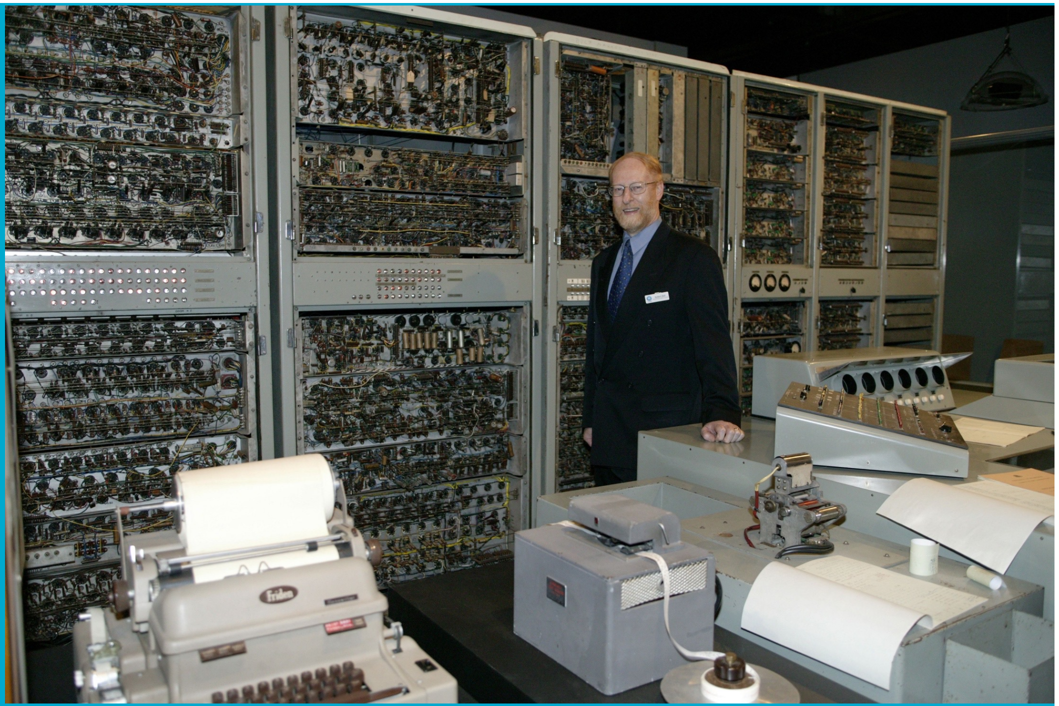


Resource Allocation – Policies and Management

Robert C. Bell | CSIRO IMT Scientific Computing

2 August 2017

Context

- Acquiring an HPC system, server, facility, storage
- Funded by someone - stakeholders
- Need to apportion resources to reflect the funding sources
- Or, need to sell access to resources to cover costs
- And, need to stop one user/group/application/organisation dominating
- Need to aim for high utilisation to justify acquisition
- ”Resources” are valuables, like computer time
  - You can’t save up computer time – not a physical asset
- Sharing: need to deal with contention between users
  - Continuing from lessons supposedly learnt in kindergarten!

Introduction

- Definitions of resources
- Policies for allocation of resources
  - For users, administrators and managers
  - Why?
    - To reflect agreements, partnerships
    - To deal with contention for resources
    - To maximise productivity (reliability, performance, capacity)
    - To minimise loss, downtime, frustration, wasted effort, expense
- Shared systems
- My view, as a service provider and facilitator
- Main focus: compute resources, then storage

History

- CSIRAC: 1949-1956
- CSIRO Division of Computing Research and Csironet: 1963-1990
  - CSIRO developed a packet switched network in 1969 – same year as DARPANET
  - CSIRO had the equal of the world's fastest computer from ~1973
    - CDC Cyber 76
  - Served CSIRO and government departments

History: Csironet allocation

- Csironet:
  - Government policy – cost recovery from users
  - Charging for almost everything – CPU time, printing, storage
    - To reflect actual costs, amortised costs of capital, encourage good user behaviour
  - Charged after usage
  - Costs debited from operating funds of projects
  - Hard to budget
  - Couldn't afford to make mistakes
  - Loop:
    - Flight to other options, e.g. mini-computers
    - Expensive systems under-utilised
    - Charging rates increased to cover fixed costs . . .
  - End loop

Services Note No 1

Csironet ACCOUNTING
COMPUTING CHARGES AND CREDITING

Edition 13.0

January 1989

CSIRONET

CONTENTS

3 VSOS SYSTEM

3.1 VSOS Batch Job Processing (Table 31)

Batch jobs may be initiated from the front end computers or from other VSOS jobs. Various job categories are provided, for different grades of service at different charging rates. The job category is set to JDEFAULT if no user specification is given and is otherwise specified as the value of the JCAT parameter on the RESOURCE statement. Job category resource limits and operational restrictions all described in Csironet Services Note No 3, 'VSOS System Services' (SN3), and Csironet Users Manual Volume 3, 'VSOS Users Manual' (UM3).

Resources used by a job are expressed in System Billing Units (SBUs). SBUs are multiplied by the user class factor (See Table 02) and by a base rate multiplier to give a dollar charge. The complicated SBU formula is intended to give as nearly as possible a constant result for the same work - i.e. independent of system load.

