New backend for the AliEn File Catalogue

Miguel Martinez Pedreira

A Large Ion Collider Experiment

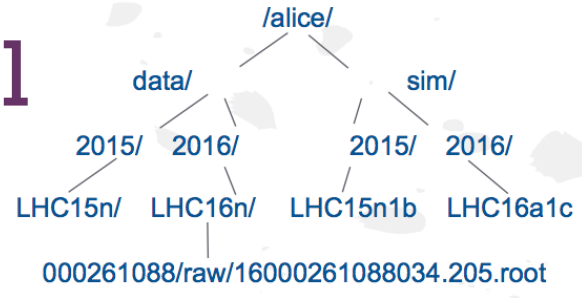
European Organisation for Nuclear Research

Current implementation

- MySQL-based AliEn File Catalogue
  - 3B logical entries
- One (powerful) DB master
  - 1.5TB RAM, 2.4TB on disk size
- DB slaves for hot standby / backups
  - 4h to dump, ~2 days to restore

	Machine status			Machine type						Disk	CPU utilisation (%)								Memory utilisation				
Machine	Online	Uptime	Load	Kernel	OS	Machine model	CPU	CPUs	MHz	Space	usr	sys	iow	int	sint	steal	nice	idle	Total	Used	Buffers	Cached	Free
1. db6c		20d 22:34	5.71	4.4.0-64-...	16.04	ProLiant DL380 Gen9	Xeon E5-2667 v4 3.20GHz	32	3500		9.981	1.15	0.276	0	0.365	0	0	88.23	1.476 TB	213.1 GB	226.5 MB	1.264 TB	3.989 GB
Total								**32**											**1.476 TB**	**213.1 GB**	**226.5 MB**	**1.264 TB**	**3.989 GB**

Catalogue in a nutshell



- LFN namespace
  - **/alice/data/2016/LHC16n/000261088/raw/16000261088034.205.root**
  - 1180 tables (max 50M), 3B entries, namespace split into tables
  - Metadata

```
-rwxr-xr-x  alidaq  alidaq  264403565  Sep 09 22:10 0f24bce32446ea22840d188e035b11a9
```

- GUID namespace
  - 76CEBD12-76A0-11E6-9717-0D38A10ABEEF
  - 173 tables (max 210M), 2.8B entries, split by time intervals (append)
  - Version 1 UUIDs (MAC+timestamp)

- Physical File Pointers

root://alice-tape-se.gridka.de:1094//10/33903/76cebd12-76a0-11e6-9717-0d38a10abeef
root://voalice10.cern.ch//castor/cern.ch/.../16000261088034.205.root

  - 3.5B entries, 1B physical files, pointers to ZIP members, 70PB over 70 Storage Elements

Catalogue in a nutshell

- LFN namespace
  - **/alice/data/2016/LHC16n/000261088/raw/16000261088034.205.root**
  - 1180 tables (max 50M), 3B entries, namespace split into t
  - Metadata

-rwxr-xr-x alidaq alidaq 264403565 Sep 09 22:10 0f24bce32

- GUID namespace
  - 76CEBD12-76A0-11E6-9717-0D38A10ABEEF
  - 173 tables (max 210M), 2.8B entries, split by time interva
  - Version 1 UUIDs (MAC+timestamp)

- Physical File Pointers

root://alice-tape-se.gridka.de:1094//10/33903/76cebd12-76a0-11e6-9717-0d38a10abeef
root://voalice10.cern.ch//castor/cern.ch/.../16000261088034.205.root

  - 3.5B entries, 1B physical files, pointers to ZIP members, 70PB over 70 Storage Elements

DB query rates

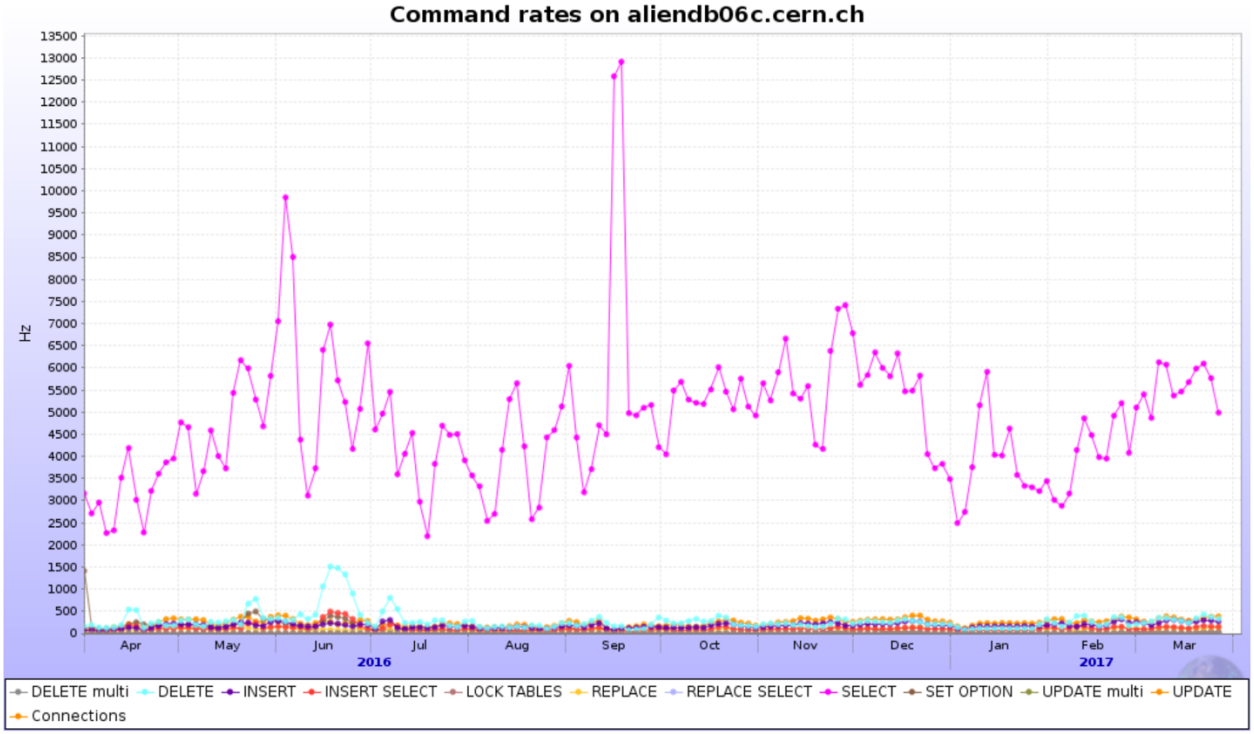
- Averages (1y)

