Corpus collapsum
Partition tolerance of Galera put to test
RICON 2014

Raghavendra Prabhu

✉️ raghavendra.d.prabhu@gmail.com
Percona  ➡️ raghavendra.prabhu@percona.com
🐦 randomsurfer  🏡 wnohang.net  🟥 rdprabhu  ⚗️ ronin13
The Title?
Our Cluster
“’Network is reliable’ - a fallacy of the distributed system.”

“A distributed system is one in which the failure of a computer you didn’t even know existed can render your own computer unusable.” - Leslie Lamport

“Never attribute to malice that which is adequately explained by stupidity.” - Hanlon’s Razor

“Never attribute to Byzantine failure which can be explained by an ill node(s)” - Me
“’Network is reliable’ - a fallacy of the distributed system.”

“A distributed system is one in which the failure of a computer you didn’t even know existed can render your own computer unusable.” - Leslie Lamport

“Never attribute to malice that which is adequately explained by stupidity.” - Hanlon’s Razor

“Never attribute to Byzantine failure which can be explained by an ill node(s)” - Me
“ ’Network is reliable’ - a fallacy of the distributed system. ”
“A distributed system is one in which the failure of a computer you didn’t even know existed can render your own computer unusable. ” - Leslie Lamport
“Never attribute to malice that which is adequately explained by stupidity. ” - Hanlon’s Razor
“Never attribute to Byzantine failure which can be explained by an ill node(s) ” - Me
“’Network is reliable’ - a fallacy of the distributed system.”

“A distributed system is one in which the failure of a computer you didn’t even know existed can render your own computer unusable.” - Leslie Lamport

“Never attribute to malice that which is adequately explained by stupidity.” - Hanlon’s Razor

“Never attribute to Byzantine failure which can be explained by an ill node(s)” - Me
Introduction

Actors

- Database - WSREP/PXC
- Plugin - Galera
- Traffic control
  - Traffic Control - tc
  - NetEm
Introduction

Actors

- Database - WSREP/PXC
- Plugin - Galera
- Traffic control
  - Traffic Control - tc
  - NetEm
Actors

- Database - WSREP/PXC
- Plugin - Galera
- Traffic control
  - Traffic Control - tc
  - NetEm
Akkiranga Prabhu (Percona)

Introduction

Akkiranga Prabhu (Percona)

Corpus collapsum

28 October, 2014

8 / 58

Akkiranga Prabhu (Percona)

Corpus collapsum

28 October, 2014

8 / 58

Actors

- Containers - Docker
- Load
  - Generators - Sysbench, RQG
- Network
  - Dnsmasq
  - nsenter
Introduction

Actors

- Containers - Docker
- Load
  - Generators - Sysbench, RQG
- Network
  - Dnsmasq
  - nsenter
Actors

- Jenkins
  - Build flow and CI
- Storage
  - Why
But why

- The ‘P’ in CAP
- WAN scalability
- Real Reason - fun!
- Tolerance to latency variance
But why

- The ‘P’ in CAP
- WAN scalability
- Real Reason - fun!
- Tolerance to latency variance
But why

- The ‘P’ in CAP
- WAN scalability
- Real Reason - fun!
- Tolerance to latency variance
But why

- The ‘P’ in CAP
- WAN scalability
- Real Reason - fun!
- Tolerance to latency variance
But why

- Failures in warehouses.
- Not quorum, but consensus.
- Real world networks and synchronous replication
  - Delay
  - Partition
Galera

- Data-centric approach
- EVS
- Causality and Synchronous
- Latency
Where did it start
Where did it start

- Bug! [https://bugs.launchpad.net/galera/+bug/1274192](https://bugs.launchpad.net/galera/+bug/1274192)
- Loss of PC
- Crash
- HA goal
One can bring the whole down
The Flow
Basic Flow

