CS615 - Aspects of System Administration

Backup, Monitoring

Department of Computer Science
Stevens Institute of Technology
Jan Schaumann
jschauma@stevens.edu
https://www.cs.stevens.edu/~jschauma/615/

”The website is down...”

”The website is down...”

```
$ curl -I https://www.cs.stevens-tech.edu/~jschauma/615/
curl: (51) SSL: no alternative certificate subject name matches
        target host name ’www.cs.stevens-tech.edu’
```

”The website is down...”

```
$ curl -I https://www.cs.stevens.edu/~jschauma
HTTP/1.1 301 Moved Permanently
Date: Sat, 31 Mar 2018 21:09:57 GMT
Server: Apache
Location: https://www.stevens.edu/ses/cs/errors/404.html
Vary: Accept-Encoding
Content-Type: text/html; charset=iso-8859-1

$ curl -I https://www.stevens.edu/ses/cs/errors/404.html
HTTP/2 404
[...]
```

”The website is down...”

```
$ curl -I https://www.cs.stevens.edu/~jschauma
HTTP/1.1 301 Moved Permanently
Date: Sat, 31 Mar 2018 21:09:57 GMT
Server: Apache
Location: https://www.stevens.edu/ses/cs/errors/404.html
Vary: Accept-Encoding
Content-Type: text/html; charset=iso-8859-1

$ curl -I https://www.stevens.edu/ses/cs/errors/404.html
HTTP/2 404
[...]

$ ssh jschauma@git.srcit.stevens-tech.edu
jschauma@git.srcit.stevens-tech.edu’s password:
```

”The website is back up... ish”

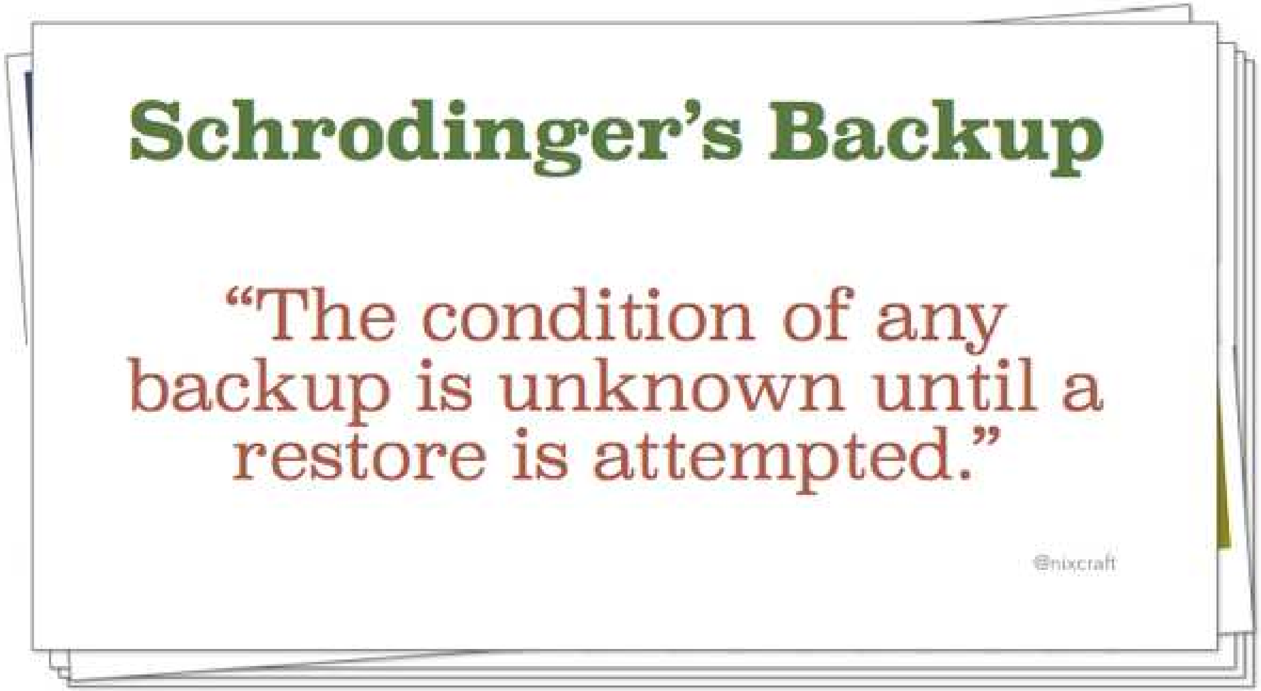
```
$ curl -I https://www.cs.stevens.edu/~jschauma/615/
HTTP/1.1 200 OK
Date: Sat, 31 Mar 2018 21:21:39 GMT
Server: Apache
Last-Modified: Tue, 25 Apr 2017 16:38:05 GMT
```

Backups vs. Restores

Backups are just a *means* to accomplish a specific *goal*:

To have the ability to restore data.

To the backups!



Schrodinger's Backup

"The condition of any backup is unknown until a restore is attempted."

@nixcraft

Backups and Restore Basics

When do we need backups?

- long-term storage / archival
- recover from data loss

Long-term storage

Long-term storage

Long-term storage

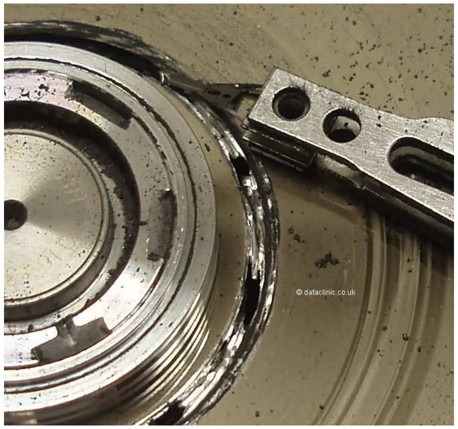
Long-term storage

- *full* set of level 0 backups
- separate set from regular backups
- usually stored off-site
- recovery / retrieval takes time
- limited granularity
- storage media considerations
- storage media transport considerations
- backup encryption and recovery key management

Backups and Restore Basics

When do we need backups?

- long-term storage / archival
- recover from data loss due to



Backups and Restore Basics

When do we need backups?

- long-term storage / archival
- recover from data loss due to

Backups and Restore Basics

When do we need backups?

- long-term storage / archival
- recover from data loss due to

Backups and Restore Basics

When do we need backups?

- long-term storage / archival
- recover from data loss due to

Backups and Restore Basics

When do we need backups?

- long-term storage / archival
- recover from data loss due to

Backups and Restore Basics

When do we need backups?

- long-term storage / archival
- recover from data loss due to
  - equipment failure
  - bozotic users
  - natural disaster
  - security breach
  - software bugs

Backups and Restore Basics

When do we need backups?

- long-term storage / archival
- recover from data loss due to
  - equipment failure
  - bozotic users
  - natural disaster
  - security breach
  - software bugs

Think of your backups as *insurance*: you invest and pay for it, hoping you will never need it.

Disaster Recovery

- loss of e.g. entire file system
- leads to downtime (of individual systems)
- RAID may help
- takes long time to restore
- may require retrieval of archival backups from long-term storage
- often involves *some* data loss

Disaster Recovery

- loss of e.g. entire file system
- leads to downtime (of individual systems)
- RAID may help
- takes long time to restore
- may require retrieval of archival backups from long-term storage
- often involves *some* data loss

Beware: disasters scale up much faster than your backup strategy!

File deletion recovery

Accidentally deleted files ought to be recoverable for a certain amount of time:

- ”Undo”
- time window and granularity requirements
- restore time, including
  - actual time spent restoring
  - waiting until resources permit the restore
  - staff availability
- self-service restore

But note: sometimes people *do* want to delete data and it be gone!