  9223 Hz Reads
  618 Hz Changes
  282 Hz Deletes

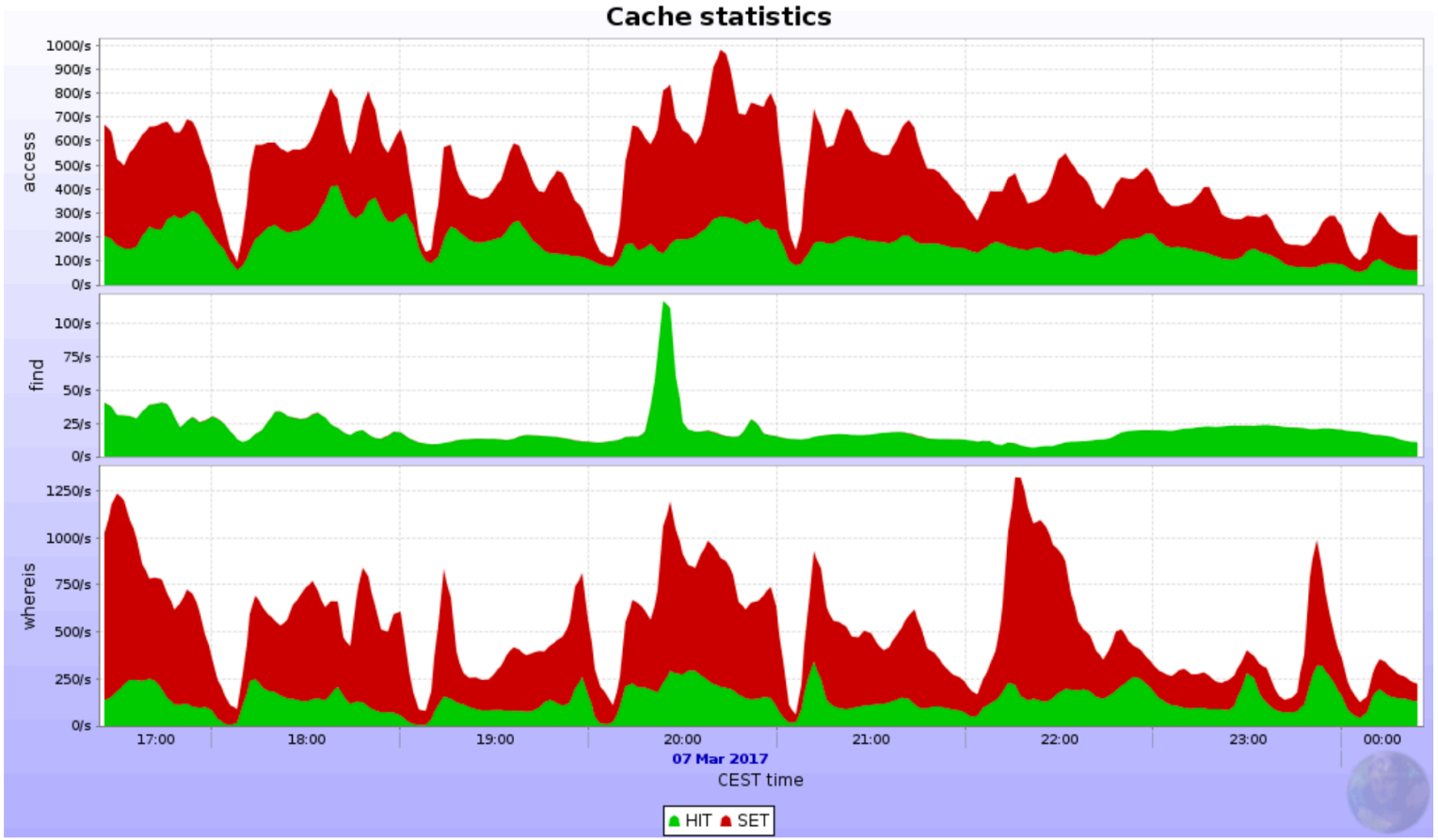
77500 running jobs

15:1 select/change ratio

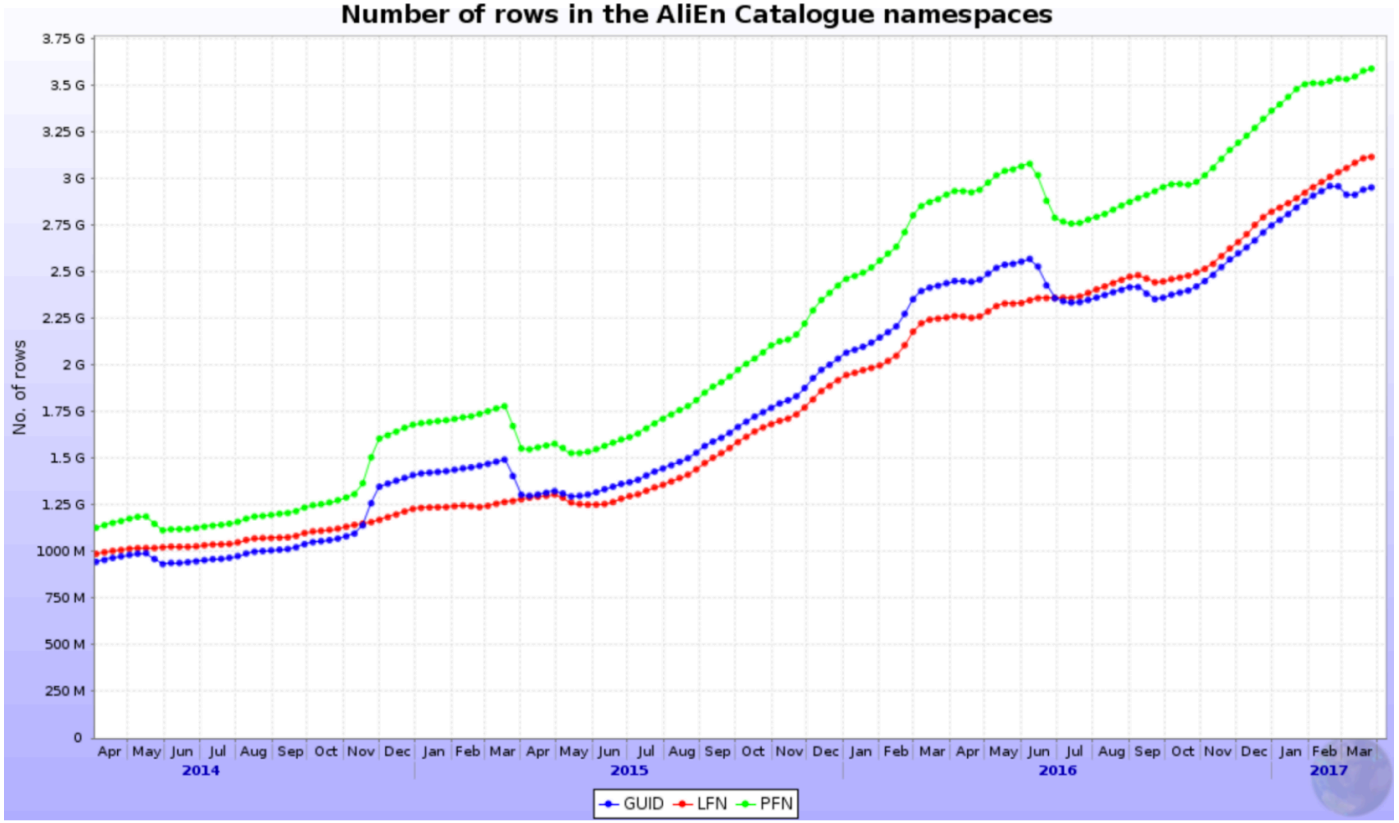
10:1 read/write data volume



Cache


Cache statistics
access
1000/s
900/s
800/s
700/s
600/s
500/s
400/s
300/s
200/s
100/s
0/s
find
100/s
75/s
50/s
25/s
0/s
whereis
1250/s
1000/s
750/s
500/s
250/s
0/s
17:00
18:00
19:00
20:00
21:00
22:00
23:00
00:00
07 Mar 2017
CEST time
HIT
SET

The AliEn catalogue in time



Future needs

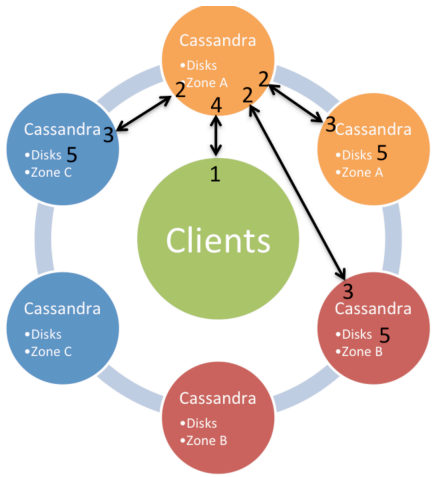
- In Run3 we will have 5x more computing resources (300K CPUs + 5000GPUs)
- 10x more disk and tape storage =>
~10x more files to manage
- The goal is to sustain ~200kHz queries (stable)
~1Mhz queries (peaks)
  - Numbers are a bit *inflated*
    - Query cache represents half or more of the select
    - Many of those queries won't be needed:
      - New backend schemas simplify (next slides)
      - Improvements on the framework
        - Preparing file envelopes for jobs at split
        - More aggressive caching in JAliEn

