CS615 - Aspects of System Administration

Backup, Monitoring

Department of Computer Science
Stevens Institute of Technology
Jan Schaumann
jschauma@stevens.edu
https://stevens.netmeister.org/615/
“Only wimps use tape backup. Just upload important stuff to FTP and let the rest of the world mirror it.” – *Linus Torvalds* (paraphrased)
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.”
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
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

Black Team link:

https://is.gd/zDB8WL
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!
T rusting your backups

A backup is a copy of the data. If the data is corrupt, your backup may become corrupt.

To restore data from a trusted backup, you can only use trusted tools.

Verify the authenticity of your backups!

See also: Red Team Link
https://is.gd/ZawHrG
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

```bash
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 -fr /usr/pkg

$ bzip2 -d -c ec2.1.bz2 | ssh ec2-instance "cd /; /sbin/restore xf -"

...
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

https://www.xkcd.com/1168/
https://stevens.netmeister.org/615/tar.html
### Filesystem considerations

Recall from Lecture 03 that our data can often be classified like so:

<table>
<thead>
<tr>
<th>shareable content</th>
<th>unshareable content</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static data</strong></td>
<td>/usr</td>
</tr>
<tr>
<td></td>
<td>/opt</td>
</tr>
<tr>
<td><strong>variable data</strong></td>
<td>/var/mail</td>
</tr>
<tr>
<td></td>
<td>/var/spool/news</td>
</tr>
<tr>
<td></td>
<td>/var/run</td>
</tr>
<tr>
<td></td>
<td>/var/lock</td>
</tr>
</tbody>
</table>

See also: [fstab(5)](https://stevens.netmeister.org/615/backup-exercise.html)
Backups vs. Snapshots

dump(8) preserves files (and file attribute), so that deletion of a file can be undone.

But what about intended file deletions in incremental backups?
Backups vs. Snapshots

dump(8) preserves files (and file attribute), so that deletion of a file can be undone.

But what about intended file deletions in incremental backups?

Enter rsync(1):

$ rsync -e ssh -az remote:/. backup/.
$ ssh remote
remote# pkg_add whatever
remote# rm some files
remote# exit
$ rsync -e ssh -az --delete remote:/. backup/.
Backups vs. Snapshots

dump(8) preserves files (and file attribute), so that deletion of a file can be undone.

But what about intended file deletions in incremental backups?

Enter rsync(1):

$ rsync -e ssh -az remote:/. backup/.
$ ssh remote
remote# pkg_add whatever
remote# rm some files
remote# exit
$ rsync -e ssh -az --delete remote:/. backup/.

But now we’ve lost the ability to restore a file we once intended to remove but then (much later) changed our mind about. Grrrr. I wish there was a way to simply go back in time to when the file still existed...
Filesystem backup

\[
\frac{X \times Y \times 6Y}{Z \times 3 \times 4} = \frac{6Z}{X \times 4Y}
\]

\[\text{FLUX COMPRESSION}\]
Filesystem backup
Filesystem backup
FileStream 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

$ start-omnios
$ ssh e2-instance
# snapshot syspool/ROOT/omnios@201904081035
# ls -la / | grep zfs
# ls -la /.zfs

```
total 6
  dr-xr-xr-x 4 root  root  4 May 3 2018 .
  drwxr-xr-x 23 root root 24 Apr 8 14:33 ..
  dr-xr-xr-x  2 root root  2 May 3 2018 shares
  dr-xr-xr-x  4 root root 4 Apr 8 14:35 snapshot
```

# rm /root/.ssh/authorized_keys
# echo oh no > /root/file
ZFS Snapshots

Restoring individual files

```
Only in /root/.: .bash_history
Common subdirectories: /root/./.ssh and /.zfs/snapshot/201904081035/root/./.ssh
Only in /root/.: file
Only in /.zfs/snapshot/201904081035/root/./.ssh: authorized_keys
# cp /.zfs/snapshot/201904081035/root/./.ssh/authorized_keys /root/.ssh/authorized_keys
```

Rolling back:

```
# rm /root/.ssh/authorized_keys
# zfs rollback syspool/ROOT/omnios@201904081035
# ls -l /root/.ssh/authorized_keys
-rw------- 1 root root 389 Apr 8 15:19 /root/.ssh/authorized_keys
# ls /root/file
/root/file: No such file or directory
```
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!

- system logs
- mail logs
- web logs
- vpn 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.”

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.”

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.
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

- **idle**: min=51.14, max=72.14
- **user**: min=15.66, max=16.87
- **system**: min=11.36, max=13.05
- **interrupt**: min=0.01, max=0.26
- **nice**: min=0.00, max=19.43
Disk I/O - 12 hours

- Reads: min=0, max=336, average=27
- Writes: min=28, max=55, average=33
- Busy count: min=0, max=0, average=0
Load Average - 12 hours
Memory - 12 hours

- Total memory: min=8115504 max=8115504
- Free memory: min=77002 max=6661555
- Active memory: min=46317 max=52116
- Inactive memory: min=1008720 max=7278141
- Wired memory: min=374892 max=561227
- Cached memory: min=9033 max=332417

Backup, Monitoring

April 8, 2019
Some context.

12 hours
7 days

Backup, Monitoring
Know your systems.

CPU load - 30 days

Backup, Monitoring
April 8, 2019
Know your systems.

30 days
Turn *events* into *metrics*.

- Log it!

- Export counters/timers from within your application.

- Process logs and produce counters/timers:
  
  ```bash
  awk '{print $9}' /var/log/httpd/access.log | sort | uniq -c
  ```

- Graph it.
  
  [https://is.gd/tDCmQI](https://is.gd/tDCmQI)
Monitoring/graphing

SNMP based:
- **Cacti**: [http://www.cacti.net/](http://www.cacti.net/)
- **MRTG**: [http://oss.oetiker.ch/mrtg/](http://oss.oetiker.ch/mrtg/)
- **Observium**: [http://demo.observium.org/](http://demo.observium.org/)
- ...

Other / complementary:
- **Ganglia**: [http://monitor.millennium.berkeley.edu/](http://monitor.millennium.berkeley.edu/)
- **Munin**: [http://munin.ping.uio.no/](http://munin.ping.uio.no/)
- **Graphite**: [http://graphite.wikidot.com/](http://graphite.wikidot.com/)

Backup, Monitoring

April 8, 2019
Context doesn’t explain everything...

...but it helps you look into what to investigate.
Context doesn’t explain everything...

...but it helps you look into what to investigate.
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.

Most of the value of your metrics only becomes evident over time.
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.

Most of the value of your metrics only becomes evident over time.

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:

- http://comet.lehman.cuny.edu/jung/cmp426697/WAFL.pdf

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

Monitoring:

- https://monitorama.com
- https://www.datadoghq.com/
- https://www.newrelic.com/
- https://www.elastic.co/products/logstash
- https://www.splunk.com/