Apache Cassandra Resilience

"If anything just cannot go wrong, it will anyway." - Murphy Law

Carlos Rolo - Cassandra Consultant

April 2015

DATASTAX

About me

- Pythian Cassandra Consultant since January 2015
- Working with Cassandra since 2011
- Working with distributed systems since 2010
- History:
  - Pythian
  - Leaseweb CDN
  - Portugal Telecom
  - DRI
- Twitter: @cjrolo

About Pythian

Remote data management services

Three main offerings
Managed Services
Consulting
DevOps

Engaged by IT and Operations executives looking to address skills or resourcing gaps

18 years in business
Growing at 30+% per year
400+ employees
250+ customers worldwide

Oracle Platinum Partner
Microsoft Gold Data Platform Partner
Microsoft Silver Cloud Competency Partner
AWS Advanced Consulting Partner
Datastax Gold Partner

Top 5% industry expertise
10 Oracle ACEs
2 Oracle ACE Directors
1 Oracle ACE Associates
6 Microsoft MVPs
2 Microsoft MCMs
1 Cloudera Champion of Big Data
1 DataStax Platinum Certified Consultant

Agenda

- The Setup
- Guaranteeing Reliability
- Testing
- The (BIG) problem
- Recovering
- Lessons Learned
- Q&A

The Setup

DATASTAX

- 15 Servers
  a. Bare metal, no virtualization
- 5 Datacenters
  a. Geographic Distant
  b. EU 1, EU 2, US East, US West, APAC1
  c. First 2 DC deployed with Cassandra 0.8!
- Spinning disks
  a. RAID 1
- RF 1 or 3 depending on Data

The Clients

DATASTAX

- Thrift
- HUGE write volume, Minimal read volume
    a. Reads could spike, still won't match writes
- Write Consistency = ONE
- Read Consistency = QUORUM
- The anti-patterns…
    a. Counters
    b. Batches
    c. Deletes

Guaranteeing Reliability

DATASTAX

- System was new, Cassandra was new…
- ... but the data was needed!
  a. HDD Redundancy, RAID 1
  b. Server Redundancy (RF=3)
  c. Datacenter redundancy (Base was 2)
  d. Several fallbacks programmed into the clients
  e. Raw data backups to different location
  f. And more!
- Something has to go REALLY wrong to make this unavailable

Testing

- First test was accidental…
  a. Lost link between DC's
  b. No issues… with Cassandra!
- After this…

Testing (2)

- Ok, now something serious!
- Use upgrades, maintenance windows, etc to test reliability
- The Good:
  a. Never had downtime
  b. App latency was affected, but still responsive
  c. Client fallbacks where perfect

Testing (3)

- The Bad:
  a. Commitlog replays
  b. Repairs
  c. Some I/O pressure
  d. Write load + compaction = Run out of HDD
- Learn and Prepare:
  a. “Fix” counter drift - Tricky but worked
  b. Repairs… We have to live with them!
  c. “Tweak” commitlogs settings
  d. Some changes to the clients

The (BIG) Problem

DATASTAX

- Too much confidence!
- Traffic was flowing nice!

But...

- The service was growing, the cluster was not.
- The cluster also had more work to do
  a. Map-Reduce and the likes
- But was still some room available, so, all fine!

“If everything seems to be going well, you have obviously overlooked something” - Murphy Law

What about upgrading?

DATASTAX

- Sometimes “add a node” is not that easy
    a. No vnodes…
    b. Geographic distributed is not all roses,
    c. You actually need physical room in the rack,
    d. Other constraints exist
        i. Power
        ii. Budget
        iii. Network Ports/Capacity
        iv. etc...

What Happened?

- 5 DC’s, 2 and 1/2 Down...
- System had a huge upgrade in capacity… not Cassandra
- Huge volume of incoming writes + Half of the cluster down…
- No longer a simple situation!

Brace for impact!

DATASTAX

- Assess the situation, define priorities…
- First try:
  a. Keep everything alive
  b. Didn't work…
     i. Massive OOM happening all-around the remaining nodes
     ii. Huge GC spikes
     iii. Remember the compaction + writes? Yup, running out of HDD space

Impact!

DATASTAX

- Clients where getting time-outs, traffic rerouted, heavy loaded servers got even more load…
- But…
  a. Some other fallbacks kicked in! Reducing load,
  b. Always 2 servers up in each (surviving) DC, writes still getting in,
  c. Read clients were highly tolerant to latency, so reads still being delivered

Everything counts!

- Delay writes to the HDD (dirty_ratio)!
- Tweak compaction!
- Delete commitlogs before restart!
    a. But sync the commitlog!
- Disable read_repair
- Reads… CL=1
- Non-Critical data… RF=1
- Map-Reduces were put on a hold
- Throttle traffic with IPTables

Recovering

- Once Datacenters start going online, clients redirected.
- Remove any “magic” inserted in the process
  a. Dirty_ratio, IPtables, etc...
- Enable read_repair
- Set Read back to QUORUM
- Disable compaction throttling
  a. Until SSTable count is nice
- Repair

Lessons Learned

DATASTAX

- Cassandra is highly reliable! As long as 1 server was up we knew that data would be there.
- It is (somewhat) predictable during failure.
- Lots of tweaks can be made during emergencies
- Use SSDs no matter what anybody tells you!
- Swap can save data!
    a. But keep swappiness low enough!

Q&A

DATASTAX

- Thanks for listening!
- Questions?


DATASTAX
Cassandra Days brought to you by DataStax