- Looking for a solution providing:
  - Horizontal scaling
  - No single point of failure
  - High query rate
  - HA

Apache Cassandra

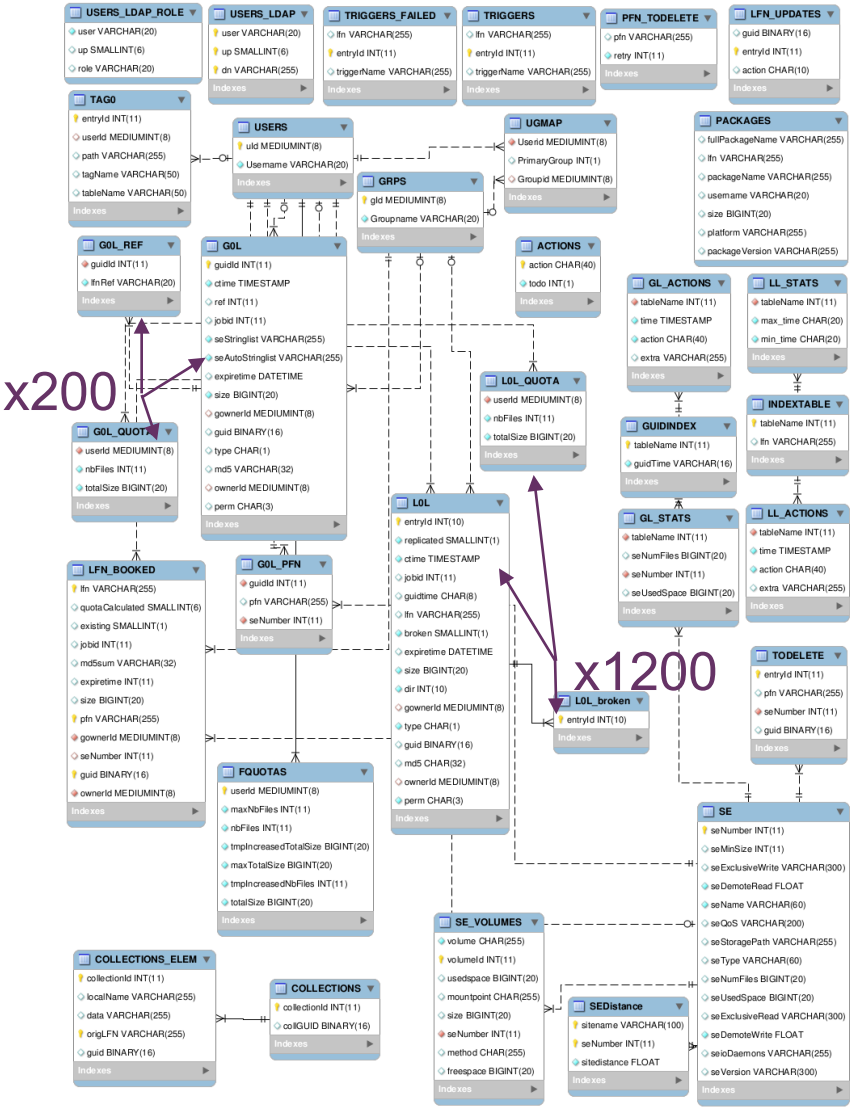
- Provides all the requirements mentioned before:
  - Horizontal scale
    - Add nodes to keep up ops/s
  - HA – No point of failure
  - Performance (see later initial benchmarks)
- Consistency
  - Tunable levels, key factor for us
- We move from N to few tables for the namespace
  - Simplification
- Easy setup
- Mapping certain SQL operations not trivial
  - Groupings, quota calculations, 'where' possibilities…
  - NoSQL re-implementation, CQL helps too

Cassandra Write Data Flows



[1] Netflix techblog

First schema in C*



Simplification!

File tree

(Path)
Child
Ctime

Owner
Group
Size
Type
Perm
EntryID
JobID
Checksum
Metadata
URLs

SE lookups

(SENumber)
EntryID

FQFileName
Size
Owner

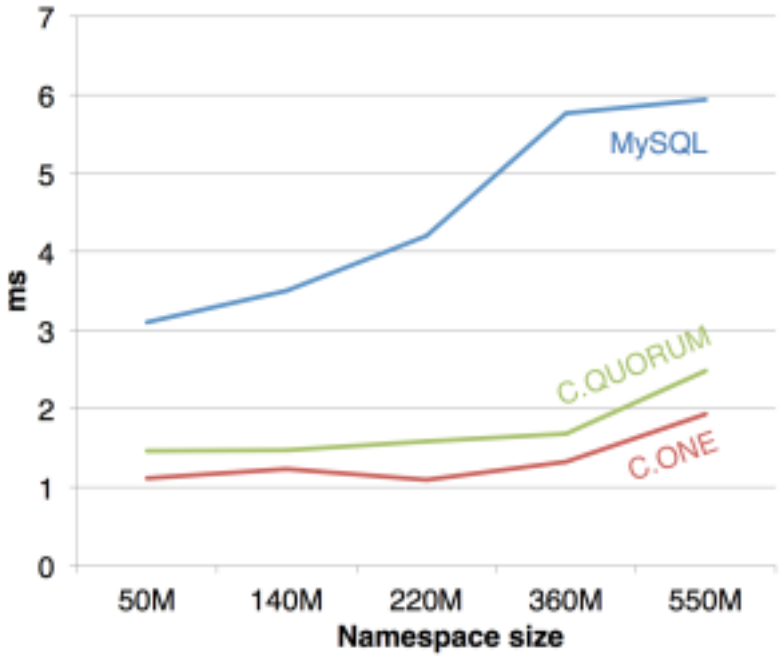
PS: some tables will stay in MySQL

Cassandra benchmarks

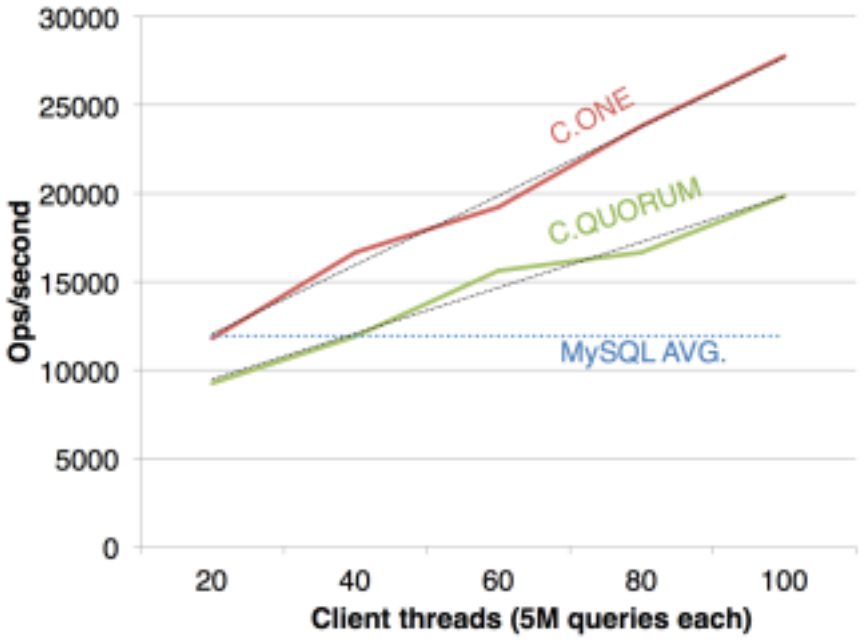
- Setup a 5-node ring
  - Server power: 16-48 cores, 100-350GB RAM
  - Java 8 Oracle, no swap, nofile/memlock limits (no degraded mode)
  - Mapped namespace into a column family that is able to do `whereis` and `ls`: entry contains lfn+pfns metadata
    - Starting a new round of benchmarking on a implementation that allows `find` as well
  - RF 3, LeveledCompaction + LZ4 compression
- Data dump
  - MySQL to Cassandra -> slow
  - Artificial lfns and dirs -> very quick!
- Execution
  - Java sized thread pool, configure hierarchies, number of LFNS, etc…