Jenkins -> Build images -> Start Dnsmasq -> Bootstrap

nsenter/netem <-> Pre-sanity <-> SST/Others <-> Load/Sysbench
Basic Flow

Jenkins → Build images → Start Dnsmasq → Bootstrap

nsenter/netem ← Pre-sanity ← SST/Others ← Load/Sysbench
Basic Flow

Jenkins → Build images → Start Dnsmasq → Bootstrap

nsenter/netem ← Pre-sanity ← SST/Others ← Load/Sysbench
Basic Flow

- Jenkins
- Build images
- Start Dnsmasq
- Bootstrap

- nsenter/netem
- Pre-sanity
- SST/Others
- Load/Sysbench
Basic Flow

1. Jenkins
2. Build images
3. Start Dnsmasq
4. Bootstrap
5. Load/Sysbench
6. nsenter/netem
7. Pre-sanity
8. SST/Others

Corpus collapsed
Basic Flow

Jenkins → Build images → Start Dnsmasq → Bootstrap

Pre-sanity → nsenter/netem

Load/Sysbench → SST/Others
Basic Flow

1. Jenkins
2. Build images
3. Start Dnsmasq
4. Bootstrap
5. nsenter/netem
6. Pre-sanity
7. SST/Others
8. Load/Sysbench
Basic Flow

Jenkins → Build images → Start Dnsmasq → Bootstrap

nsenter/netem ← Pre-sanity ← SST/Others ← Load/Sysbench
Basic Flow

RR sysbench

Detach/Keep

Sanity check

Post sanity

Reconciliation

Core trace

Cleanup

Collect logs
Basic Flow

RR sysbench -> Detach/Keep -> Sanity check
                       -> Post sanity
                       -> Reconciliation
                       -> Collect logs
                       -> Cleanup
                       -> Core trace
Basic Flow

RR sysbench

Detach/Keep

Sanity check

Post sanity

Core trace

Reconciliation

Cleanup

Collect logs
Basic Flow

1. RR sysbench
2. Detach/Keep
   - Post sanity
   - Core trace
3. Sanity check
   - Reconciliation
   - Cleanup
   - Collect logs
Basic Flow

1. RR sysbench
2. Detach/Keep
3. Sanity check
4. Reconciliation
5. Post sanity
6. Core trace
7. Clean up
8. Collect logs
Basic Flow

- RR sysbench
- Detach/Keep
- Sanity check
- Post sanity
- Reconciliation
- Core trace
- Cleanup
- Collect logs
Basic Flow

RR sysbench

Detach/Keep

Sanity check

Post sanity

Reconciliation

Core trace

Cleanup

Collect logs
Basic Flow

1. RR sysbench
2. Detach/Keep
3. Sanity check
4. Post sanity
5. Reconciliation
6. Core trace
7. Cleanup
8. Collect logs
Empty your mind.

Be formless. Shapeless. Like water.

You put water into a cup,
It becomes the cup.

You put water into a bottle,
It becomes the bottle.

You put water into a teapot,
It becomes the teapot.

Water can Flow or it can Crash.
Parameters

- Sysbench
- Segment
- Reconciliation period
- Loss nodes
Parameters

- Sysbench
- Segment
- Reconciliation period
- Loss nodes
Parameters

- Sysbench
- Segment
- Reconciliation period
- Loss nodes
Parameters

- Sysbench
- Segment
- Reconciliation period
- Loss nodes
Parameters

- NetEm
- Detach loss
- Fsync
- Shutdown
Parameters

- NetEm
- Detach loss
- Fsync
- Shutdown
Parameters

- NetEm
- Detach loss
- Fsync
- Shutdown
Parameters

- NetEm
- Detach loss
- Fsync
- Shutdown
Containers!
Docker

- Why not virtualize
  - Occam
  - Namespaces

- Simplicity
  - Network
  - One application per node
Docker

- Portability
  - See same qualitative behavior that I do.