Table 31. VSOS Batch Job Processing Charge

Item	Description
3100	VSOS batch job processing charge = bu(X+f)S + L $
3101	b = base rate = 0.0637
	u = user class factor (see Table 02)
3102	S = resource usage in System Billing Units
	= C * (3.42*(UCPU + SCPU)
	+ 0.00082*MEMU*(1+SCPU/UCPU)
	+ 0.024*(LPACCX + SPACCX)
	+ 0.0024*(SPSECX)
	+ 0.6*(LPGFLT - LLPC)
	+ 0.021*(SPGFLT - LSPC)
	+ 0.001*(VSCALLS))
where	C = Job category multiplier.
3105	X = Special charge factor
	= 0.25 for CSIRO users
	= 1.00 for other users
3130	category unknown
3131	JDEFAULT , C = 3.333

Table 31. (Cont...) VSOS Batch Job Processing Charge

Item	Description
3140	n6= software surcharge units L = software surcharge = n*r r is rate per software unit r = 2.1429 for VIP r = 0.5785 for MAGEV
3150	UCPU = user execution time (seconds).
3151	SCPU = system execution time (seconds).
3152	MEMU = memory usage (working set in blocks X CPU time in seconds). A block is 4096 bytes.
3153	LPACCX = disk accesses for large page explicit reads and writes.
3154	SPACCX = disk accesses for small page explicit reads and writes.
3155	SPSECX = number of disk sectors transferred for explicit reads and writes. A disk sector is 4096 bytes.
3156	LPGFLT = large page "faults".
3157	LLPC = large pages "lost".
3158	SPGFLT = small page "faults".
3159	LSPC = small pages "lost".
3160	VSCALLS = number of calls to the virtual system.
3180	f is software royalty charge factor as follows f = 0.25 for use of ADINA f = 0.25 for use of BITEMJ f = 2.00 for use of ECLIPSE f = 0.40 for use of TWODEPEP f = 0.55 for use of ADINA 84, ADINAT 84, ADINPL, ADINA-IN f = 1.5 for use of VIP f = 1.5 for use of IMSL f = 0.6 for use of BEFE f = 0.01 for use of GAMESS, KONDO
3195	Manually Entered Charge.
3198	Processing Discount (See 0.4)

History: Csironet allocation

- Different system for Cyber 205: 1984-
  - Grants of ’money’ to use on the system: both CSIRO and external
  - Real cost-recovery on front-end and other systems!
- Csironet:
  - Broken, principally because of charging policy
  - Privatised
    - I found that administration had negotiated a large discount for administrative processing, smaller discount for science processing!
  - Disappeared
- Example of a wrong allocation/funding policy leading to the death of a service: end-user charging
- Scientists are good at avoiding paying for things!

History: CSIRO computing restart, 1990-

- Budget for systems and services!
- Joint facility with private company
  - ran the system, and had a share of a Cray Y-MP
- Fixed shares agreed between parties
- Fair-share scheduler to apportion CPU-time between and within parties
  - But didn't make decisions about which jobs to run
  - Found (tip-off) that out of business hours, provider was favouring its jobs!
- Found that industry was not beating a path to its door for supercomputer time, and folded. (Others failed to attract customers, e.g. ANSTO, ACCI)

History: CSIRO computing restart, 1990-

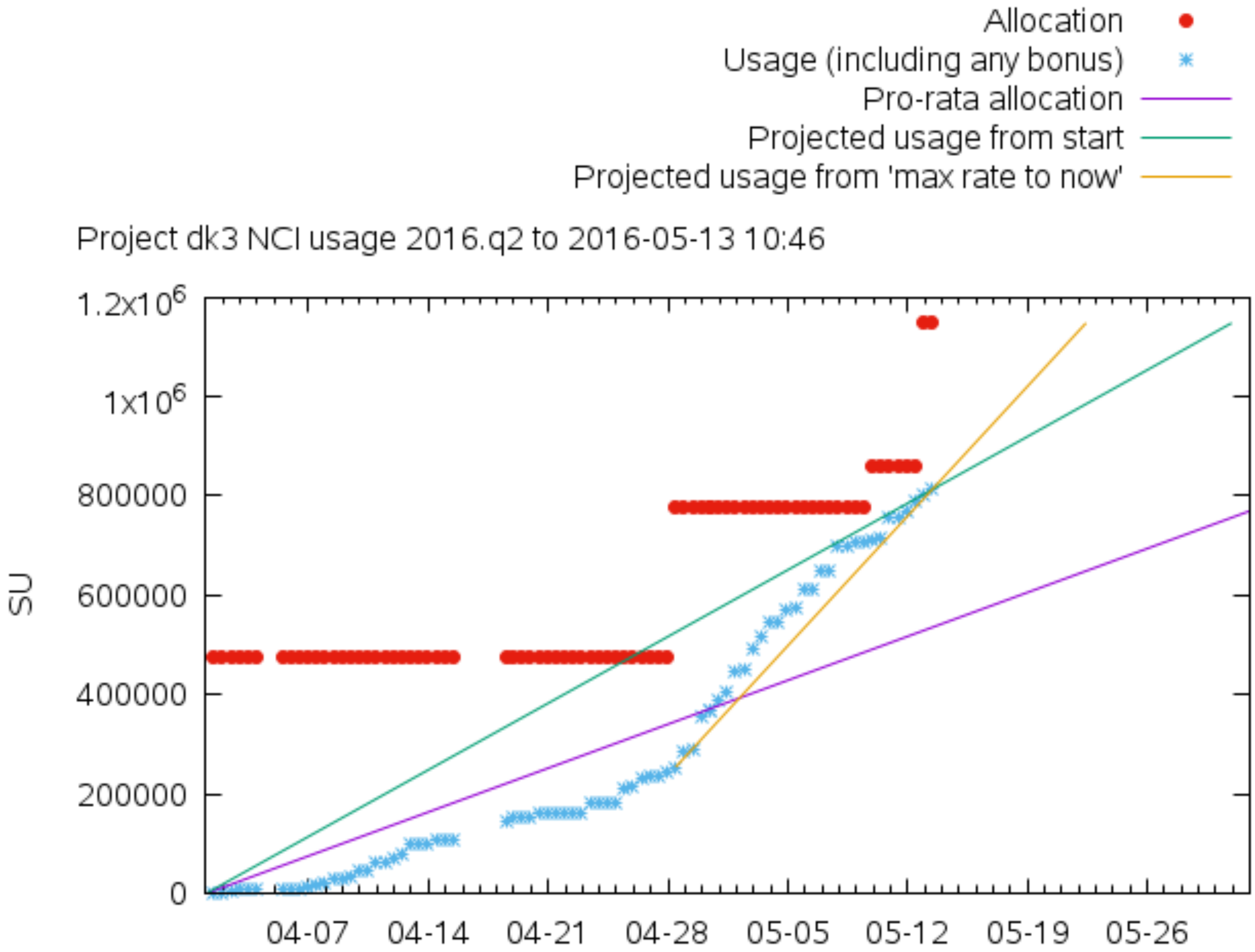
- Lessons
  - Resisted charging for compute time: “Divisions will have to pay for it”
  - Set up share scheme, where shares on system were proportional to contributions to a Development Fund for enhancements
  - Scheduler enhanced to make decisions about which jobs to run based on FSS
  - Recognition that since most costs are fixed, charging by usage is the wrong model!