Cassandra benchmarks

- Initial benchmarking shows promising results

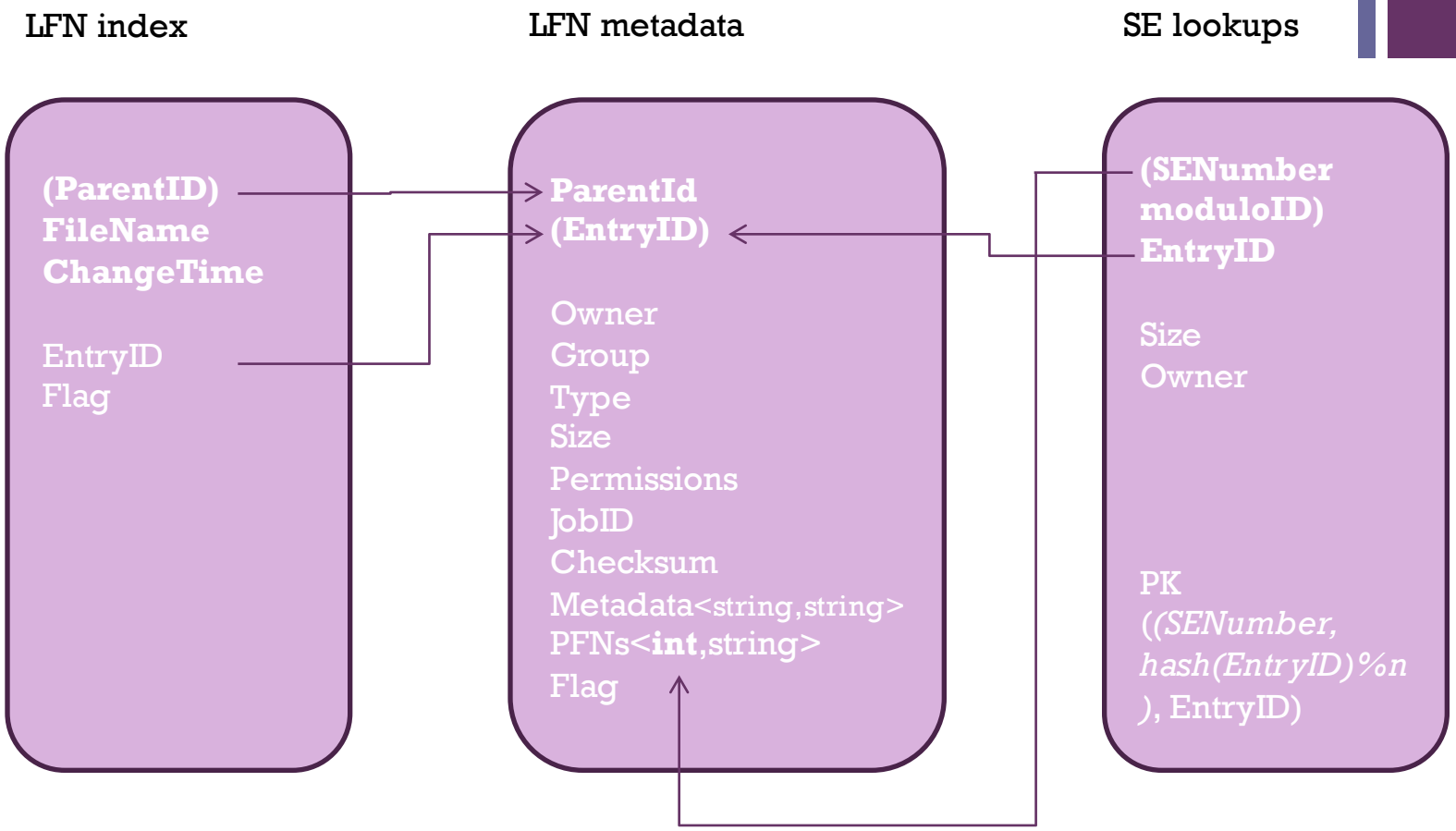


Time to retrieve logical and physical information of a file



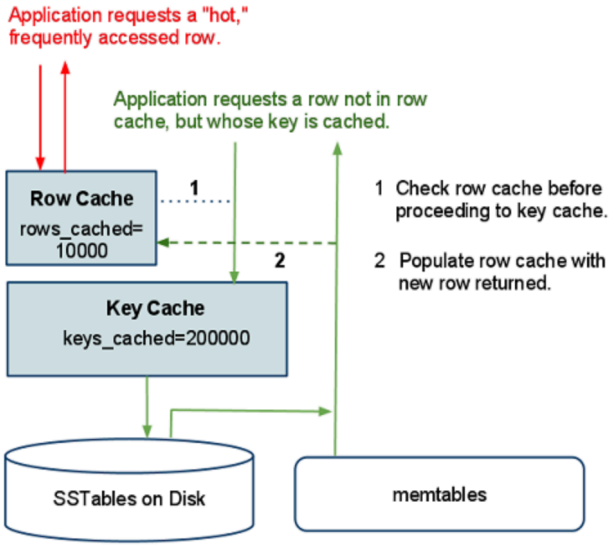
Operations per second based on number of clients

New schema in C*



New vs old schema

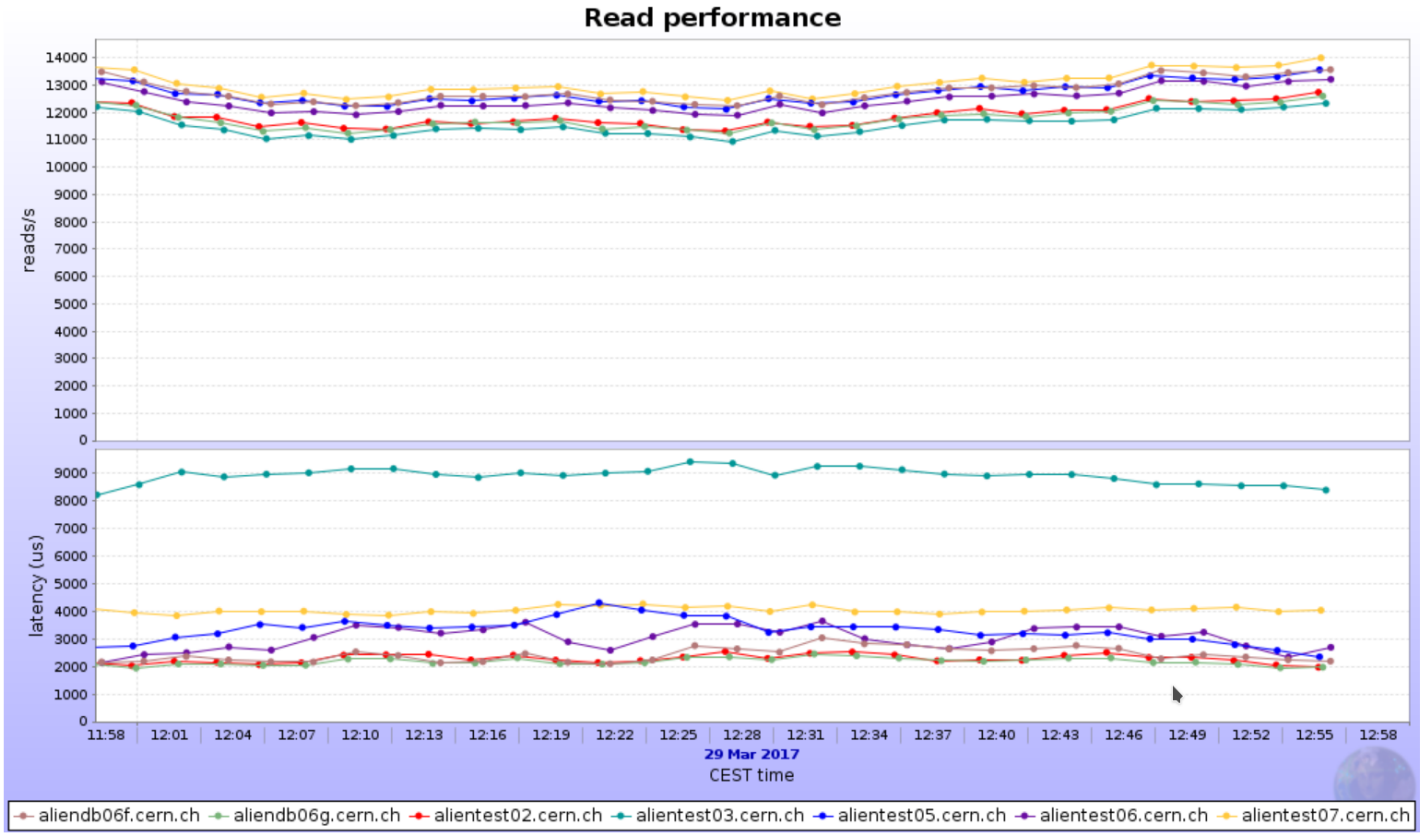
- Cons
  - Need to loop over the lfn_index (hierarchy) to do file operations
    - To avoid contention on the servers and thus latency:
      - Can be cached by client
      - (We hope) can be cache by Cassandra -> RowCache
- Pros
  - Some complex and heavy operations become much easier
    - mv dir: delete old parent and insert new
    - rm dir: mark/delete parent
  - Easy to keep a trash bin