Filesystem backup

```
ssh ec2-instance "dump -u -0 -f - /" | bzip2 -c -9 >tmp/ec2.0.bz2
  DUMP: Found /dev/rxbd1a on / in /etc/fstab
  DUMP: Date of this level 0 dump: Mon Apr  2 19:34:30 2018
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rxbd1a (/) to standard output
  DUMP: Label: none
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 962609 tape blocks.
  DUMP: Volume 1 started at: Mon Apr  2 19:34:34 2018
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: 42.40% done, finished in 0:06
  DUMP: 83.38% done, finished in 0:01
  DUMP: 963445 tape blocks
  DUMP: Volume 1 completed at: Mon Apr  2 19:46:38 2018
  DUMP: Volume 1 took 0:12:04
  DUMP: Volume 1 transfer rate: 1330 KB/s
  DUMP: Date of this level 0 dump: Mon Apr  2 19:34:30 2018
  DUMP: Date this dump completed:  Mon Apr  2 19:46:38 2018
  DUMP: Average transfer rate: 1330 KB/s
  DUMP: level 0 dump on Mon Apr  2 19:34:30 2018
  DUMP: DUMP IS DONE
```

Filesystem backup

```
$ cat /etc/dumpdates
/dev/rxbd1a      0 Mon Apr  2 19:34:30 2018
$ ssh ec2-instance "dump -u -i -f - /" | bzip2 -c -9 >tmp/ec2.1.bz2
  DUMP: Found /dev/rxbd1a on / in /etc/fstab
  DUMP: Date of this level i dump: Mon Apr  2 20:09:24 2018
  DUMP: Date of last level 0 dump: Mon Apr  2 19:34:30 2018
  DUMP: Dumping /dev/rxbd1a (/) to standard output
  DUMP: Label: none
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 25307 tape blocks.
  DUMP: Volume 1 started at: Mon Apr  2 20:09:33 2018
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: 25244 tape blocks
  DUMP: Volume 1 completed at: Mon Apr  2 20:09:50 2018
  DUMP: Volume 1 took 0:00:17
  DUMP: Volume 1 transfer rate: 1484 KB/s
  DUMP: Date of this level i dump: Mon Apr  2 20:09:24 2018
  DUMP: Date this dump completed:  Mon Apr  2 20:09:50 2018
  DUMP: Average transfer rate: 1484 KB/s
  DUMP: level i dump on Mon Apr  2 20:09:24 2018
  DUMP: DUMP IS DONE
```

Filesystem backup

```
$ rm /etc/resolv.conf # oops
$ restore -i -f /backups/ec2.0
...
```

Poor Man's Cloud Backup via tar(1)

Copying to a file system:

```
$ tar cf - data/ | ssh ec2-instance "tar -xf - -C /var/backups/$(date)"
```

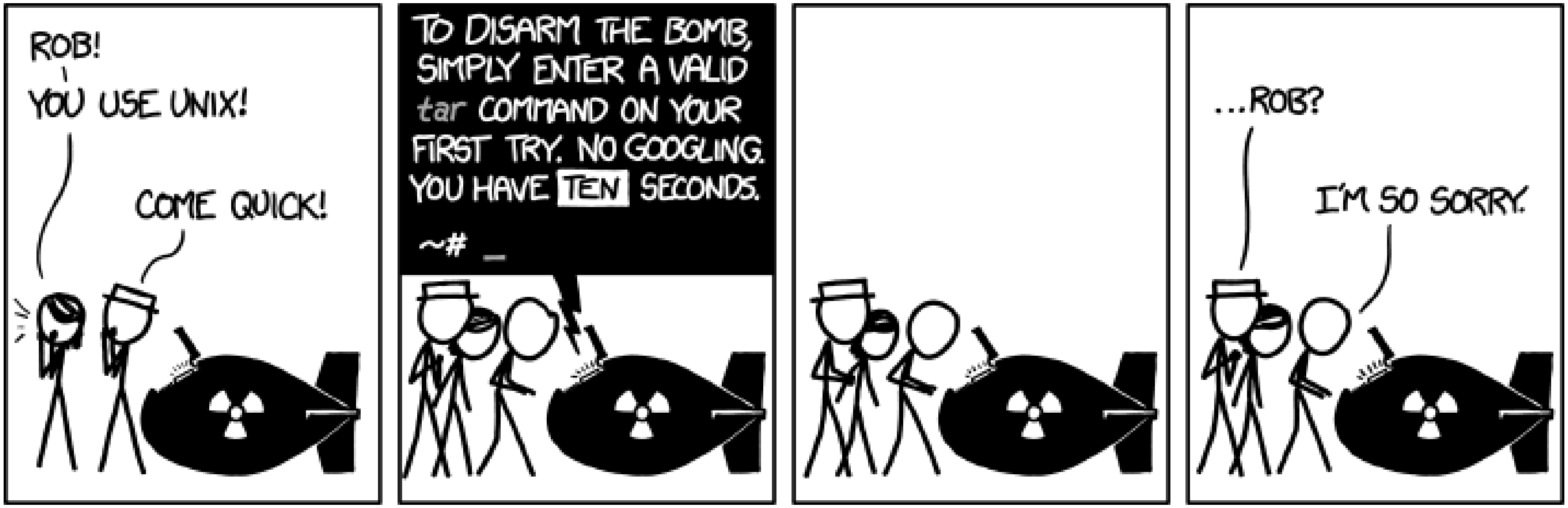
Writing to a block device, no filesystem necessary:

```
$ tar cf - data/ | ssh ec2-instance "dd of=/dev/rxb2a"
$ ssh ec2-instance "dd if=/dev/rxb2a" | tar tvf -
```

Encrypting along the way:

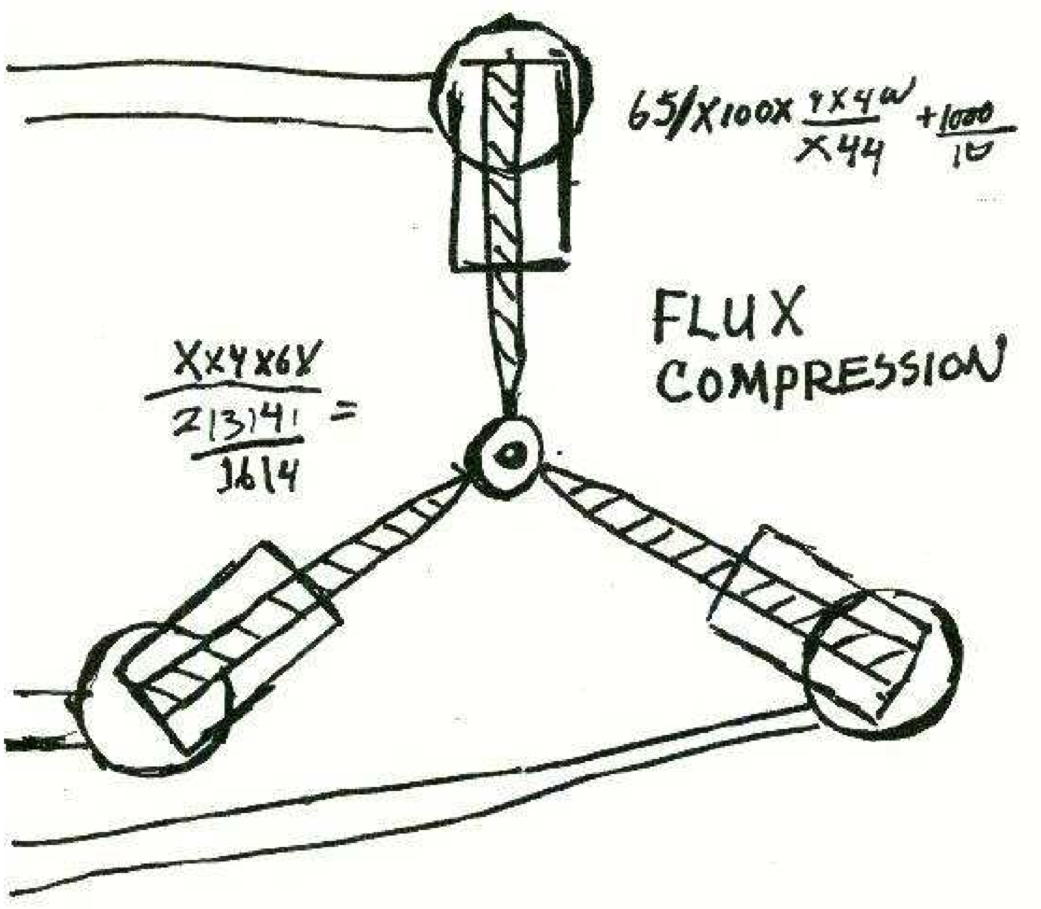
```
$ tar cf - data/ | gpg --encrypt -r recipient | ssh ec2-instance "dd of=/dev/rxb2a"
```