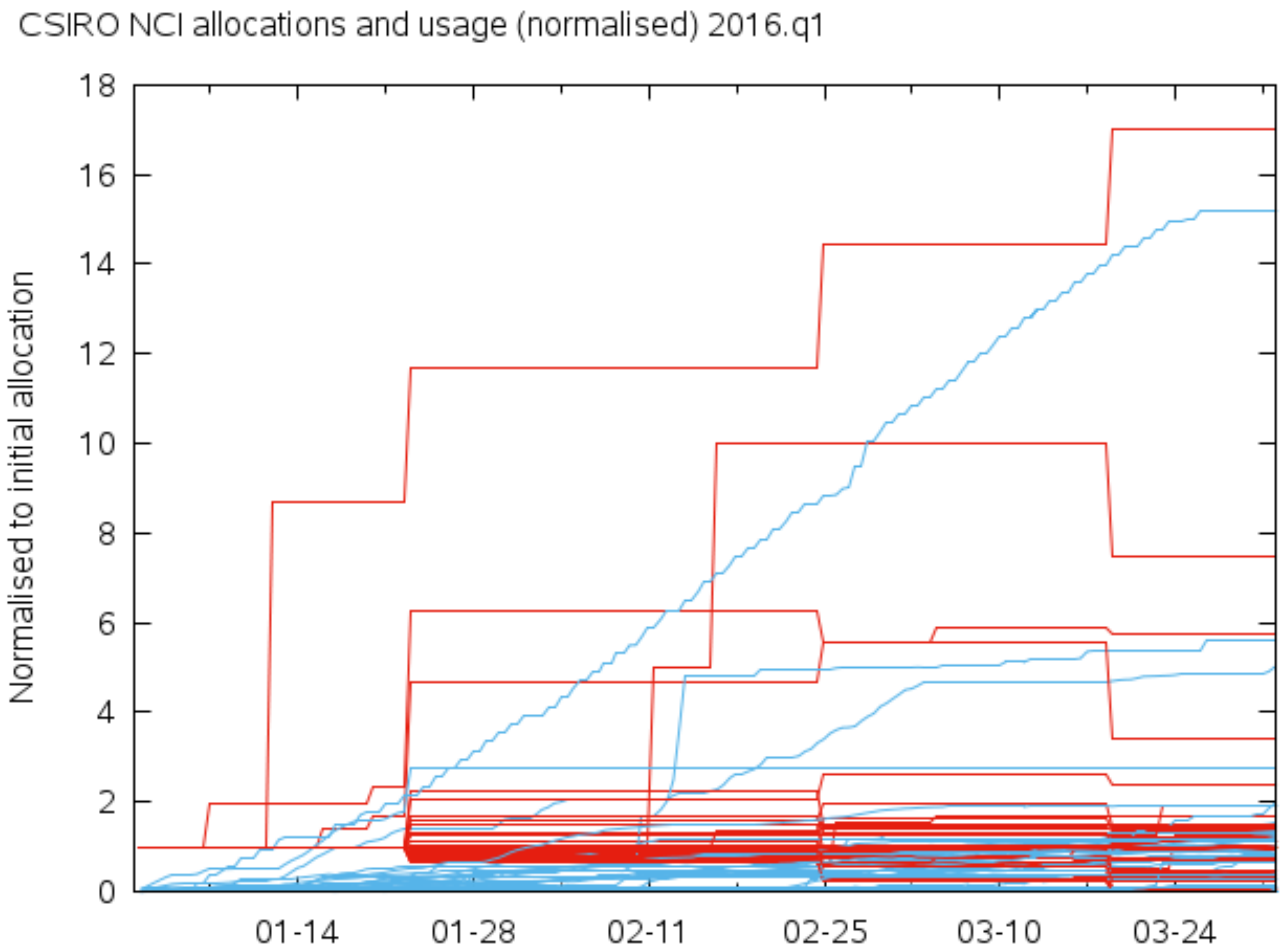
History: Bureau and CSIRO HPCCC, 1997-

- Joint centre, 50:50
- Had NEC build Enhanced Resource Scheduler to support URGENT and NORMAL jobs. FSS for organisations and groups and users.
- Had checkpoint/restart for pre-emption (as did Crays before them).
- Are all CPU hours equal? No: on-demand is more valuable than crumbs from the table.
  - Resources need to be scaled by priority given
    - Never implemented
  - Later systems split instead