Monitoring

- Cassandra internally calculates and exposes an ample set of metrics -> MBean (JMX)
- Naming is misleading and documentation is scarce
- Pluggable to some extended tools
- ML will have a dashboard to have a global view of the cluster and detect problems -> link to C* monitor
- Feeding most important metrics:
  - Read/Write latencies and throughput
  - KeyCache hits/requests
  - Compaction+GC stat
  - Timeouts, Unavailable, Exceptions
  - CF stats
  - Usual machine status: load, cpu/memory/network usage…

Stress test: read



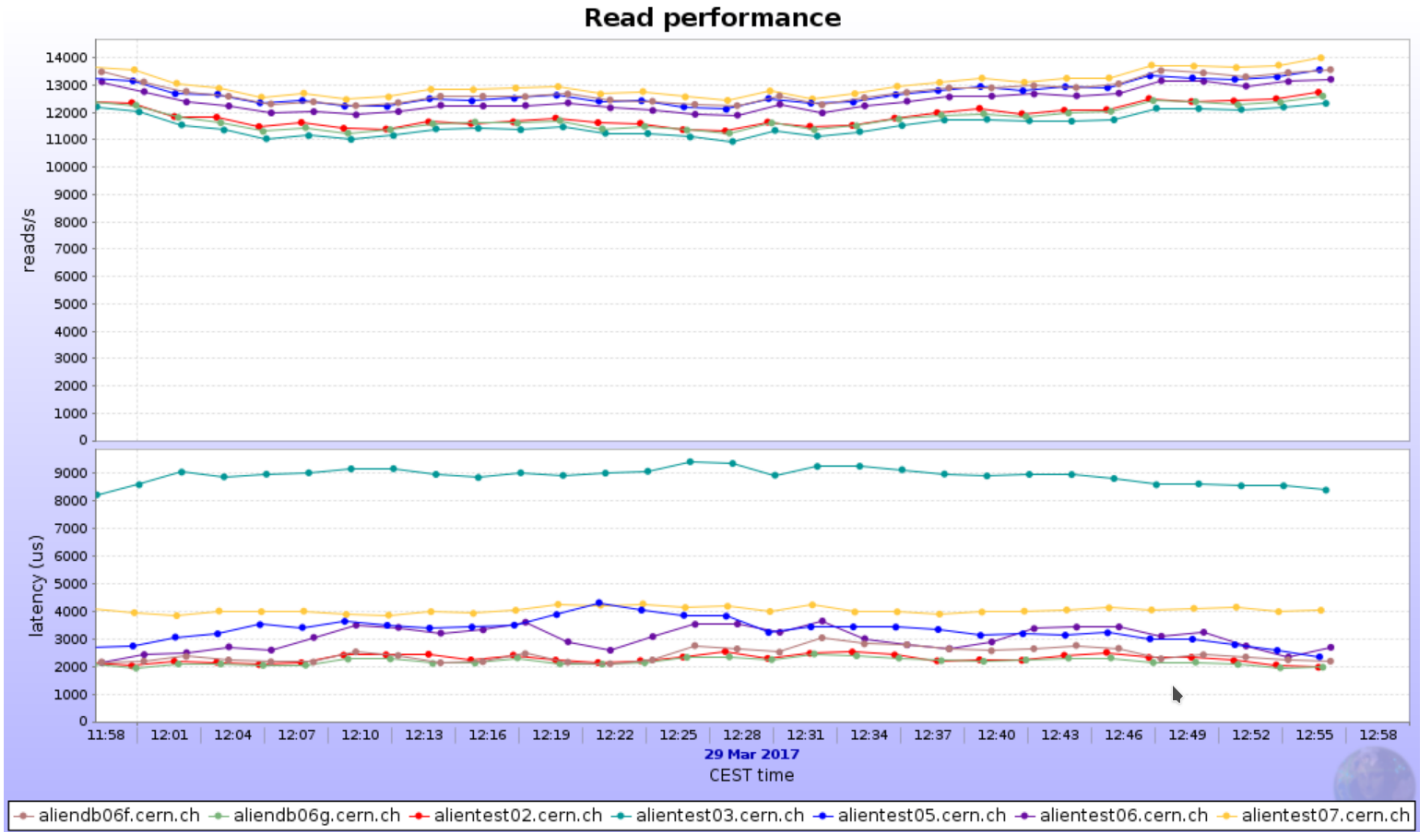
Read operations / second

	Series	Last value	Min	Avg	Max
1.	aliendb06f.cern.ch	13569	12047	12778	13701
2.	aliendb06g.cern.ch	12597	11061	11784	12727
3.	alientest02.cern.ch	12737	11216	11876	12738
4.	alientest03.cern.ch	12333	10850	11550	12415
5.	alientest05.cern.ch	13563	12002	12705	13625
6.	alientest06.cern.ch	13216	11679	12445	13299
7.	alientest07.cern.ch	13995	12338	13039	14106
	Total	92014		86179	

Average latency (us)

	Series	Last value	Min	Avg	Max
1.	aliendb06f.cern.ch	2194	1960	2426	3108
2.	aliendb06g.cern.ch	1974	1864	2176	2534
3.	alientest02.cern.ch	1969	1836	2276	2698
4.	alientest03.cern.ch	8394	7785	8923	9571
5.	alientest05.cern.ch	2349	2342	3308	4982
6.	alientest06.cern.ch	2692	1969	3023	3845
7.	alientest07.cern.ch	4040	3767	4041	4670
	Total	23614		26176	