Know a Unix Command


ROB!
YOU USE UNIX!
COME QUICK!
TO DISARM THE BOMB, SIMPLY ENTER A VALID tar COMMAND ON YOUR FIRST TRY. NO GOOGLING. YOU HAVE TEN SECONDS.
~# _
...ROB?
I'M SO SORRY.

https://www.xkcd.com/1168/
https://www.cs.stevens.edu/~jschauma/615/tar.html

Filesystem backup



Filesystem backup

Filesystem backup

Filesystem backup

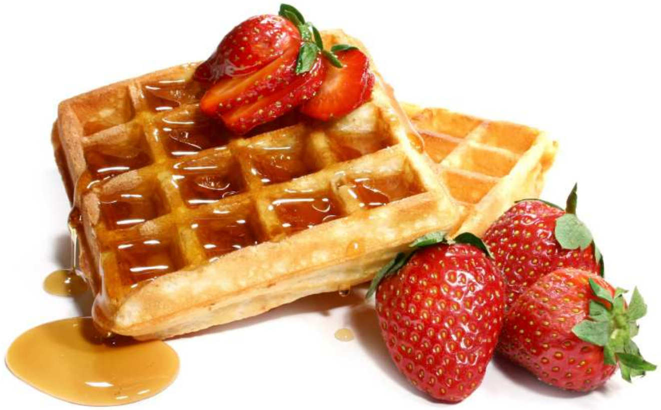
Example: Mac OS X “Time Machine”:

- automatically creates a full backup (equivalent of a ”level 0 dump”) to separate device or NAS, recording (specifically) last-modified date of all directories
- every hour, creates a full copy via *hardlinks* (hence no additional disk space consumed) for files that have not changed, new copy of files that have changed
- changed files are determined by inspecting last-modified date of directories (cheaper than doing comparison of all files’ last-modified date or data)
- saves hourly backups for 24 hours, daily backups for the past month, and weekly backups for everything older than a month.

Filesystem backup

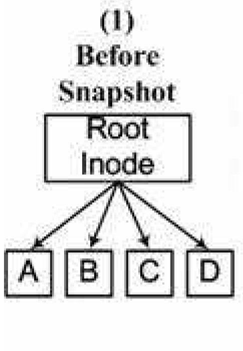
Example: WAFL (Write Anywhere File Layout)

- used by NetApp’s “Data ONTAP” OS
- a snapshot is a read-only copy of a file system (cheap and near instantaneous, due to CoW)
- uses regular snapshots (“consistency points”, every 10 seconds) to allow for speedy recovery from crashes



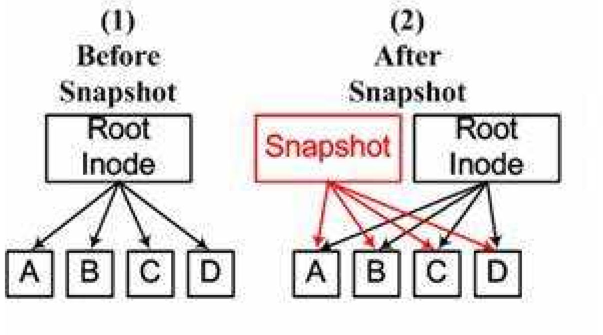
Filesystem backup

Example: WAFL (Write Anywhere File Layout)



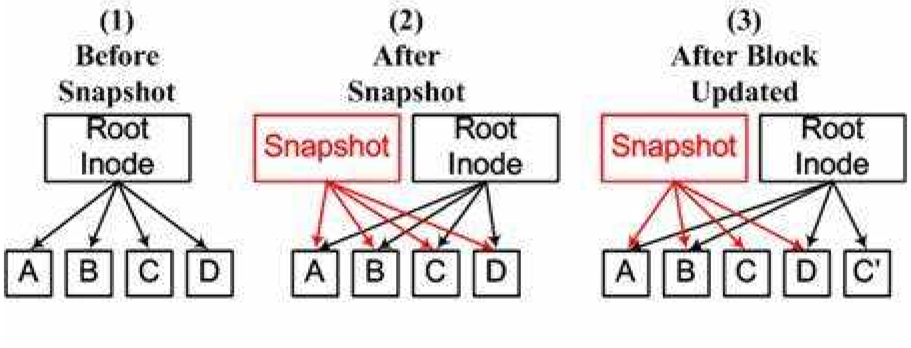
Filesystem backup

Example: WAFL (Write Anywhere File Layout)



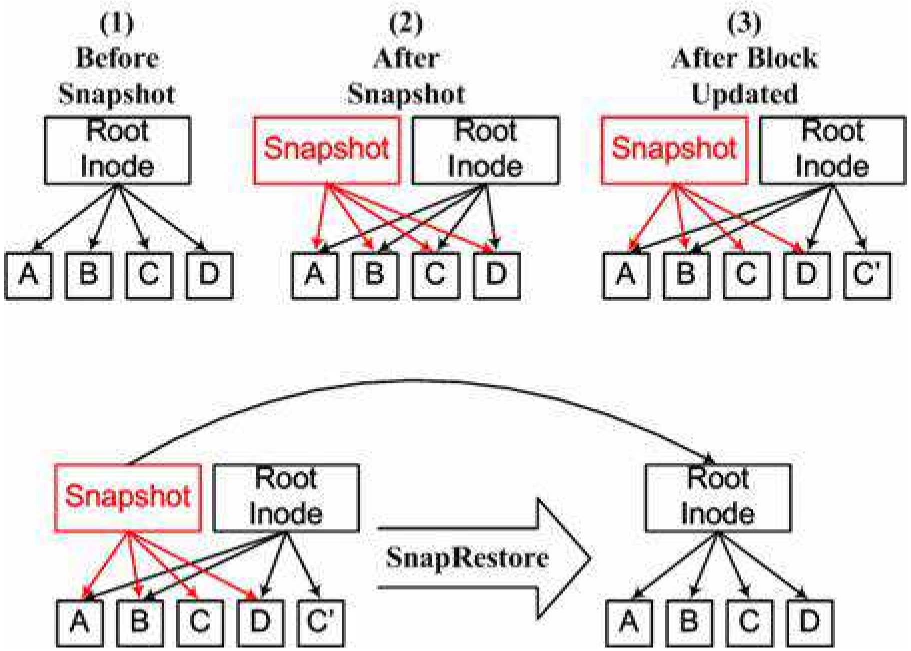
Filesystem backup

Example: WAFL (Write Anywhere File Layout)



Filesystem backup

Example: WAFL (Write Anywhere File Layout)



Filesystem backup

Example: ZFS snapshots

- ZFS uses a copy-on-write transactional object model (new data does not overwrite existing data, instead modifications are written to a new location with existing data being referenced), similar to WAFL
- a snapshot is a read-only copy of a file system (cheap and near instantaneous, due to CoW)
- initially consumes no additional disk space; the writable filesystem is made available as a “clone”
- conceptually provides a branched view of the filesystem; normally only the “active” filesystem is writable

ZFS Snapshots

```
$ pwd
/home/jschauma
$ ls -l .z*
ls: cannot access .z*: No such file or directory
$
```

ZFS Snapshots

```
$ pwd
/home/jschauma
$ ls -l .z*
ls: cannot access .z*: No such file or directory
$ ls -lid .zfs
1 dr-xr-xr-x 3 root root 3 Jan 10  2013 .zfs
$
```