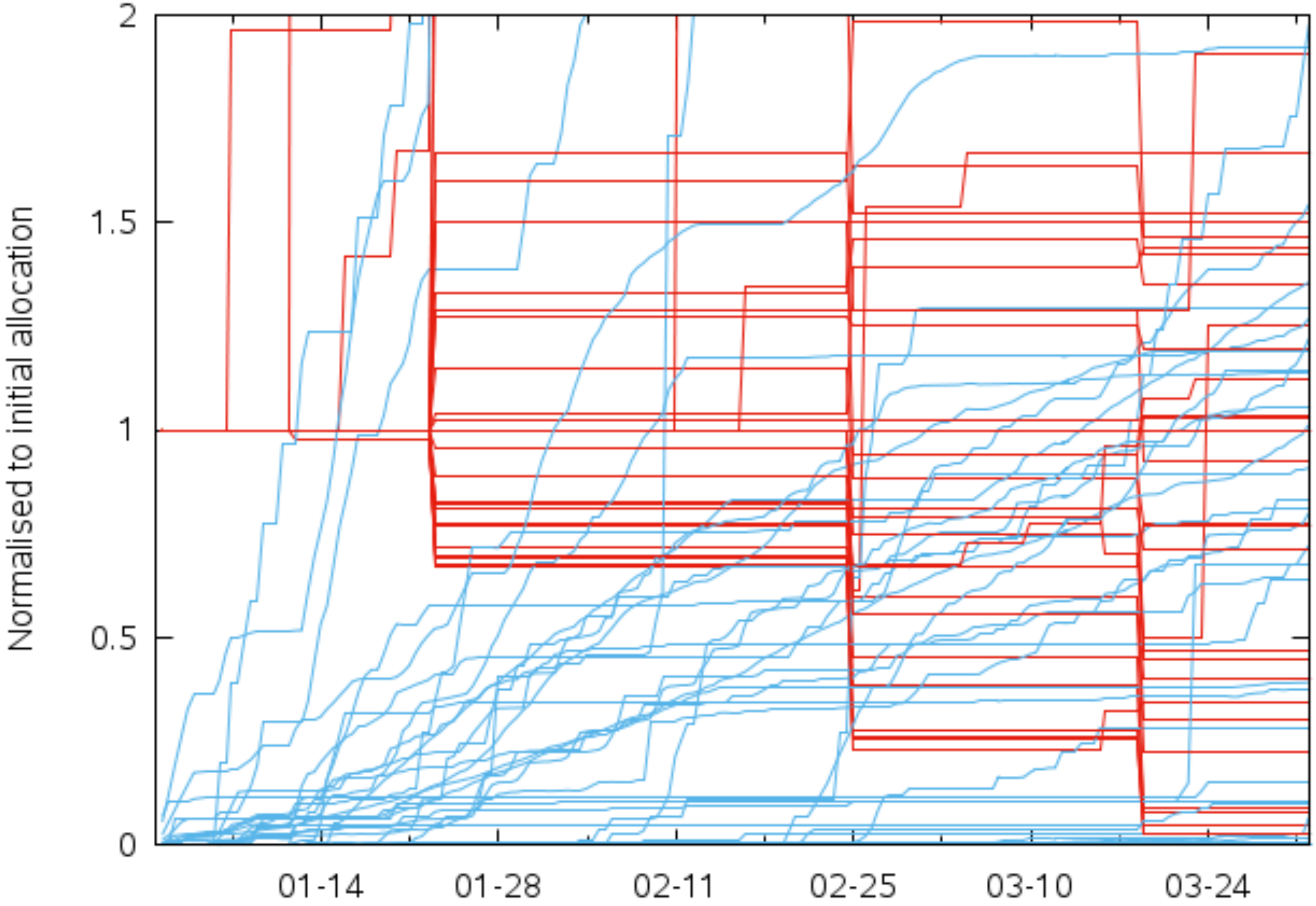
History: Partner facilities

- APAC and NCI
- MAS and partner shares
- Originally, same process for both: too ‘heavy’ for partner share projects
- Allocation of CPU time (service units)
- Allocation of storage
- Quarterly in advance
- Research is not like that – highly varying!
- => Demand management by CSIRO


Allocation
Usage (including any bonus)
Pro-rata allocation
Projected usage from start
Projected usage from 'max rate to now'
Project dk3 NCI usage 2016.q2 to 2016-05-13 10:46
1.2x10^6
1x10^6
800000
600000
400000
200000
0
SU
04-07
04-14
04-21
04-28
05-05
05-12
05-19
05-26


CSIRO NCI allocations and usage (normalised) 2016.q1
Normalised to initial allocation
0
2
4
6
8
10
12
14
16
18
01-14
01-28
02-11
02-25
03-10
03-24

CSIRO NCI allocations and usage (normalised) 2016.q1



Allocations – options?

- CSIRO internal – Fair Share, and squeaky wheel principle
- NCI Bonus time concept helps, but …
  - Computer time is like a banana
    – a right time to eat, can’t save it up.
- Some time is more valuable than others – Friday afternoon rush compared with Monday morning – market?
- Bad effects at beginning and end of quarters
- Rolling allocations?
- Realisation that instead of allocations entitling holders to hours of time, it should entitle them to a guaranteed access.
- Changes at VLSCI/Melbourne Bioinformatics – Michael Kuiper
- See
    - https://docs.google.com/document/d/1U7bZQ-AeiYojIS20AiCFdN0Ud0OkfLPiqmX_9NabUNI/edit?pref=2&pli=1

Allocations – VLSCI and Fair Share (2016)

- **1) Poor usage estimations / low utilization of Avoca.**
- Despite being oversubscribed, Avoca usage for the last year has been less than **58%**. The primary reason for this is that it is difficult for users to estimate their usage over the course of a year.
- **2) End of quarter rush** - **uneven usage patterns.**
- Usage typically spikes towards the end of a quarter as users try to use up their quotas. This can lead to frustrations as jobs wait in queue for days at a time.

Allocations – VLSCI and Fair Share

- **3) Official RAS allocation is once a year.**
- There is one official RAS allocation round per year. Despite the availability of 'start-up' grants, this has been discouraging for new users.
- **4) Data backlog.**
- A problem with user data is looming. Should the VLSCI need to close projects and move off data this would take most of a year to do so. Users would have to migrate hundreds of terabytes of data to backup space, assuming they have them.

Allocations – VLSCI and Fair Share

- **A) Implementation of fairshare system.**
- **What is fairshare?**
  Fair-share is a job scheduling algorithm designed to distribute computational resources proportional to the group allocation. Fair-share is widely used in many high performance computing centers and can be implemented by many job schedulers such as Slurm. (which we have at VLSCI). Currently VLSCI employs a quota system which carries a significant administrative overhead.
- The most important concept to understand is that members have bought an **instantaneous** share of the resources which allows them guaranteed access to their share of the machine at any time during the year.

Allocations – VLSCI and Fair Share

- With the instantaneous share **we are not obliged** to fulfill cpu hours at a later date. Member subscription could be thought similar to gym membership. The more you use it, the more value you get from it.
- Currently, with a quota allocation, the situation arises where low usage at the beginning of the quarter results in oversubscription by the end of the quarter, with users requesting more hours than we can physically provide. Removing the quota system removes the driving force for these artifacts: **usage should reflect actual need and demand.**

Allocations – compute: conclusions

- Reflect agreements
- Should be based on some agreed metric
- Period-based allocations have short-comings
- Fair-share model better suited to a continuous supply
- Need to be implemented in scheduler
- Need pre-emption mechanism for high-priority work
- Need priority scheme for users

Resource allocation – storage

Tennyson, The Brook:
For men may come and men may go,
But I go on for ever.