Stress test: read



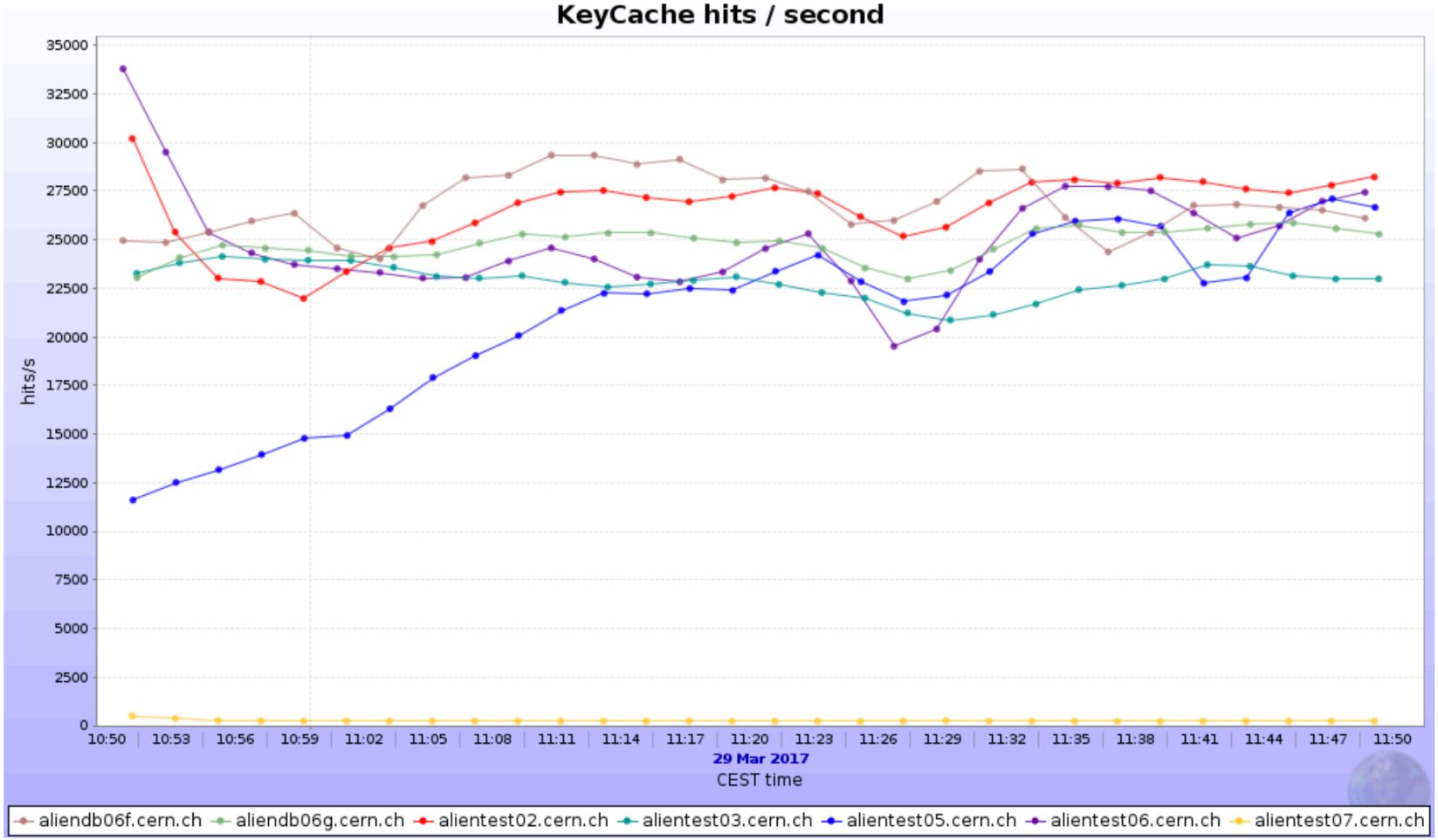
What is this about?

