funcionament correcte del model, ja que Python és un llenguatge d'alt nivell i s'allunya gaire de les definicions matemàtiques del model. En altres SGBD de nivell productiu, com és el cas de RRDtool, el model RRD no es reconeix tant immediatament ja que hi ha massa detalls d'implementació. A més, RRDtool presenta una particularitat del model ja que en la implementació dels buffers en comptes d'interpoliar sobre una sèrie temporal, per cada interval de consolidació es va acumulant la interpolació en un sol valor. L'objectiu és que la mida de la base de dades quedi perfectament definida però restringeix els interpoladors a funcions que es puguin calcular acumulativament. En la mateixa línia, es podria plantejar un model particular, esquematitzat a la figura 8.1, on els buffers fossin el disc d'un altre disc Round Robin amb el període de mostreig més petit. És a dir, els buffers no desen mesures sinó que utilitzen l'emmagatzematge dels altres discs. Com a contrapartida, els passos de consolidació dels discs es restringeixen a múltiples dels altres discs.

Un concepte al qual RRDtool dedica molta importància és al de comptadors, velocitats mitjanes i magnituds, però no s'ha trobat rellevant descriure-ho com a part del model ja és fruit de l'especialitat que RRDtool té amb els comptadors. No obstant, el model RRD si que permet l'anàlisi d'aquests sistemes, es poden configurar els paràmetres d'interpoladors i períodes de mostreig adequadament per tal que es corresponguin amb el comportament que té RRDtool.

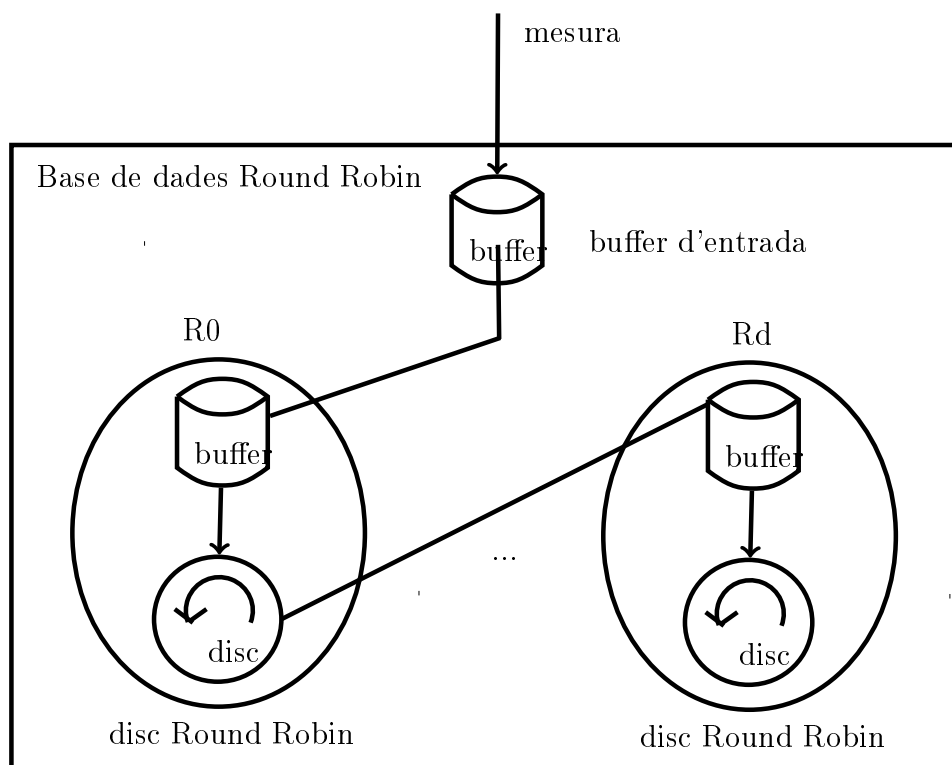


Figura 8.1: Esquema particular del model Round Robin sense emmagatzemar a buffer

Altres conceptes que es creu que sí que podrien formar part del model es proposen tot seguit.

8.1 Treball futur

El model de dades Round Robin dissenyat apropa l'àmbit de sistemes de monitoratge al de mineria de dades per sèries temporals. A partir d'aquest model, hi ha espai per investigar i desenvolupar noves tècniques que tractin les mesures que s'han adquirit en el monitoratge com a sèries temporals.