ZFS Snapshots

```
$ pwd
/home/jschauma
$ ls -l .z*
ls: cannot access .z*: No such file or directory
$ ls -lid .zfs
1 dr-xr-xr-x 3 root root 3 Jan 10  2013 .zfs
$ ls -lai .zfs/snapshot
total 13
2 dr-xr-xr-x  4 root     root       4 Feb 28 21:00 .
1 dr-xr-xr-x  3 root     root       3 Jan 10  2013 ..
4 drwx--x--x 37 jschauma professor 88 Feb 24 22:32 amanda-_export_home_jschauma-0
4 drwx--x--x 37 jschauma professor 88 Feb 26 11:47 amanda-_export_home_jschauma-1
$
```

ZFS Snapshots

```
$ pwd
/home/jschauma
$ ls -l .z*
ls: cannot access .z*: No such file or directory
$ ls -lid .zfs
1 dr-xr-xr-x 3 root root 3 Jan 10  2013 .zfs
$ ls -lai .zfs/snapshot
total 13
2 dr-xr-xr-x  4 root     root       4 Feb 28 21:00 .
1 dr-xr-xr-x  3 root     root       3 Jan 10  2013 ..
4 drwx--x--x 37 jschauma professor 88 Feb 24 22:32 amanda-_export_home_jschauma-0
4 drwx--x--x 37 jschauma professor 88 Feb 26 11:47 amanda-_export_home_jschauma-1
$ cd .zfs/snapshot
$ echo foo > amanda-_export_home_jschauma-0/oink
-ksh: amanda-_export_home_jschauma-0/oink: cannot create [Read-only file system]
$ ls -laid . /
2 dr-xr-xr-x  4 root root    4 Feb 28 21:00 .
2 drwxr-xr-x 26 root root 4096 Jan 27 11:44 /
```

ZFS Snapshots

```
$ pwd
/home/jschauma/.zfs/snapshot
$ ls -lai amanda-_export_home_jschauma-0 >/tmp/a
$ ls -lai amanda-_export_home_jschauma-1 >/tmp/b
$ diff -bu /tmp/[ab]
--- /tmp/a 2014-03-01 22:55:49.000000000 -0500
+++ /tmp/b 2014-03-01 22:55:59.000000000 -0500
@@ -35,7 +35,7 @@
 57723 drwx------  3 jschauma professor         6 Dec 31 15:08 .subversion
 49431 -rw-------  1 jschauma professor         6 Dec 22 12:25 .sws.pid
    20 drwx------  2 jschauma professor         3 Jan 26 10:30 .vim
-61768 -rw-------  1 jschauma professor     14538 Feb 24 22:32 .viminfo
+61775 -rw-------  1 jschauma professor     14557 Feb 26 09:23 .viminfo
   173 -rw-------  1 jschauma professor      4355 Sep 17  2012 .vimrc
 45744 -rw-r--r--  1 jschauma professor         0 Jul 28  2013 .xsession-errors
    21 drwxr-xr-x  3 jschauma professor         6 Apr  4  2010 CS615A
$
```

Summary

- backups are most commonly done as incrementals of a filesystem, mountpoint, or directory hierarchy
- consider (long-term) storage:
  - media and location
  - increased storage requirements
  - privacy and safety of the data
- self-service restores and filesystem snapshots
- backups need to be:
  - regular, frequent, automated
  - invisible
  - verifiable
  - regularly tested

Hooray!

5 minute break

Problem Report

“Something’s wrong.”

Now what?

Problem Report

“The system feels slow.”

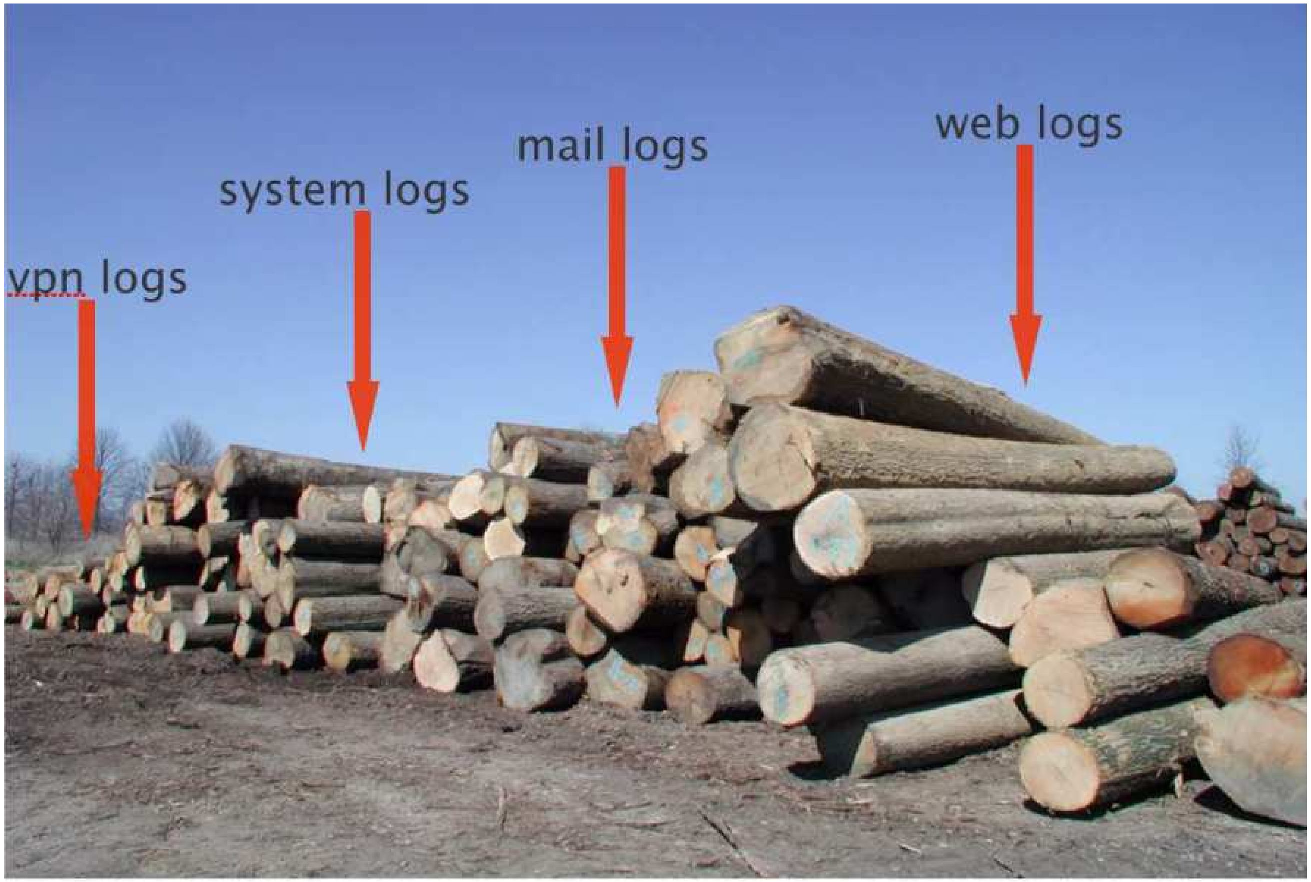
“I can’t log in.”

“My mail was not delivered.”

“The site is down.”

Now what?

To the logs!



Answers

“The system feels slow.”

```
up 1318 days, 13:46, 1 user, load averages: 993.81, 272.91, 1012.18
```

“I can’t log in.”

```
Apr 6 09:25:56 <auth.info>hostname sshd[1624]: Failed password for jdoe from
115.239.231.100 port 1047 ssh2
```

“My mail was not delivered.”

```
Apr 11 16:15:40 panix postfix/smtpd[7566]: connect from unknown[122.3.68.122]
Apr 11 16:15:41 panix postfix/smtpd[7566]: NOQUEUE: reject_warning: RCPT from
unknown[122.3.68.122]: 450 4.7.1 Client host rejected: cannot find your hostname,
[122.3.68.122]; from=<McneilRomany28@pldt.net> to=<jschauma@stevens.edu>
proto=ESMTP helo=<122.3.68.122.pldt.net>
```

Answers

“The site is down.”

```
94.242.252.41 - "" [11/Apr/2016:19:18:47 -0400] "GET /secret/ HTTP/1.1"
403 524 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:28.0)
Gecko/20100101 Firefox/28.0"
```

Answers

“The site is down.”

```
94.242.252.41 - "" [11/Apr/2016:19:18:47 -0400] "GET /secret/ HTTP/1.1"
403 524 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:28.0)
Gecko/20100101 Firefox/28.0"
```

Events

"Something's wrong." is just an *unexpected* or *undesirable* event.