Machines status

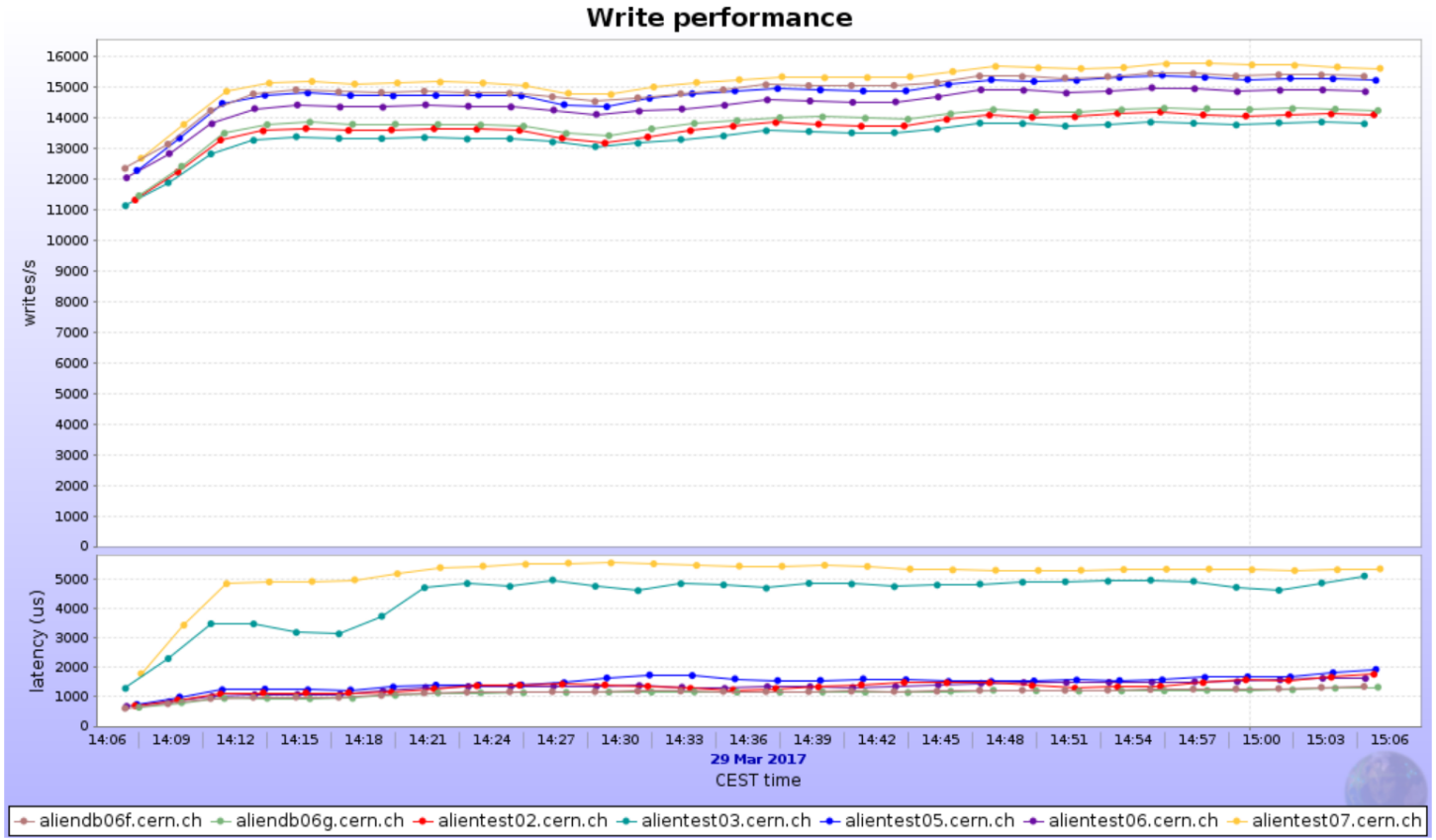
alientest,db06f,d

	Machine status					Machine type				Disk	CPU utilisation (%)								Memory utilisation				
Machine	Online	Uptime	Load	Kernel	OS	Machine model	CPU	CPUs	MHz	Space	usr	sys	iow	int	sint	steal	nice	idle	Total	Used	Buffers	Cached	Free
1. db6f		22d 20:04	18.13	4.4.0-65-...	16.04	ProLiant DL380 Gen9	Xeon E5-2687W v4 3.00GHz	48	3199		22.33	6.738	0.034	0	0.564	0	1.921	68.41	755.8 GB	178 GB	364.6 MB	571.7 GB	5.762 GB
2. db6g		19d 19:41	14.86	4.4.0-66-...	16.04	ProLiant DL380 Gen9	Xeon E5-2687W v3 3.10GHz	40	1768		25.13	7.856	0.01	0	0.902	0	2.253	63.85	755.8 GB	19.51 GB	279.8 MB	97.95 GB	638.1 GB
3. alientest02		83d 21:23	28.8	4.4.0-57-...	16.04	ProLiant DL380p Gen8	Xeon E5-2690 v2 3.00GHz	40	3000		30.79	11.07	0.002	0	1.532	0	2.477	54.13	377.9 GB	172 GB	10.56 GB	163.6 GB	31.74 GB
4. alientest03		76d 17:58	62.59	4.4.0-59-...	16.04	ProLiant DL380 G6	Xeon X5560 2.80GHz	16	2794		65.14	12.19	0.001	0	4.269	0	4.013	14.39	141.7 GB	17.98 GB	332.3 MB	94.68 GB	28.67 GB
5. alientest05		8d 1:53	37.64	4.4.0-67-...	16.04	ProLiant DL380p Gen8	Xeon E5-2697 v2 2.70GHz	48	2699		34.74	12.37	0.035	0	2.039	0	3.843	46.97	188.9 GB	57.67 GB	3.423 GB	127.1 GB	690.1 MB
6. alientest06		83d 21:24	58.49	4.4.0-57-...	16.04	ProLiant DL380p Gen8	Xeon E5-2697 v2 2.70GHz	48	2999		54.8	12.55	0.007	0	2.689	0	3.006	26.94	188.9 GB	62.19 GB	2.689 GB	99.19 GB	24.81 GB
7. alientest07		83d 21:26	61.68	4.4.0-57-...	16.04	ProLiant DL380 G6	Xeon X5560 2.80GHz	16	2794		48.34	18.14	11.69	0	6.412	0	6.288	9.125	55.03 GB	16.45 GB	209.8 MB	38.14 GB	244.1 MB
Total								**256**											**2.406 TB**	**523.7 GB**	**17.83 GB**	**1.164 TB**	**730 GB**
Average		**54d 3:59**	**40.31**								**40.18**	**11.56**	**1.683**	**0**	**2.629**	**0**	**3.4**	**40.55**	**352 GB**	**74.82 GB**	**2.548 GB**	**170.3 GB**	

Stress test: read



Stress test: write



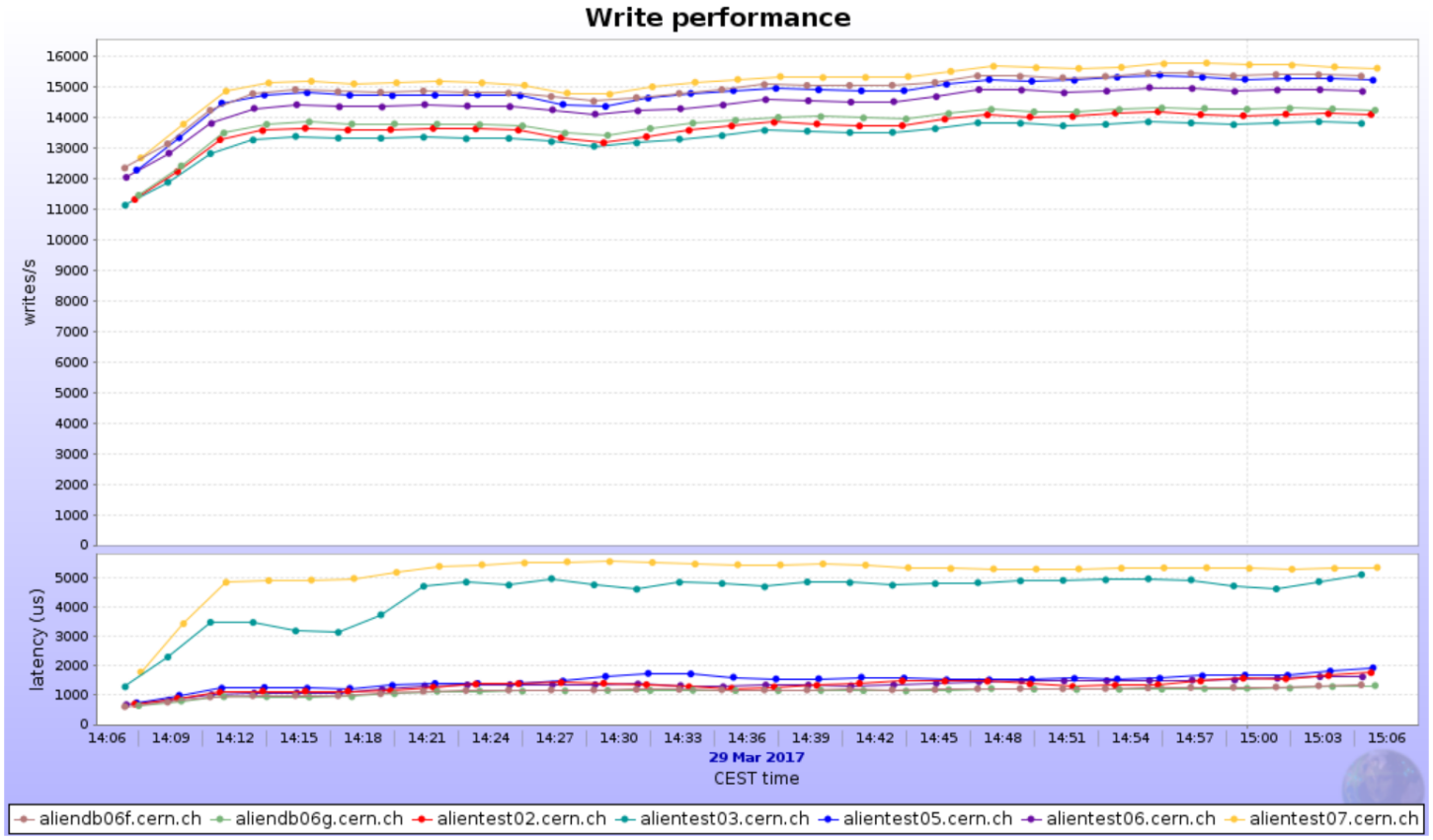
Write operations / second

	Series	Last value	Min	Avg	Max
1.	aliendb06f.cern.ch	15361	12150	14885	15571
2.	aliendb06g.cern.ch	14243	11282	13834	14425
3.	alientest02.cern.ch	14106	11193	13651	14336
4.	alientest03.cern.ch	13814	10900	13377	14029
5.	alientest05.cern.ch	15230	12097	14801	15471
6.	alientest06.cern.ch	14858	11751	14417	15103
7.	alientest07.cern.ch	15625	12342	15203	15923
	Total	103241		100171	

Average latency (us)

	Series	Last value	Min	Avg	Max
1.	aliendb06f.cern.ch	1348	570.4	1140	1380
2.	aliendb06g.cern.ch	1321	617.2	1117	1338
3.	alientest02.cern.ch	1763	642.8	1325	1824
4.	alientest03.cern.ch	5093	1135	4378	5625
5.	alientest05.cern.ch	1927	678.7	1489	2030
6.	alientest06.cern.ch	1619	640.8	1321	1703
7.	alientest07.cern.ch	5340	1391	5126	5748
	Total	18414		15900	

Stress test: write



What is this about?

