Being Afraid - How Paranoia at Dropbox Protects your Data

David Mah
Dropbox
Trust
Trust!!
Me?
User Clients
Stateless Services

User Clients
Blockstore
Durability
Architecture
Architecture
Fears and Defenses
Architecture
Fears and Defenses
File
Key
Unique ID for a block.

Block
Blob of data on MB scale
API:
put(key, block)
get(key)
delete(key)
Blockstore Zone

PUT
DELETE
GET
Zone A

PUT

GET

DELETE

Storage Nodes

Frontends

Block Index
Architecture
Fears and Defenses
Fears
Moving things are scary
Moving things create accidents
What accidents are the scariest?
chance \times severity = danger
Alien attack?

not in lifetime x death → minor concern

probably
Operator reboots wrong server?

once a week? x node down → valid concern
Automation reimages wrong disks?

once a year? x data loss → HUGE concern
Biggest Fears
Biggest Fears

Software
Biggest Fears

Software
Hardware
Biggest Fears

Software
Hardware
Humans
Biggest Fears

Software
Hardware
Humans
Tooling/Automation
Combating fears?
Verify!
Protect!
Biggest Fears

Software
Hardware
Humans
Tooling/Automation
Biggest Fears

- Software
- Hardware
- Humans
- Tooling/Automation
- Protections
Biggest Fears

- Software
- Hardware
- Humans
- Tooling/Automation
- Protections
Fear of Software
Corruption inducing bugs...
Software bugs? Verify!
Testing is cool..
Bugs/Crashes?
Those are normal case
Story Time!
PUT Zone A PUT Zone D
# Replication Queue (MySQL)

<table>
<thead>
<tr>
<th>Index</th>
<th>Key to replicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>RRRRRRRRRRR</td>
</tr>
</tbody>
</table>
## Replication Queue (MySQL)

<table>
<thead>
<tr>
<th>Index</th>
<th>Key to replicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>RRRRRRRRRRRRR</td>
</tr>
<tr>
<td>1112</td>
<td>SSSSSSSSSSSSSSSS</td>
</tr>
</tbody>
</table>
### Replication Queue (MySQL)

<table>
<thead>
<tr>
<th>Index</th>
<th>Key to replicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>RRRRRRRRRRRRR</td>
</tr>
<tr>
<td>1112</td>
<td>SSSSSSSSSSSS</td>
</tr>
<tr>
<td>1113</td>
<td>TTTTTTTTTTTTT</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Replication Queue (MySQL)

<table>
<thead>
<tr>
<th>Index</th>
<th>Key to replicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>RRRRRRRRRRRRR</td>
</tr>
<tr>
<td>1112</td>
<td>SSSSSSSSSSSSSSS</td>
</tr>
<tr>
<td>1113</td>
<td>TTTTTTTTTTTTTTT</td>
</tr>
<tr>
<td>1114</td>
<td>UUUUUUUUUUUUUUU</td>
</tr>
</tbody>
</table>
Replication Queue (MySQL)

<table>
<thead>
<tr>
<th>Index</th>
<th>Key to replicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>RRRRRRRRRRR</td>
</tr>
</tbody>
</table>

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Replication Queue (MySQL)

<table>
<thead>
<tr>
<th>Index</th>
<th>Key to replicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>RRRRRRRRRRRRR</td>
</tr>
<tr>
<td>1112</td>
<td>SSSSSSSSSSSSSSS</td>
</tr>
</tbody>
</table>

## Replication Queue (MySQL)

<table>
<thead>
<tr>
<th>Index</th>
<th>Key to replicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>RRRRRRRRRRRRR</td>
</tr>
<tr>
<td>1112</td>
<td>SSSSSSSSSSSS</td>
</tr>
<tr>
<td>1114</td>
<td>UUUUUUUUUUUUU</td>
</tr>
</tbody>
</table>
### Replication Queue (MySQL)

<table>
<thead>
<tr>
<th>Index</th>
<th>Key to replicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>RRRRRRRRRRRRRRR</td>
</tr>
<tr>
<td>1112</td>
<td>SSSSSSSSSSSSSS</td>
</tr>
<tr>
<td>1113</td>
<td>TTTTTTTTTTTTTTT</td>
</tr>
<tr>
<td>1114</td>
<td>UUUUUUUUUUUUUUU</td>
</tr>
</tbody>
</table>
PUT Zone A

PUT Never Happens

Zone D
Series of Checkers
Can we actually get this off disk?
Can we actually get this off disk?

Do we have what we think we have?

Can we actually get this off disk?
Watcher
Can we serve the data?

Scanner
Do we have what we think we have?

Scrubber
Can we actually get this off disk?
## Replication Queue (MySQL)

<table>
<thead>
<tr>
<th>Index</th>
<th>Key to replicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>RRRRRRRRRRR</td>
</tr>
<tr>
<td>1112</td>
<td>SSSSSSSSSS</td>
</tr>
<tr>
<td>1114</td>
<td>UUUUUUUUUU</td>
</tr>
</tbody>
</table>
# Replication Queue (MySQL)

<table>
<thead>
<tr>
<th>Index</th>
<th>Key to replicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1111</td>
<td>RRRRRRRRRRRRRRR</td>
</tr>
<tr>
<td>1112</td>
<td>SSSSSSSSSSSSSSSSS</td>
</tr>
<tr>
<td>1113</td>
<td>TTTTTTTTTTTTTTTTT</td>
</tr>
<tr>
<td>1114</td>
<td>UUUUUUUUUUUUUUUUUUU</td>
</tr>
</tbody>
</table>
Software bugs? Protect!
Gradual Release Process
Software Release 1 (one week)