El tractament de mesures desconegudes és un bon candidat a formar part del model ja que en el monitoratge de sèries temporals no es poden evitar. Primer, s'hauria d'ampliar el conjunt de valors d'una mesura amb un valor especial que fos el 'desconegut'. Segon, de manera conjunta als buffers i als interpoladors, s'hauria de definir uns predicats que marquin quan cal considerar que el resultat d'una interpolació no pot prendre el valor calculat sinó que ha de prendre el valor 'desconegut'. Tercer, els interpoladors haurien de saber treballar interpolant quan en els intervals de consolidació hi ha mesures desconegudes. En el model RRD dissenyat es permet interpolat encara que en un interval no s'hagin pres mesures, però no es

defineix com es comporta quan en un interval hi ha barrejades mesures conegudes i desconegudes.

L'operació de consulta també és un bon candidat pel model. En el model RRD s'ha definit la representació de sèries temporals exemplificada amb funcions a trossos constants. Però la representació només és la primera part de l'operació de consulta, a les bases de dades ja solen ser complicades i per sèries temporals encara més. Les consultes han de definir el comportament davant de les tasques que es recerquen en l'àmbit de les sèries temporals, com per exemple l'indexat, l'agrupament, la classificació o la segmentació, [12]. Cal estudiar detingudament les tècniques que donen bons resultats i definir-les des del punt de vista del model RRD; és a dir com consultar a través dels diferents discs Round Robin, i determinar si donen bons resultats amb el model RRD. Aquestes operacions per sèries temporals se solen basar en cercar similituds entre dues sèries temporals. Així doncs, les operacions de consulta també han de ser capaces de fusionar i calcular amb dades de dues o més bases de dades RRD.

En l'operació de consulta s'hi inclouen els procediments de predicció. Hi ha una gran col·lecció de mètodes de predicció en el camp de les sèries temporals: *moving average*, *exponential smoothing*, *Autoregressive integrated moving average* (ARIMA), etc. i s'ha de plantejar si és possible aplicar-los a una sèrie temporal que està partida amb diferents temps de mostreig, com és el cas dels diferents discs Round Robin. Els mètodes de predicció també permeten validar i reconstruir valors a la base de dades.

En el model RRD hi ha uns quants paràmetres que cal configurar per cada base de dades, com són els interpoladors, els períodes de mostreig, la quantitat de discs Round Robin, etc. En aquest sentit es pot estudiar quin efecte tenen diferents configuracions de paràmetres i plantejar la hipòtesis que donades dues bases de dades RRD per a la mateixa sèrie temporal, en una hi ha més informació. El SGBD RRDtool ha implementat el model RRD amb una configuració determinada de paràmetres de tal manera que conserva les dades segons el criteri del total; aquest s'entén molt bé per comptadors però no per altres tipus de dades. En el model RRD, es pot estudiar amb més detall l'efecte d'aquest criteri i d'altres, com per exemple criteris que minimitzin l'error de representació.

A part del model RRD, hi ha altres tècniques que proposen solucions d'emmagatzematge per a les sèries temporals, com per exemple T-Time, [2], o iSAX, [13]. Es poden comparar amb el model RRD pel que fa a avantatges i inconvenients i ampliar el model RRD amb conceptes interessants que aquestes tècniques puguin tenir.

Finalment, cal comprovar quina resposta tenen els SGBD RRD amb dades utilitzades en altres experiments, com els proposats per Keogh i Kasetty, [12]. Tal com recomanen, en els experiments de mineria de dades de sèries temporals s'han d'utilitzar els mateixos valors ja que sinó es pot produir una desviació interessada en les conclusions.