Rob:
Flops may come and flops may go,
But bytes go on for ever.

John Mashey:
Disks are binary devices: new or full

Resource allocation – storage

- My 2005 statement
  - “Users typically want every file kept and backed-up, and would be happy to use only one file system, globally visible across all the systems they use, with high-performance everywhere, and infinite capacity!”
  - User added: zero cost!
- Same principles as compute:
  - Reflect agreements, define measure of cost, maximise use of resources, minimise disruption to others, maximise performance.
- Storage is a different entity from compute time
- Your Data, Our Responsibility (2009)
  - The facilities and policies for users’ data affect the productivity of the users and their perception of the service.

Resource allocation – storage – methods

- Government funded, merit allocation:
  - e.g. RDSI
- Institution service
- For HPC, ancilliary of compute allocation
- Free-for-all!
- Basic quotas
- Leased
- Purchased
- Cloud

Personal devices, desktops

- One file system per device
  - if visible at all
- Protected?
  - only if the user takes action with iTunes, Time Machine, etc
- Performance decays over time
- Cloud for global
- Infinite if you buy another device, or use cloud!

Storage on CSIRO SC shared systems

- $HOME – standard POSIX
- $TMPDIR – standard POSIX
- $FLUSHnDIR
  - formerly $PTMPDIR and $WORKDIR
- $DATADIR
- $LOCALDIR
- $STOREDIR
- $MEMDIR
- OSMs
- Data Access Portal

• Why so many?

Storage on CSIRO SC shared systems

- $HOME – small, backed-up
- $TMPDIR – job or session temporary
- $FLUSHDIR – longer
  - added $FLUSHnDIR – different perf.
- $DATADIR – ‘project’ area, but no backup
- $LOCALDIR – local on node – performance
- $STOREDIR – DMF
- $MEMDIR – in memory
- $OSMs – area per project
- Why? – getting around limitations

Policy for (shared) storage areas

Need:

- To control access and what can be stored there
  - (music or video library? – corporate policies)
- ‘quiet enjoyment’ for all users
- Performance
- Protection/recovery
- Cover
  - Protection for staff/management from users’ misunderstandings
- Controls to stop the FS from filling
  - for every user-accessible FS

Policy for (shared) storage areas - space management

- Quotas (space and inodes)
- HSM (automatic)
- Expiry (remove old files)
- Flushing (remove old/unused files)
- "Name and shame"
- Project/data areas
  - Hard to manage
  - Accumulate
  - No mechanism to recover space

CSIRO SC policies

- Had very little, until:
  - 1) Hobart user lost many files (including scripts) when we flushed $WORKDIR
  - 2) Talked with other sites: found one that made users sign a statement about storage policy, to absolve providers of blame!
- Implemented user guide and statement in registration

CSIRO SC registration

…

For further information on CSIRO SC systems please see the SC User Manual at: ….

In particular, please read the 'File System Conventions' section at: ….

It is imperative that you understand the file systems management policies on SC systems, including:

- *automated flushing/removal of files*
- *backups are limited to a few file systems*
- *file migration to tape on the CSIRO Data Store*
- *no guarantee of any file recovery in the event of major disasters.*

…

CSIRO SC Clusters - filesystems

In the table below: 'Properties' denotes the management attributes of the underlying filesystem: back-up (b), quota (q), global (g), local (l), job-temporary (j), flush (f), and/or migrated (m).

Variable name	Properties	purpose
$HOME	q, b	Login settings, scripts, source code and built software A limited amount of space will be available.
$DATADIR	q, g	Persistent files for use in multiple jobs. Ensure that critical files left here are backed up elsewhere.
$FLUSH1DIR, $FLUSH2DIR	q, f, g	Working files semi-persistent between sessions. Ensure that …
$STOREDIR	q, m, b	Long term storage - (nfs mount of) datastore on Ruby.
…		

NCI: file system policies – excerpt

Name[(1)]	Purpose	Availability	Quota[(2)]	Timelimit[(3)]	Backup
/home/unigrp/user	Irreproducible data eg. source code	raijin only	2GB (user)	none	Yes
/short/projectid	Large data IO, data maintained beyond one job	raijin only	72GB (project)	365 days	No

3. Timelimit defines time after which a file is erased on the file system since its most recent access time, as defined by the file access timestamp.

- Revised to have no timelimit
- Exact allocations are used, so FS can't fill.

Pawsey

- Pawsey Supercomputing Centre Data Storage and Management Policy – 16 pp
  - http://www.pawsey.org.au/wp-content/uploads/2015/01/PawseyDataManagementPolicy20151.pdf
- 1_Data_ownership_legal_ethical_guide.pdf
- 2_Data_documentation_guide.pdf
- 4_Data_Publication_Re-use_Guide.pdf
- 3_Data_storage_sharing_guide.pdf
  - 15 pages total

File system policies - Pawsey

System	Location	Initial Quota	Back-up	Purged	Time limit	Permissions
$HOME	/home/[username]	10 GB	From Q2 2015	No	-	700
$MYGROUP	/group/[project]/[username]	1 TB	No	No	-	750
$MYSCRATCH	/scratch/[project]/[username]	None enforced	No	Yes	30 days	750

- 30 day time limit on /scratch
- https://portal.pawsey.org.au/docs/Supercomputers/Magnus_Purge_Policy

File system purge policy – Pawsey magnus

Motivation

- *On previous supercomputers, the scratch file system was statically allocated to projects for their duration. This led to unintended results:*

1. *Users unable to take advantage of the full scratch system....*
2. *The use of the scratch system for long-term storage....*

- Fixed age – using Robinhood

2. Implementation …

of policy for (shared) storage areas

Implementation of policy for (shared) storage areas – space management

- Quotas (space and inodes)
- HSM (automatic)
- Expiry (remove old files)
- Flushing (remove unused files)
- “Name and shame”

