An Unexpected Open Source Win

Amy Tobey
Database Reliability Engineering Manager
I joined Tenable in September 2017
We have a multi-cloud mandate
DynamoDB → Cassandra
34 clusters
I ❤️ (some) Pickles

- Salty
- Spicy
- Sweet
- Vinegar
- import pickle
I ❤️ (many) Pickles

- Salty
- Spicy
- Sweet
- Vinegar
- import pickle
- Cassandra Consultants
THE LAST PICKLE
More on Pickles

- Configuration Analysis
- Cluster analysis
- Data Model Review
- ALL FOR ONE LOW LOW MODERATELY GOOD PRICE!
Use new token allocation for non bootstrap case as well.

Details
Type: Improvement  Status: RESOLVED
Priority: Major  Resolution: Fixed
Component/s: Core  Fix Version/s: 4.0
Labels: None

Description
There are couple reasons I think we should use the new token allocation for non bootstrap case as well.
Cassandra

- Cassandra 4.0 isn’t even in beta yet
- And even if it was, it’s a large complicated OSS project...
- Furthermore it’s Cassandra...
- Not getting near prod until 4.0.10 *minimum*
// (don't need to record Token here since it's still part of tokenToEndpointMap until it's done leaving)
private final Set<InetSocketAddress> leavingEndpoints = new HashSet<>();

// this is a cache of the calculation from {tokenToEndpointMap, bootstrapTokens, leavingEndpoints}
private final ConcurrentHashMap<String, Multimap<Range<Token>, InetSocketAddress>> pendingRanges = new ConcurrentHashMap<>();

private final ConcurrentHashMap<String, PendingRangeMap> pendingRanges = new ConcurrentHashMap<>();

// nodes which are migrating to the new tokens in the ring
private final Set<Pair<Token, InetSocketAddress>> movingEndpoints = new HashSet<>();

public InetSocketAddress getEndpoint(TokenType type)
{
    return sortedTokens;
}

private Multimap<Range<Token>, InetSocketAddress> getPendingRangesMM(String keyspaceName)
{
    Multimap<Range<Token>, InetSocketAddress> map = pendingRanges.get(keyspaceName);
    if (map == null)
Step 1: git-format-patch 5f3b35981d281

Step 2: git am < lol.patch

Step 3: ???????????????????!?!?!?!?!?!?!?!?!?!?!?!?!??!

Step 4: PROFIT
1. Create backport Jira
2. checkout cassandra
3. Maven (ivy) download internet
4. Export Eclipse data
5. Import Eclipse data into IntelliJ
6. Find code
7. Step through patch and manually apply
8. Run dtests
9. Prepare patch
10. Mail to mailing list
11. Wait
12. Get yelled at
13. Fix complaints
14. Send another patch
15. Wait
16. Patch accepted
17. Wait
18. Release!
19. Wait for packages
20. DEPLOY YAY!
Pickle to the Rescue!

Step 1: Ask the consultants to do it!
Step 1: Tell the consultants to do it!
Step 1: Pay the consultants to do it!
Step 2: PROFIT
Pickled Cassandra

- Started with a patched cassandra.jar
- Deployed to **new** clusters
- More waiting
Use new token allocation for non bootstrap case as well.

- Backport CASSANDRA-13088: Use new token allocation for non bootstrap case as well (CASSANDRA-14212)
- Remove dependencies on JENKINS classes from UnderCluster (CASSANDRA-14179)
- Add DEFAULT, UNSET, PBEAN and MEANS to 'ReservedKeywords' (CASSANDRA-14205)
- Add unittest for schema migration fix (CASSANDRA-14140)
- Print correct snitch info from nodetool describecluster (CASSANDRA-13528)
- Close socket on error during connect on OutboundTcpConnection (CASSANDRA-9636)
- Enable CDC unittest (CASSANDRA-14141)
- Acquire read lock before accessing CompactionStrategyManager fields (CASSANDRA-14139)
- Split CommittingStressTest to avoid timeout (CASSANDRA-14143)
- Avoid invalidating disk boundaries unnecessarily (CASSANDRA-14083)
- Avoid exposing compaction strategy index externally (CASSANDRA-14082)
- Prevent continuous schema exchange between 3.0 and 3.11 nodes (CASSANDRA-14109)
- Fix imbalanced disks when replacing node with same address with JBOID (CASSANDRA-14084)
- Reload compaction strategies when disk boundaries are invalidated (CASSANDRA-13948)
- Remove OpenJDK log warning (CASSANDRA-13916)
- Prevent compaction strategies from looping indefinitely (CASSANDRA-14079)
- Cache disk boundaries (CASSANDRA-13215)
- Add jar to build vel for maven builds (CASSANDRA-11108)
More Waiting 🕒
The Cassandra team is pleased to announce the release of Apache Cassandra version 3.11.2.

Apache Cassandra is a fully distributed database. It is the right choice when you need scalability and high availability without compromising performance.

http://cassandra.apache.org/

Downloads of source and binary distributions are listed in our download section:

http://cassandra.apache.org/download/

This version is a bug fix release[1] on the 3.11 series. As always, please pay attention to the release notes[2] and let us know[3] if you were to encounter any problem.

Enjoy!

[1]: (CHANGES.txt) https://goo.gl/m0jYnb
[2]: (NEWS.txt) https://goo.gl/NW60hU
[3]: https://issues.apache.org/jira/browse/CASSANDRA
Packages! 🐬

- Packages built & uploaded
- Update Ansible playbooks
- Run Ansible Playbooks
- Over and over and over and over.....
- PROFIT!
- Things suck less!
Local patches are awful for everyone
Work with shops that value OSS (Like TLP!)

@MissAmyTobey
TL;DR Part Deux

- TNSTAAFL
- Pay your partners to push patches upstream!
- Tip your wait staff!

@MissAmyTobey
Bye bye! 👋 (We’re hiring SREs & DBREs)

@MissAmyTobey
atobey@tenable.com