Events

“Something’s wrong.” is just an *unexpected* or *undesirable* event.

Events happen all the time.

Events

“Something’s wrong.” is just an *unexpected* or *undesirable* event.

Events happen all the time.

Being able to identify *relevant* events allows you to diagnose, predict and even prevent *undesirable* events.

Events

In order to be able to identify an event as *unexpected*, you have to have *expected* events.

Expected Events

Know your applications.

Expected Events

Know your applications.

Know your users.

Expected Events

Know your applications.

Know your users.

Know your traffic patterns.

Expected Events

Know your applications.

Know your users.

Know your traffic patterns.

Know your systems.

Events and Metrics

```
$ dict event
  event
      n 1: something that happens at a given place and time
      2: a special set of circumstances; "in that event, the first
         possibility is excluded"; "it may rain in which case the
         picnic will be canceled" [syn: {event}, {case}]
```

```
$ dict metric
  metric
      3: a system of related measures that facilitates the
         quantification of some particular characteristic [syn:
         {system of measurement}, {metric}]
```

Events and Metrics

Event

Metric

You

Events and Metrics

Events

- may occur rarely / frequently / constantly
- can be collected in logs
- may be comprised of other events
- may be: something happened
- may be: nothing (new) happened

Metrics:

- correlation of related events
- may help identify outliers
- may trigger events
- may help make (automated or interactive) decisions

Collecting Data

Counters: easy, numeric data tracking individual events. Example: HTTP status codes

Timers: easy, numeric data tracking event duration. Example: Time to send all data for a successful HTTP request.

Thresholds: easy, numeric trigger for events; may itself trigger events or metrics. Example: more than N HTTP hits in X seconds yield 404.

Know Your Systems

Profile your application:

- execution time (for example: `time(1)`)
- data sources and destination affect execution
- `strace(1)` and friends for more detailed analysis

Understand your system performance:

- CPU load, memory (for example: `top(1)`, `vmstat(1)`)
- disk I/O (for example: `iostat(1)`)
- user activity (for example: `ac(1)`, `lsof(8)`, `sa(8)`)

Know Your Systems

Network statistics:

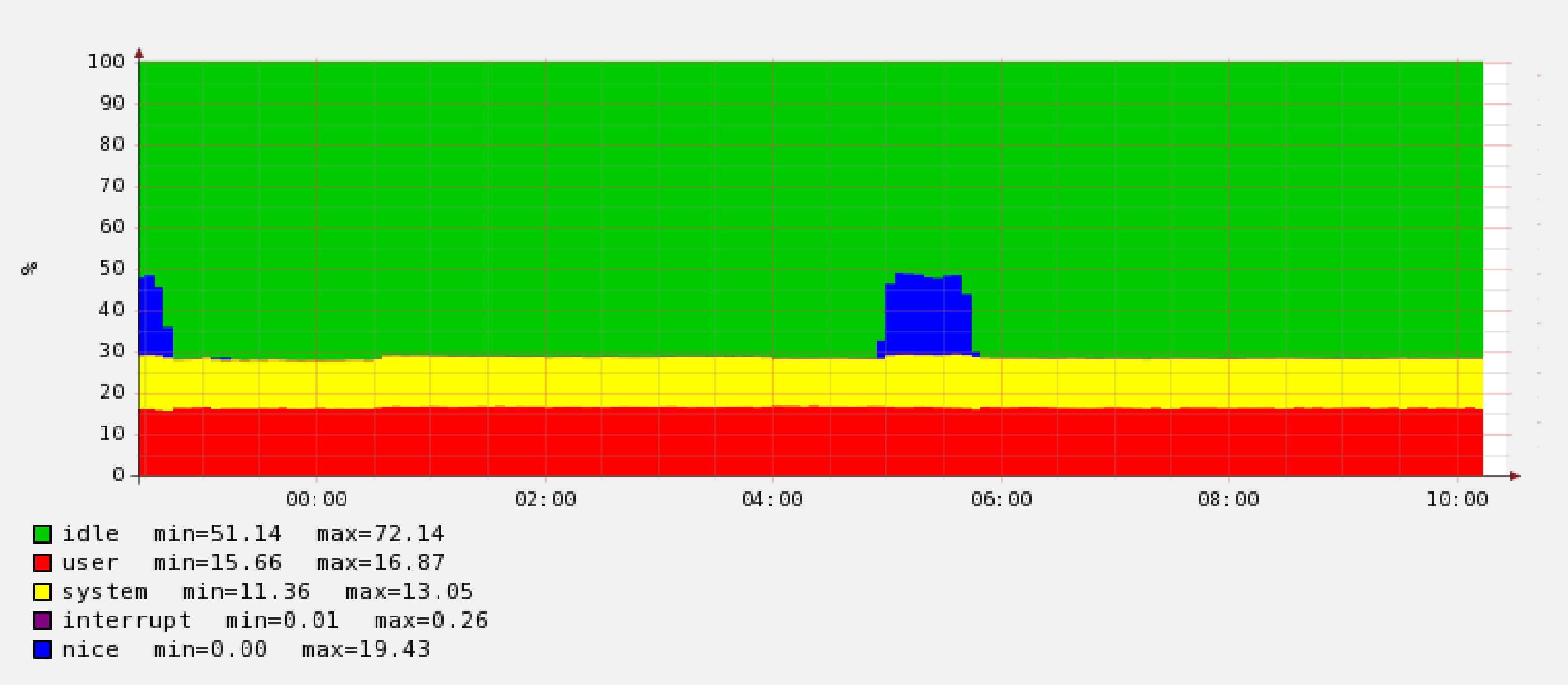
- ports and applications (for example: `lsof(8)`, `netstat(8)`)
- packets in and out
- connection origin
- *NetFlow* etc.

Context

Context lets you find *relevant* events in your haystack of metrics.