Space management – quotas

- To control space (most often) & files
- To report on usage quickly
- For all areas that users could fill (/var)
  - Better for a user to hit a quota limit than for a FS to fill –> re-boot…
- Allocation by quotas. Either:
  - Balkanisation
    - sum of all quotas = available space
    - very wasteful
  - Over-commitment – harder to manage

Space management – HSM – big winner

- CSIRO SC Data Store – 25 years
  - No limits on data volumes (but limits on file numbers)
- HSM – ‘infinite’ storage, 'elastic' disk
  - 10 Tbyte /home on-line quota: biggest user has 1.2 Pbyte
- CSIRO SC provides direct user access with /home on ruby, SGI UV 3000
  - True meaning of HSM: otherwise, just an archive store, with users having to move data in and out
- CSIRO enhancements
- No form-filling; immediate access, low administration

Space management – expiry

- One site used to remove all files from /home more than a year old
- Other site wiped all the user areas apart from /home at the beginning of each new allocation cycle
  - Did not suit continuing projects

Space management - flushing

- Often not implemented
- Gets rid of old files – clutter that prevents active users getting sufficient ‘campaign’ storage
- CSIRO SC – scripts and program matching policy
  - triggered when usage reaches a threshold (typically 95%)
  - file audit – sorts, and deletes oldest until second threshold reached (typically 90%), or 7 days (rare)
  - uses mtime and atime – problem for FSes that don’t do atime
  - also removes empty directories
  - New scalable flushing algorithm
    - 2 second response time!

Space management – flushing – why is it hard?

- Need to be careful!
- Access and modify time
  - access time is dodgy with some filesystems (e.g. NFS)
- Sheer amount of work:
  - 283 M, 460 M files seen
- Global FSes – slow metadata operations – typically 1000/s
- Questionable policies – e.g. fixed times
  - wastes space and processing
- Bad practice to have policy and not implement it

Space management – “name and shame”

- Last resort – only thing left for over-quota’ed project areas
- Lists of big users
  - sometimes augmented with measures of ‘waste’
- Ineffective, except for small groups
- Relies on harnessing peer-group pressure
- Now deprecated – ‘bullying’
- Few options left for data/project space
- CSIRO moving to abolish such spaces on HPC systems – Data Access Portal

Policy for (shared) storage areas
- space management – "Name and shame"

For filesystem /flush, in the last 183 days.

directory	Accessed or modified files	Accessed or modified Gbyte	% of	used Gbyte	untouched Gbyte
/flush/root	2	0	0.0	30452	30452
/flush/user01	44	174	5.2	3373	3198
/flush/user02	447	327	11.7	2805	2478
/flush/user03	14365	3218	73.2	4394	1176
/flush/user04	285	2101	88.8	2367	265
/flush/user05	9827	1821	94.0	1937	116
/flush/user06	118684	4088	99.6	4104	15
/flush/user07	316	4361	100.0	4362	0
/flush/user08	11047	4238	100.0	4238	0
/flush/user09	667	1973	100.0	1973	0

Conclusion

- Policies for storage
  - necessary for users, systems staff and management
  - range of options: maximise the value of the resources
  - need to communicate the policies (beforehand!)
- Implementing policies
  - necessary, to avoid disasters and wastage
  - tends to be over-looked
  - disasters in waiting (users' ignorance and complacency – un-protected files systems), masked by reliable hardware (mostly)
  - mustn't add to the disasters!

Thank you

CSIRO IMT Scientific Computing
Robert C. Bell
CSIRO HPC National Partnerships

t +61 3 9545 2368

e Robert.Bell@csiro.au

w www.csiro.au

CSIRO SC Systems Guide

File systems

The filesystem setup on Ruby follows SC filesystem conventions.

To access Ruby folders from Windows, the table below lists the mapping between the Linux and Windows environments. Note that the data and flush areas are shared by Bragg, Pearcey and Ruby.

Ruby variable name	Share (NEXUS)
$HOME	\\ruby.hpc.csiro.au\yourident
$FLUSHDIR or $FLUSH1DIR	\\ruby.hpc.csiro.au\FLUSH1DIR
$FLUSH2DIR	\\ruby.hpc.csiro.au\FLUSH2DIR
$DATADIR	\\ruby.hpc.csiro.au\DATADIR
$STOREDIR	\\ruby.hpc.csiro.au\yourident