- Reproducibility
  - Makes it deterministic

- Configurable and CI
  - Byproducts
Docker

- QEMU and Docker
- Scalability
  - Performance
  - Feature
- Abstraction of channels
Container Networking

- Linking didn’t help
- Dnsmasq to rescue!
  - Hosts file and volumes
  - SIGHUP and refresh
- More elegant methods
  - Swarm
Noise

- Initial setup
  - Bridge
  - Egress only
  - IFB

- Present state

- NetEm
  - tc qdisc buckets
  - packet loss, delay, corruption, duplication, reordering
  - nsenter

- Future
  - Docker exec
Testing methods
Method I

- Qdisc is detached after load
- Objective
  - Time to recover of full cluster
- Done with a larger subset
Method II

► Qdisc is kept till the end
► Objective
  - Formation of primary component
► Comparatively smaller set
Observations

- Post sanity types
  - Why
- Which method is more pertinent
- State transfer issues
  - Beginning
  - During re-emergence
Observations

- Direct load to affected nodes
- Logs
  - journalctl
  - Streaming?
Other noises

- Aim
- Fsync
  - libeatmydata
  - Variance
- Correlation with network
- How with Docker
  - LD_PRELOAD
Load generation

- **Sysbench**
  - Generation
  - Reconnect on partition

- **Sockets chosen**
  - Load on affected nodes

- **Distribution of Load**
  - RR with socat
  - Native sysbench support
  - HAPerxy?
Load generation

- Nature of data/load
  - DDL
- RQG in future
  - Fuzz testing
The Fix

solution
Strike Out!
Eviction

- STONITH
- Permanent eviction
- ’N’ strikes & out!
  - Timers - evs parameters
  - wsrep_evs_delayed and wsrep_evs_evict_list
Eviction

- **Aim**
- **Quorum required**
  - Why?
  - Not shoot each other
  - Non-PC nodes also.
Eviction

- Aim
- Quorum required
  - Why? - Not shoot each other
  - Non-PC nodes also.
Eviction

- EVS version and upgrade
- TODO!
  - Ingress only
  - Follow here.
- Credits to Teemu Ollakka, Yan Zhang and Alex Yurchenko from codership.
Coredumps with Docker

- Breakdown of abstraction
- Lack of isolation
- What was done
  - Volumes
  - core_pattern & sysctl
  - suid and ulimit
WAN Segments

- How they work
- Random allocation
- Joiner starvation
- Simulates data center
- Donor selection
The code

- Github: https://github.com/percona/pxc-docker
- Jenkins: http://jenkins.percona.com/job/PXC-5.6-netem/
- Contributions/testing welcome!
- Dependencies
  - Sysbench
Code: todo

- Docker automated builds
- Orchestration
- Docker
  - Injection
  - Signal proxying
Code: todo

- Use Hoare’s channels - Go!
- Run it bare - CoreOS
- Overlay with etcd/fleet/libswarm
Future work
Future work

- Fault injection
  - Memory
    - Poisoned memory
  - Disk
    - libeatmydata
    - Opposite
    - ENOSPC
Fault injection

- CPU
  - NUMA?
  - Hotplug
- More network
  - corruption, duplication, reordering, rate-limit
  - Better distribution
  - Other shaping
More Chaos
Future work

- Disturb cluster more!
  - Membership changes
  * Manual eviction
  * Pull the cord!
  - Corrupt nodes

- Consistency voting
Further Reading

- Byzantine fault tolerance
  - Reaching agreement in presence of faults
- The Network is Reliable
- NetEm
- Latency: The New Web Performance Bottleneck
- Galera Cluster Documentation
- Auto eviction code
- Don’t Settle for Eventual Consistency
- Extended Virtual Synchrony
About

▶ /me: Raghavendra Prabhu, Product Lead, Percona XtraDB Cluster, Percona.
▶ Slides will be at slideshare.net/slidunder and owncloud
▶ About.me: raghavendra.prabhu
▶ Keybase.io: rdprabhu
▶ Presentation under CC BY-SA 4.0
Image Credits

- http://galeracluster.com/documentation-webpages/
- https://upload.wikimedia.org/wikipedia/commons/6/60/Corpus_callosum.png
- http://www.thebarrow.org/Neurological_Services/Epilepsy/204354
- https://secure.flickr.com/photos/brewbooks/7780990192
- https://www.flickr.com/photos/kwerfeldein/2649294869
- https://secure.flickr.com/photos/mindmob/51951632
- https://secure.flickr.com/photos/arenamontanus/2227769907
- https://www.flickr.com/photos/markop/477199204
- https://www.flickr.com/photos/gcwest/281385801
- https://www.flickr.com/photos/29233640@N07/13466208953
- https://www.flickr.com/photos/bob_in_thailand/9782777742/