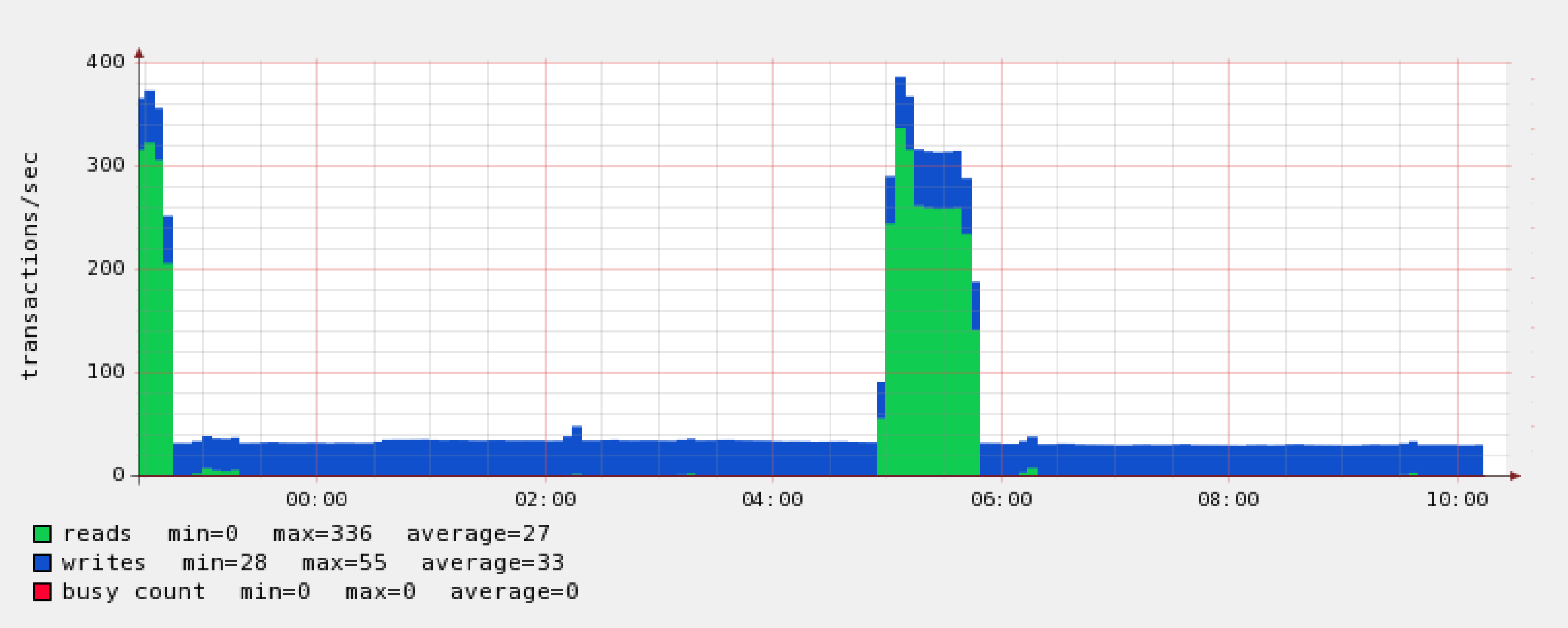
No context.

CPU load - 12 hours



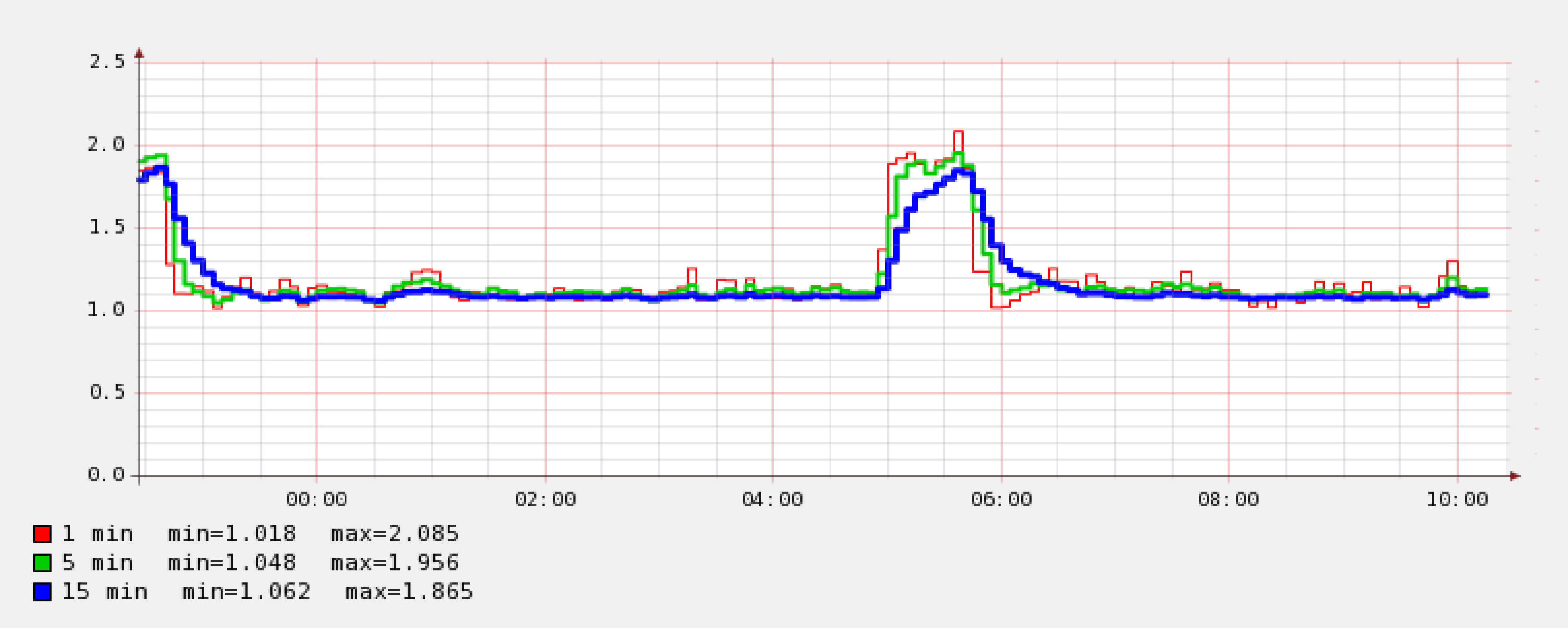
No context.

Disk I/O - 12 hours



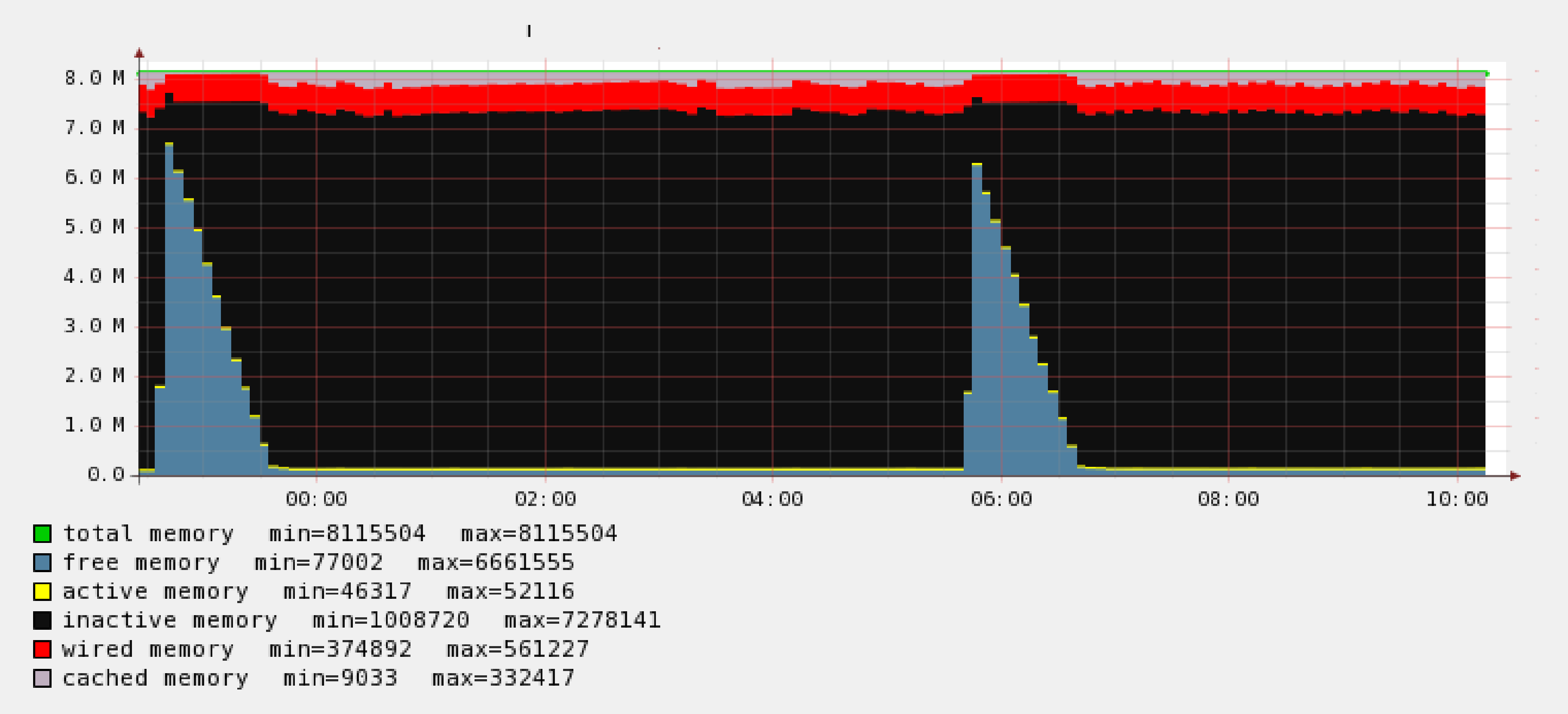
No context.