CSIRO SC Data Store Guide

Backups

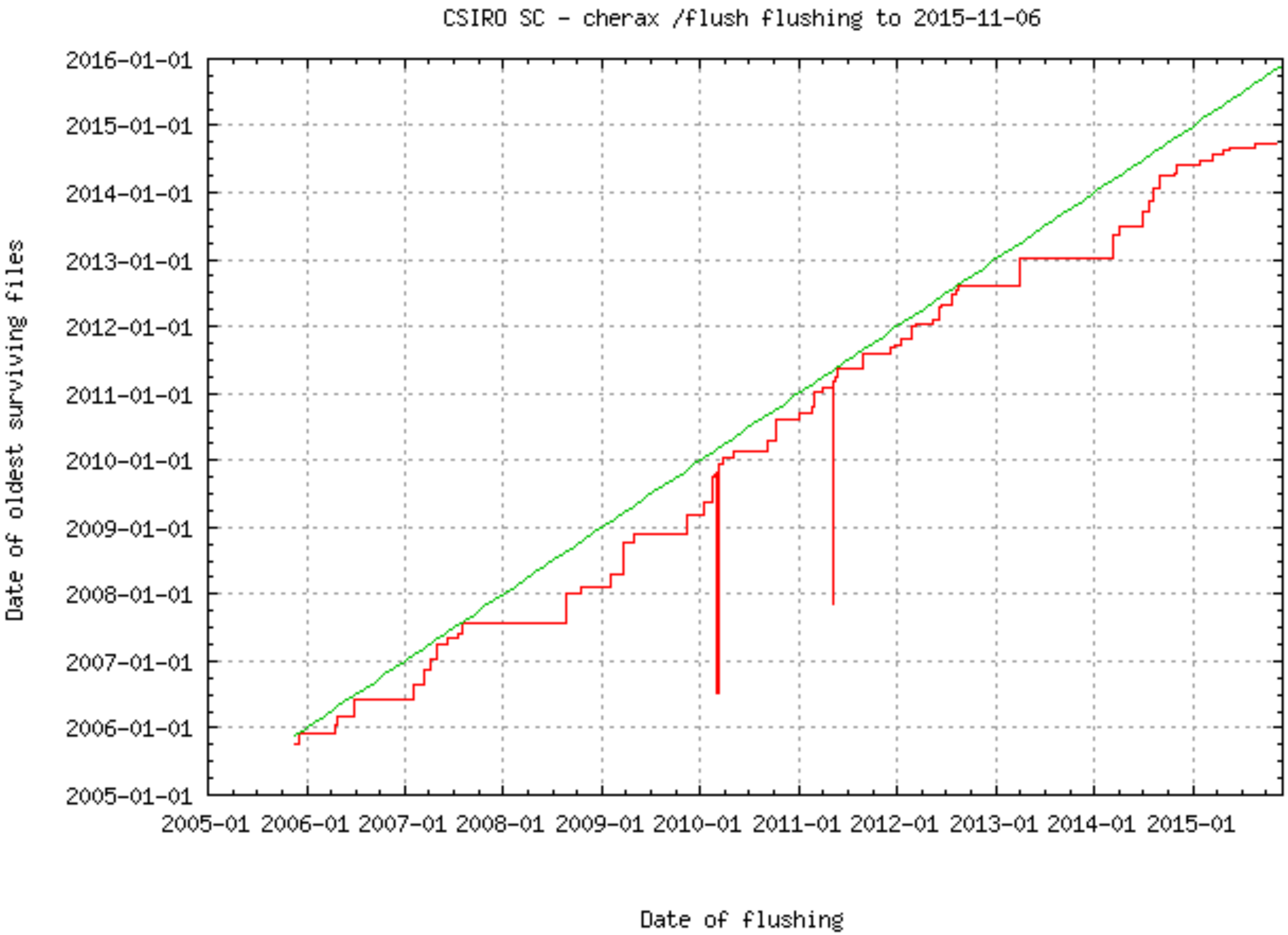
updated Nov 19, 2015 by Bell, Robert (IM&T, Docklands)

On the CSIRO SC systems, in general only the `$HOME` areas are subject to backup. This allows restoration of files that accidentally deleted or corrupted if:

- they have existed for long enough to be copied to a backup, and were not open for writing at the time of the backup
- and support staff are notified promptly.

There are circumstances where backups are difficult to do, and we do not have complete coverage. Here are some examples.

- Backups are not made of open files. For example, databases are typically open all the time. You need to make special arrangements to have databases backed-up - for example by running a snapshot utility.


CSIRO SC - cherax /flush flushing to 2015-11-06
Date of oldest surviving files
2016-01-01
2015-01-01
2014-01-01
2013-01-01
2012-01-01
2011-01-01
2010-01-01
2009-01-01
2008-01-01
2007-01-01
2006-01-01
2005-01-01
2005-01 2006-01 2007-01 2008-01 2009-01 2010-01 2011-01 2012-01 2013-01 2014-01 2015-01
Date of flushing

CSIRO SC Users Guide

Please carefully consider how to manage your data when using SC systems. Files can be kept in long-term storage on the datastore (especially large and consolidated files). Home directories are backed up but have limited space available (except for Ruby/datastore). High performance working space for files is available but only for short and medium term use. Copies of critical files should be maintained in multiple geographically separate locations where possible.

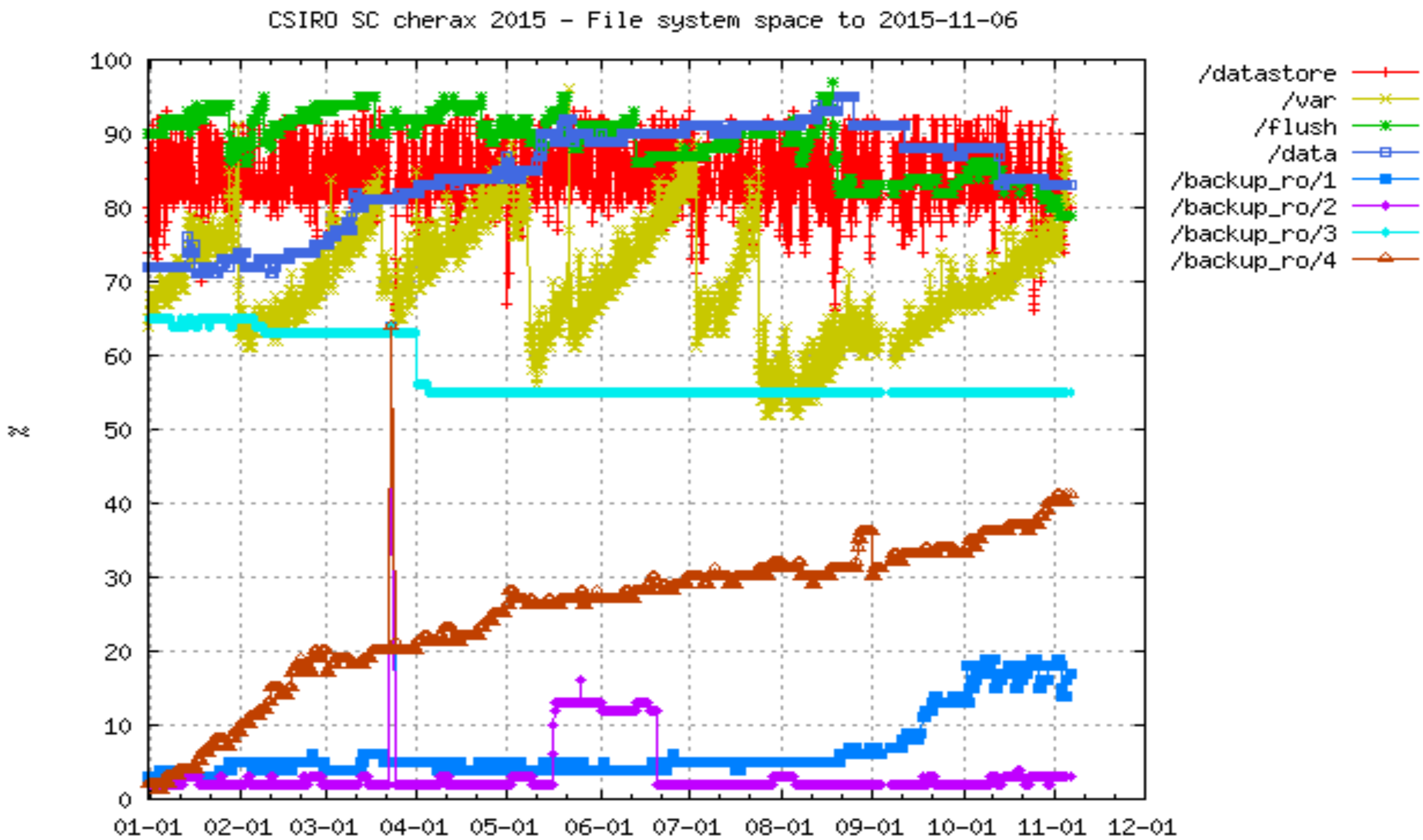
The SC storage is about as good as it gets but is still not immune to disaster. In particular most of the hardware is on one site. Only a limited subset of backed-up content gets duplicated to remote sites.