Stage Zone A
Stage Zone B

Prod Zone A
Prod Zone B
Prod Zone C
Prod Zone D
Software Release 1 (one week)

Prod Zone A

Stage Zone A

Stage Zone B

Prod Zone B

Prod Zone C

Prod Zone D
Stage Zone A
Stage Zone B
Prod Zone A
Prod Zone B
Prod Zone C
Prod Zone D
Software Release 1
Story Time!
Storage Node

Block Index
1. Snapshot State
1. Snapshot State
2. Analyze Snapshot
1. Snapshot State
2. Analyze Snapshot
3. Give Orders
1. Snapshot State
2. Analyze Snapshot
3. Give Orders
1. Snapshot State
2. Analyze Snapshot -New Data is PUT-
3. Give Orders
1. Snapshot State
2. Analyze Snapshot
   - New Data is PUT -
3. Give Orders
Purgatory and Trash
Delete

Purgatory (30 days)

Block Index

Storage Node
Storage
Node
Block
Index
Purgatory (30 days)
Trash (7 days)
Delete
Storage
Node
1. Snapshot State
2. Analyze Snapshot
   - New Data is PUT -
3. Give Orders

DELETE (bug)

Master

Storage Nodes

Block Index

Refresh
LOCK
1. Snapshot State
2. Analyze Snapshot
   - New Data is PUT -
3. Give Orders

Block Index

Master

Refresh LOCK

Trash

Storage Node

Storage Node

Storage Node

Storage Node

Storage Node
Software Bugs?

Verify!
Tests
Checkers

Protect!
Gradual Release
Purgatory+Trash
Biggest Fears

- Software
- Hardware
- Humans
- Tooling/Automation
- Protections
Fear of Hardware

It will crumble underneath me
Hardware Failure? Verify!
Pre Production Qualification
Production Machine Checkers
Hardware Failure? Protect!
Redundancy
Hardware Failure?

Verify!
Pre Production Qualification
In Production Checks

Protect!
Redundancy
Biggest Fears

- Software
- Hardware
- Humans
- Tooling/Automation
- Protections
Fear of Humans

Accidents
Story Time!
lifecycle=allocated

("live" servers)
lifecycle=allocated
("live" servers)
lifecycle=reinstall
(reprovisions, etc)
To rush upgrades:

gsh -q cache lifecycle=reinstall upgrade-host.sh
lifecycle=allocated
("live" servers)

lifecycle=reinstall
(reprovisions, etc)

To rush upgrades:

```
gsh -q cache lifecycle=reinstall upgrade-host.sh
```

Parsed (and ignored) as environment variable
All Affected, Site Outage

To rush upgrades:

gsh -q cache lifecycle=reinstall upgrade-host.sh
Human Error? Protect!
Distributed Shell Gating
dsh -q "storage-node" "upgrade-host"
dsh -q "storage-node" "upgrade-host"

ERROR:root:Exiting early because: Can't run DSH across multiple Blockstore zones.
Sudo Passwords
[/home/mah] → rm -rf /mnt
[/home/mah] → rm -rf /mnt
[sudo] password for mah:
Tomoyo
Tomoyo?
Linux Kernel module
System call access controls
[/mnt/blocks] ➔ sudo rm block-12345
rm: cannot remove `block-12345’: Operation not permitted
[/mnt/blocks] ➜ sudo rm block-12345
rm: cannot remove `block-12345': Operation not permitted

(strace output)
unlinkat(AT_FDCWD, "block-12345", 0)
= -1 EPERM (Operation not permitted)
Human Error?

Protect!

Distributed Shell Gates
Sudo Passwords
Tomoyo
Biggest Fears

Software
Hardware
Humans
Tooling/Automation
Protections
Fear of Automation

It will decide on its own to go reformat all of the hard drives
Tooling Bugs? Verify!
Tool/Automation Reports
Lifecycle of Tooling

Manual Labor
Scripts
Self driven automation
Tooling Bugs? Protect!
Lifecycle of Tooling

Manual Labor
Scripts
Human Authorized Execution
Self driven automation
Diagnosis
Diagnosis

PartitionTableInputOutputError
* Host: abc-de11-9f
* Reading /dev/sdam1’s partition table.
* Encountered IO error.
* read(3, 0xe3b600, 512) = -1 EIO (Input/output error)
Prescription
Prescription

This disk is unusable. Thus, DecommissionDisk
> bsctl osd decommission_disk abc-de11-9f 7037
Human Authorization
Human Authorization

If every diagnosis above checks out as reasonable.. Type 'yep, that evidence seems legit' to run these commands:
Replication-Dependent Gating
[/home/mah/] ➜ bsctl deallocate abc-de11-9f
bsctl deallocate abc-de11-9f

abc-de11-9f has 2917234 imperfectly replicated blocks. Aborting.
Please wait for replication before trying again.
Respecting Isolation
Automation Bugs?

Verify!
Automation Report

Protect!
Human Authorization
Replication Dependent Gating
Respect Isolation
Biggest Fears

Software
Hardware
Humans
Tooling/Automation
Protections
Fear of Protections

They’re going to brick the servers
They’re going to not actually work!
Protections bug? **Verify!**
Disaster Recovery Testing
Protections Bug? Protect!
Override Capabilities
override_tomoyo_policy.py --no_enforce
Failure in Protections?

Verify!
Disaster Recovery Testing

Protect!
Override Capabilities
Recap?
Find your fears
Verify!    Protect!
Embrace your paranoia

Verify!  Protect!

David Mah
mah@dropbox.com