Load Average - 12 hours



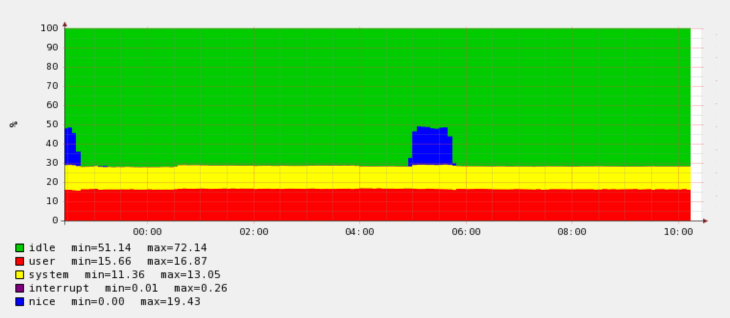
No context.

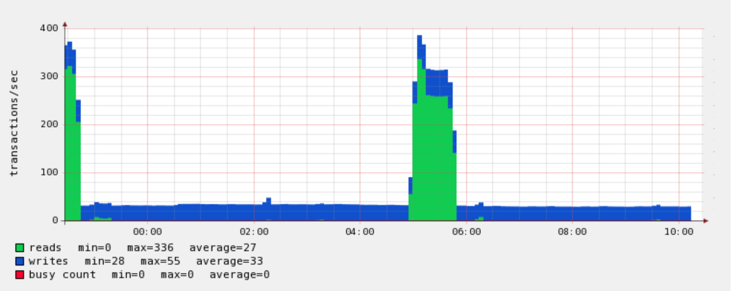
Memory - 12 hours

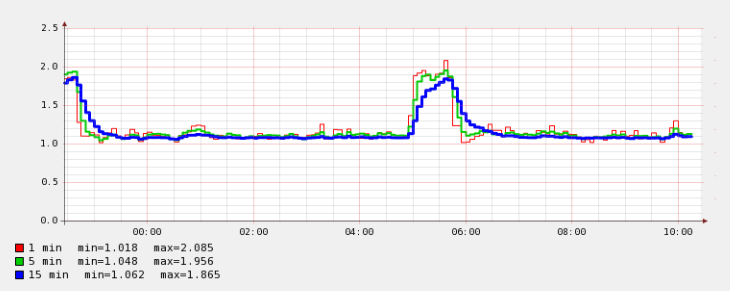


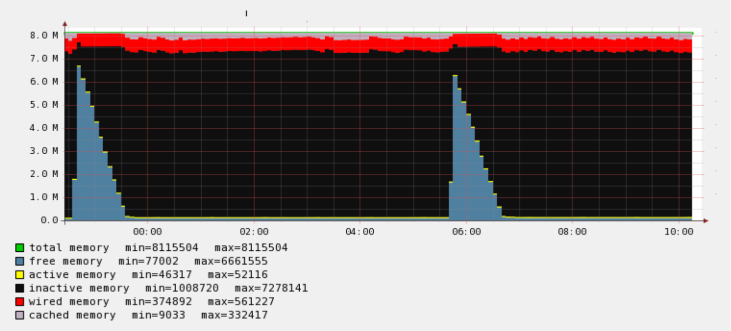
Some context.

12 hours



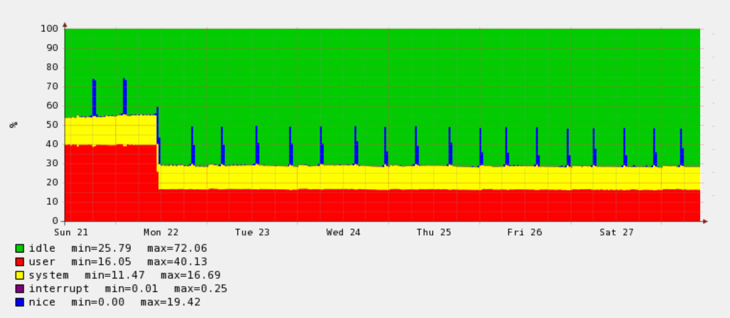


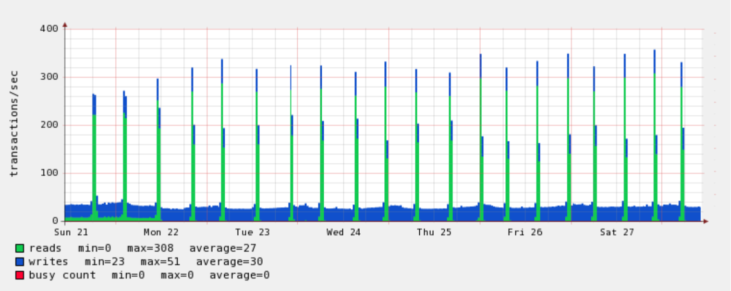


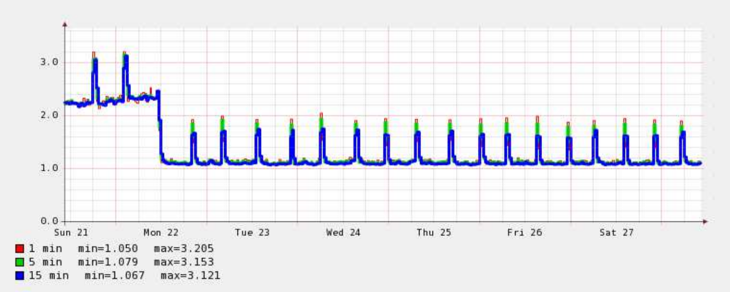


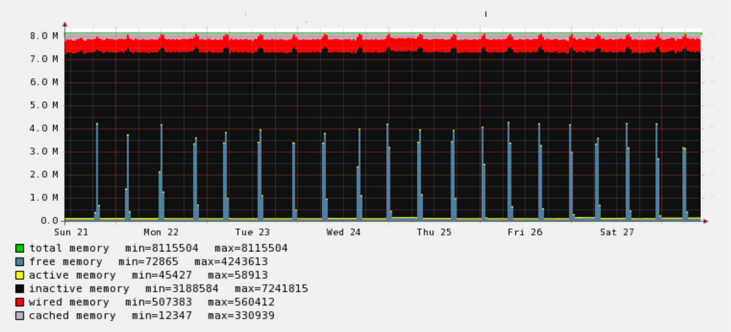
With context.