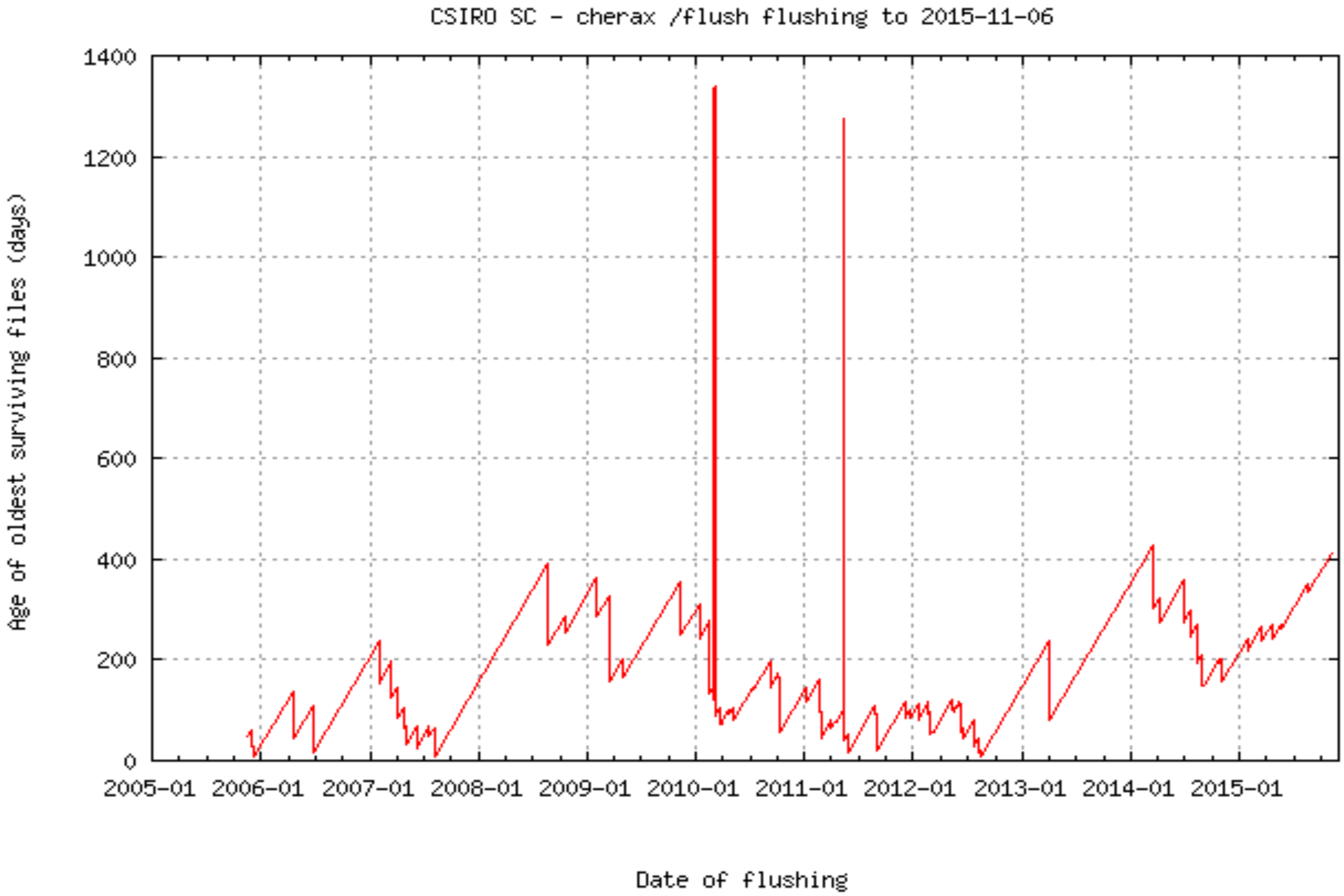
(But now dual-site DMF!)

CSIRO SC Users Guide

Understanding file systems and data management

Information on the file system structure of the SC systems is located on the "File systems" section in the System Guides. It is important to read the Data Store section to get an understanding of SC's data store policies and how large amounts of data are managed. In particular, with some directories like $FLUSHDIR and $TMPDIR, where data can be purged regularly, users need to save important data elsewhere.


CSIRO SC cherax 2015 - File system space to 2015-11-06
%
100
90
80
70
60
50
40
30
20
10
0
01-01
02-01
03-01
04-01
05-01
06-01
07-01
08-01
09-01
10-01
11-01
12-01
/datastore
/var
/flush
/data
/backup_ro/1
/backup_ro/2
/backup_ro/3
/backup_ro/4


CSIRO SC - cherax /flush flushing to 2015-11-06
Age of oldest surviving files (days)
1400
1200
1000
800
600
400
200
0
2005-01 2006-01 2007-01 2008-01 2009-01 2010-01 2011-01 2012-01 2013-01 2014-01 2015-01
Date of flushing