Bibliografia

- [1] Johannes Aßfalg. “Advanced Analysis on Temporal Data”. PhD. Thesis. Fakultät für Mathematik, Informatik und Statistik der Ludwig Maximilians Universität München, 19 de maig de 2008. URL: <http://edoc.ub.uni-muenchen.de/8798/> (vis. 14-04-2011).
- [2] Johannes Aßfalg i d’altres. “T-Time: Threshold-Based Data Mining on Time Series”. A: *Proceedings of the 24th International Conference on Data Engineering*. ICDE’08. Cancun, Mexico: IEEE, abr. de 2008, pàgs. 1620–1623.
- [3] Alex van den Bogaerdt. *RRDtool, Tutorials and explanations*. URL: <http://www.vandenbogaerdt.nl/rrdtool/> (vis. 14-12-2010).
- [4] Jake D. Brutlag. “Aberrant Behavior Detection in Time Series for Network Monitoring”. A: *Proceedings of the 14th Systems Administration Conference*. LISA’00. New Orleans, Los Angeles: USENIX Association, des. de 2000, pàgs. 139–146. URL: <http://www.usenix.org/events/lisa00/brutlag.html>.
- [5] Alessandro Camerra i d’altres. “iSAX 2.0: Indexing and Mining One Billion Time Series”. A: *Proceedings of the 10th IEEE International Conference on Data Mining*. ICDM’10. Sydney, Australia: IEEE, des. de 2010, pàgs. 58–67. URL: http://www.cs.ucr.edu/~eamonn/iSAX_2.0.pdf (vis. 15-03-2011).
- [6] Dale Carder. *RRDtool Scalability, performance in the large scale*. URL: <http://net.doit.wisc.edu/~dwcarder/rrdcache/> (vis. 22-03-2011).
- [7] S. Chilingaryan i d’altres. “Advanced data extraction infrastructure: Web based system for management of time series data”. A: *Journal of Physics: Conference Series. 17th International Conference on Computing in High Energy and Nuclear Physics (CHEP’09)* 219.4 (2010). DOI: 10.1088/1742-6596/219/4/042034.
- [8] Christopher J. Date. *An Introduction to Database Systems*. 7a ed. Boston, MA, USA: Addison-Wesley, 2000. ISBN: 0-201-38590-2.
- [9] Tak chung Fu. “A review on time series data mining”. A: *Engineering Applications of Artificial Intelligence* 24.1 (feb. de 2011), pàgs. 164–181. DOI: 10.1016/j.engappai.2010.09.007.
- [10] HP. *RPN a les calculadores HP*. URL: <http://www.hpmuseum.org/rpn.htm> (vis. 02-12-2010).

- [11] Icinga. *Icinga - Open Source Monitoring, Nagios fork*. URL: <http://www.icinga.org/> (vis. 25-10-2010).
- [12] Eamonn Keogh i Shruti Kasetty. “On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration”. A: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD’02. Edmonton, Alberta, Canada: ACM, jul. de 2002, pàgs. 102–111. URL: http://www.cs.ucr.edu/~eamonn/sigkdd_bench.pdf (vis. 15-03-2011).
- [13] Eamonn Keogh i Jessica Lin. *iSAX (Symbolic Aggregate approXimation)*. URL: <http://www.cs.ucr.edu/~eamonn/iSAX/iSAX.htm> (vis. 15-03-2011).
- [14] Eamonn Keogh i Michael Pazzani. “An enhanced representation of time series which allows fast and accurate classification clustering and relevance feedback”. A: *Proceedings of the 4th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD’98. New York: ACM, ago. de 1998, pàgs. 239–243. URL: <http://www.cs.ucr.edu/~eamonn/kdd98.pdf> (vis. 15-03-2011).
- [15] Eamonn Keogh i Padhraic Smyth. “A probabilistic approach to fast pattern matching in time series databases”. A: *Proceedings of the 3rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD’97. Newport Beach, California: ACM, ago. de 1997, pàgs. 24–20. URL: <http://www.cs.ucr.edu/~eamonn/kdd97.ps> (vis. 15-03-2011).
- [16] Eamonn Keogh i d’altres. “Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases”. A: *Knowledge and Information Systems*. KAIS 3.3 (ago. de 2001), pàgs. 263–286. URL: http://www.cs.ucr.edu/~eamonn/kais_2000.pdf (vis. 15-03-2011).
- [17] Eamonn Keogh i d’altres. “Locally adaptive dimensionality reduction for indexing large time series databases”. A: *Proceedings of ACM SIGMOD International Conference on Management of Data*. SIGMOD’01. Santa Barbara, California, USA: ACM, maig de 2001, pàgs. 151–162. URL: http://www.cs.ucr.edu/~eamonn/sigmod_apca_2001.pdf (vis. 15-03-2011).
- [18] G. Klyne i C. Newman. *Date and Time on the Internet: Timestamps*. Request for Comments 3339. Internet Engineering Task Force. IETF, jul. de 2002. URL: <http://www.ietf.org/rfc/rfc3339.txt>.
- [19] Mark Last, Abraham Kandel i Horst Bunke, eds. *Data mining in time series databases*. Series in Machine Perception and Artificial Intelligence 57. Singapore: World Scientific, 2004.
- [20] Sasa Markovic i Arne Vandamme. *JRobin: A Java implementation of RRD-tool*. URL: <http://www.jrobin.org/> (vis. 22-03-2011).
- [21] Nagios. *Nagios - The Industry Standard In IT Infrastructure Monitoring*. URL: <http://www.nagios.org/> (vis. 25-10-2010).