7 days



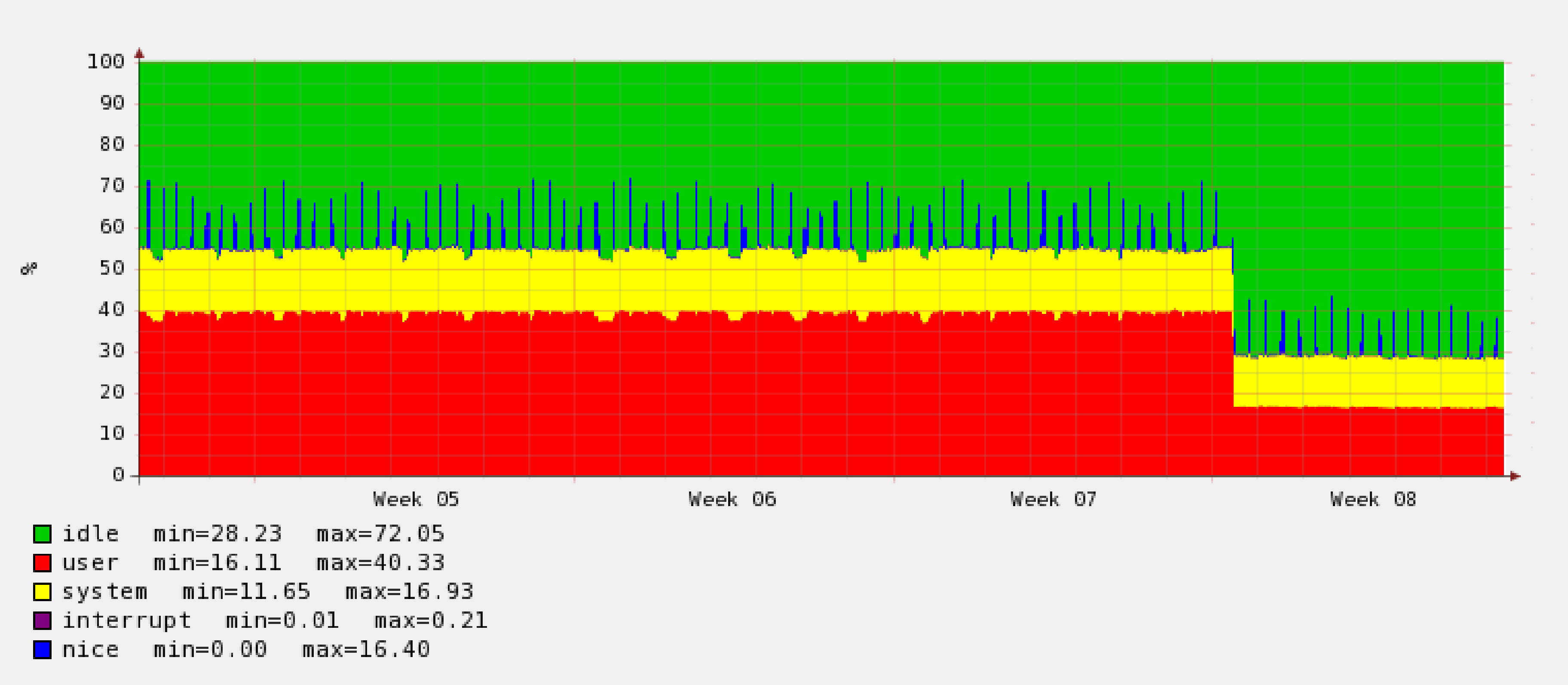






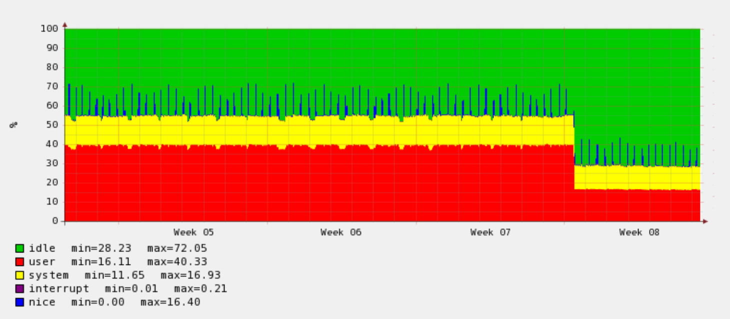
Know your systems.

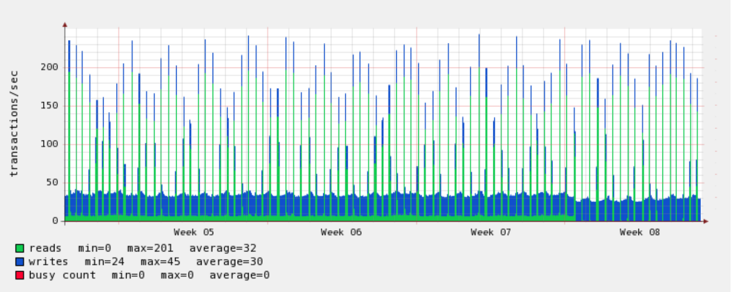
CPU load - 30 days

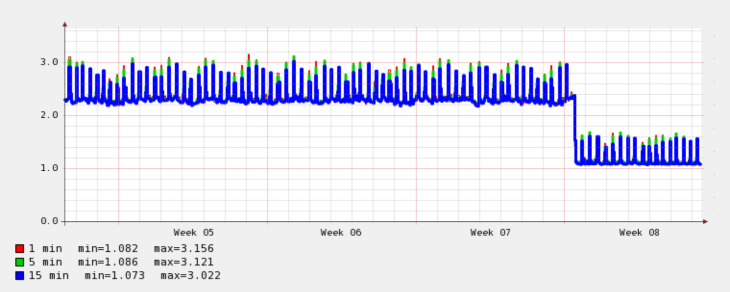


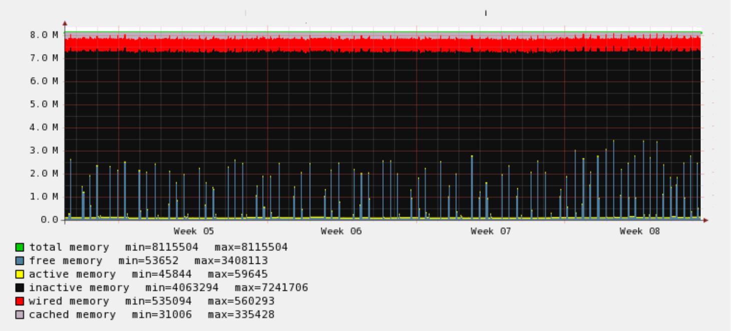
Know your systems.

30 days









Turn *events* into *metrics*.

- Log it!

- Export counters/timers from within your application.
- Process logs and produce counters/timers:

```
awk {print $9} /var/log/httpd/access.log | sort | uniq -c
```

- Graph it.
  `https://is.gd/tDCmQI`

Monitoring/graphing

SNMP based:

- Cacti: `http://www.cacti.net/`
- MRTG: `http://oss.oetiker.ch/mrtg/`
- Observium: `http://demo.observium.org/`
- ...

Other / complementary:

- Ganglia: `http://monitor.millennium.berkeley.edu/`
- Munin: `http://munin.ping.uio.no/`
- Nagios: `http://nagioscore.demos.nagios.com/`
- Graphite: `http://graphite.wikidot.com/`

To the cloud!

Theres a service for that. In the cloud.

Consider:

- support / convenience vs. do-it-yourself
- integration with your other services
- data confidentiality
- data lock-in (esp. when trending data over years)

Monitoring Pitfalls

Increasing the size of your haystack does not always help in finding the needle.

Monitoring Pitfalls

Increasing the size of your haystack does not always help in finding the needle.

Email is not a scalable network monitoring solution.

Monitoring Pitfalls

Increasing the size of your haystack does not always help in finding the needle.

Email is not a scalable network monitoring solution.

Absence of a signal can itself be a signal.

Monitoring Pitfalls

Increasing the size of your haystack does not always help in finding the needle.

Email is not a scalable network monitoring solution.

Absence of a signal can itself be a signal.

This list is incomplete.

Reading

Hurricane Sandy

- http://is.gd/aaxzvI
- http://is.gd/Y75pEA
- http://is.gd/32Az7y
- http://is.gd/FhAuFZ

Reading

Backups with `dump(8)` and `restore(8)`:

- `dump(8)` and `restore(8)`
- `https://is.gd/bXG9of`

Filesystem snapshots:

- `https://en.wikipedia.org/wiki/Snapshot_(computer_storage)`
- `https://en.wikipedia.org/wiki/Time_Machine_(Apple_software)`
- `http://comet.lehman.cuny.edu/jung/cmp426697/WAFL.pdf`

Book: `http://www.oreilly.com/catalog/unixbr/`

Reading

Monitoring:

- https://www.paperplanes.de/2013/3/28/monitoring-for-humans.html
- https://monitorama.com
- https://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html
- https://www.datadoghq.com/
- https://www.newrelic.com/
- https://www.elastic.co/products/logstash
- https://www.splunk.com/