- [22] Tobias Oetiker. “MRTG The Multi Router Traffic Grapher”. A: *Proceedings of the 12th Systems Administration Conference*. LISA’98. Boston, Massachusetts: USENIX Association, des. de 1998, pàgs. 141–148. URL: <http://www.usenix.org/publications/library/proceedings/lisa98/oetiker.html>.
- [23] Tobias Oetiker. *RRDtool graph, documentació de gràfics*. URL: http://oss.oetiker.ch/rrdtool/doc/rrdgraph_graph.en.html (vis. 15-10-2010).
- [24] Tobias Oetiker. *RRDtool, Round Robin database*. 1998–2011. URL: <http://oss.oetiker.ch/rrdtool/> (vis. 10-10-2011).
- [25] Tobias Oetiker. *The Multi Router Traffic Grapher*. URL: <http://oss.oetiker.ch/mrtg/> (vis. 15-10-2010).
- [26] David Plonka, Archit Gupta i Dale Carder. “Application Buffer-Cache Management for Performance: Running the World’s Largest MRTG”. A: *Proceedings of the 21st Systems Administration Conference*. LISA’07. Dallas, Texas: USENIX Association, nov. de 2007, pàgs. 63–78. URL: <http://www.usenix.org/events/lisa07/tech/plonka.html>.
- [27] Python Software Foundation. *Python documentation*. Ver. 2.7.1. URL: <http://docs.python.org/> (vis. 06-04-2011).
- [28] Joseba Quevedo i d’altres. “Validation and reconstruction of flow meter data in the Barcelona water distribution network”. A: *Control Engineering Practice* 18.6 (juny de 2010), pàgs. 640–651.
- [29] Jin Shieh i Eamonn Keogh. “iSAX: Indexing and Mining Terabyte Sized Time Series”. A: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD’08. Las Vegas: ACM, ago. de 2008, pàgs. 623–631. URL: <http://www.cs.ucr.edu/~eamonn/iSAX.pdf> (vis. 15-03-2011).
- [30] Wikipedia. *IEEE Standard for Floating-Point Arithmetic (IEEE 754-2008)* — *Wikipedia, The Free Encyclopedia*. Anglès. 2011. URL: http://en.wikipedia.org/wiki/IEEE_754-2008 (vis. 20-04-2011).
- [31] Wikipedia. *Temporal database* — *Wikipedia, The Free Encyclopedia*. English. 2011. URL: http://en.wikipedia.org/wiki/Temporal_database (vis. 12-10-2010).
- [32] Wikipedia. *TSDS, Time series database server* — *Wikipedia, The Free Encyclopedia*. Anglès. 2011. URL: http://en.wikipedia.org/wiki/Time_series_database (vis. 12-10-2010).
- [33] Wikipedia. *Unix time* - *Wikipedia, the free encyclopedia*. URL: http://en.wikipedia.org/wiki/Unix_time (vis. 18-01-2011).
- [34] Qiang Yang i Xindong Wu. “10 Challenging Problems in Data Mining Research”. A: *International Journal of Information Technology and Decision Making*. IJITDM 5.4 (des. de 2006), pàgs. 597–604. URL: <http://www.cs.uvm.edu/~icdm/10Problems/index.shtml> (vis. 22-05-2011).

Bibliografia

- [35] Jian Zhang i Renato Figueiredo. “Adaptive Predictor Integration for System Performance Prediction”. A: *Proceedings of the 21st International Parallel and Distributed Processing Symposium*. IPDPS'07. Long Beach, California: IEEE, març de 2007. URL: <http://www.cecs.uci.edu/~papers/ipdps07/pdfs/IPDPS-1569011105-paper-1.pdf> (vis. 03-10-2010).