Machines status

alientest,db06

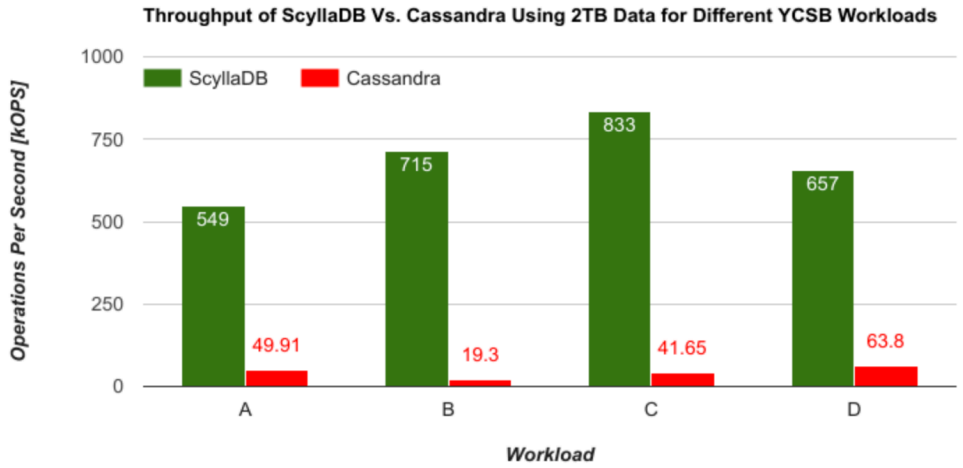
	Machine status			Machine type						Disk	CPU utilisation (%)								Memory utilisation				
Machine	Online	Uptime	Load	Kernel	OS	Machine model	CPU	CPUs	MHz	Space	usr	sys	iow	int	sint	steal	nice	idle	Total	Used	Buffers	Cached	Free
1. db6f		22d 23:20	16.25	4.4.0-65-...	16.04	ProLiant DL380 Gen9	Xeon E5-2687W v4 3.00GHz	48	3199		23.4	7.453	0.069	0	0.74	0	1.803	66.53	755.8 GB	177.7 GB	362.8 MB	544.3 GB	33.51 GB
2. db6g		19d 22:57	16.71	4.4.0-66-...	16.04	ProLiant DL380 Gen9	Xeon E5-2687W v3 3.10GHz	40	2786		26.68	8.928	0.032	0	1.134	0	2.377	60.85	755.8 GB	19.44 GB	282.8 MB	106.8 GB	629.3 GB
3. alientest02		84d 0:40	34.58	4.4.0-57-...	16.04	ProLiant DL380p Gen8	Xeon E5-2690 v2 3.00GHz	40	3000		37.16	19.46	0.051	0	3.543	0	3.853	35.93	377.9 GB	173.3 GB	10.6 GB	173.4 GB	20.55 GB
4. alientest03		76d 21:14	56.66	4.4.0-59-...	16.04	ProLiant DL380 G6	Xeon X5560 2.80GHz	16	2794		49.29	9.142	0.243	0	3.634	0	10.91	26.78	141.7 GB	18.02 GB	335.8 MB	103.8 GB	19.55 GB
5. alientest05		8d 5:10	35.96	4.4.0-67-...	16.04	ProLiant DL380p Gen8	Xeon E5-2697 v2 2.70GHz	48	2700		35.59	17.42	0.079	0	3.687	0	3.879	39.34	188.9 GB	48.75 GB	3.184 GB	118.1 GB	18.79 GB
6. alientest06		84d 0:41	32.1	4.4.0-57-...	16.04	ProLiant DL380p Gen8	Xeon E5-2697 v2 2.70GHz	48	2999		27.68	10.41	0.045	0	1.969	0	3.362	56.53	188.9 GB	51.35 GB	2.812 GB	116.5 GB	18.18 GB
7. alientest07		84d 0:43	53.67	4.4.0-57-...	16.04	ProLiant DL380 G6	Xeon X5560 2.80GHz	16	2794		53.48	12.13	0.873	0	5.487	0	6.401	21.63	55.03 GB	16.77 GB	244.1 MB	34.61 GB	3.411 GB
Total								**256**											**2.406 TB**	**505.3 GB**	**17.79 GB**	**1.169 TB**	**743.3 GB**
Average		**54d 7:15**	**35.13**								**36.18**	**12.14**	**0.199**	**0**	**2.885**	**0**	**4.655**	**43.94**	**352 GB**	**72.19 GB**	**2.541 GB**	**171.1 GB**	

Next steps

- Tune Cassandra
  - Investigate and optimize CPU usage
  - Tune JVM+GC, RowCache/KeyCache sizes…
- Run benchmarks with `cassandra-stress`
  - Comparable to similar clusters of the community?
- Get closer to a production workload
  - Mixed set of select/update/insert/delete as in the current catalogue
- Exercise critical operations
  - Backups
  - Addition/replacement of nodes

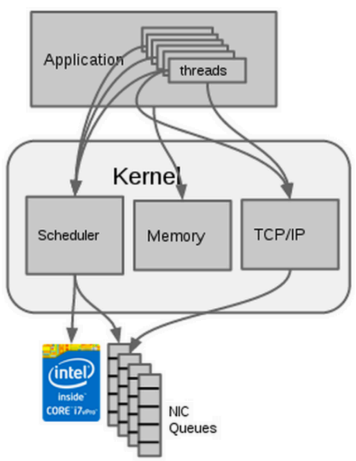
ScyllaDB

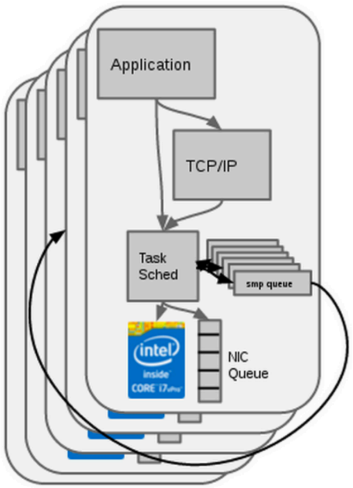
- "Next generation Cassandra"



[Full report here]

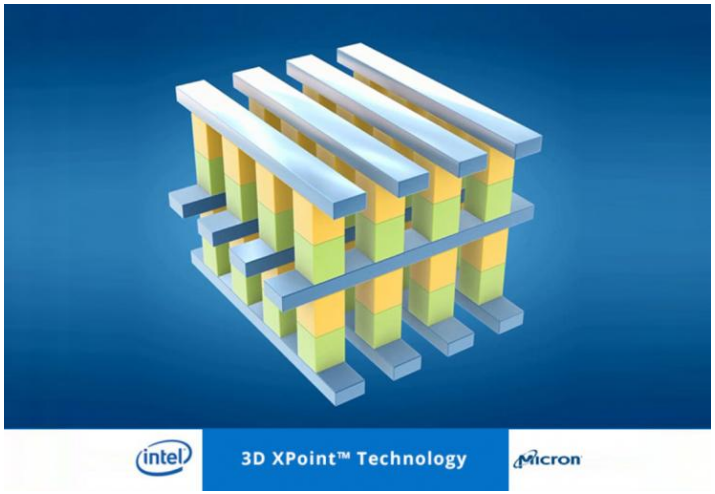
- Difference relies on core implementation





3D Crosspoint

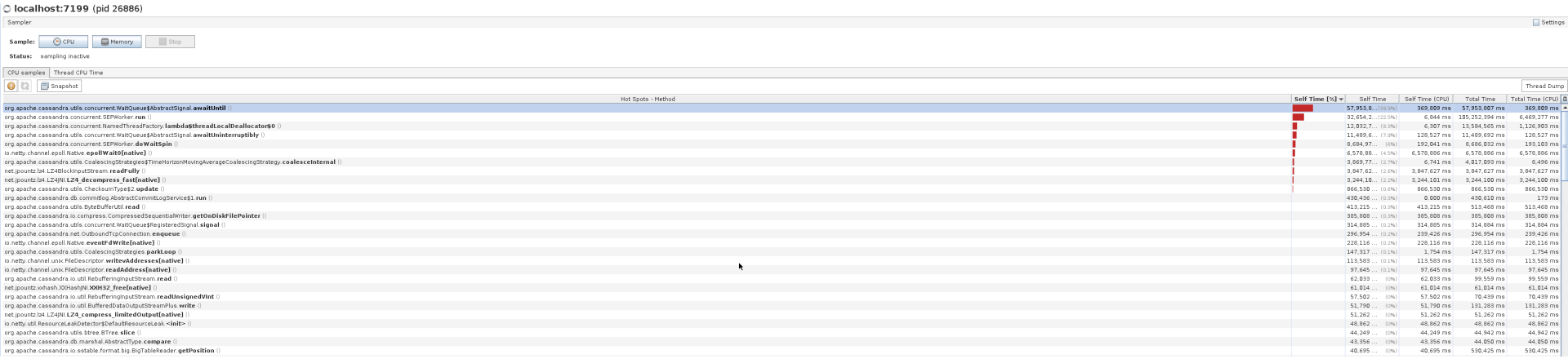
- Biggest memory breakthrough in 25 years
- Not very clear yet how it will work, but provides:
  - Higher data volume than RAM
  - Low latency
  - ½ price RAM?
- Persistent RAM
  - Remove slow I/O layers -> In-memory DB?
  - Booking area…



Thanks

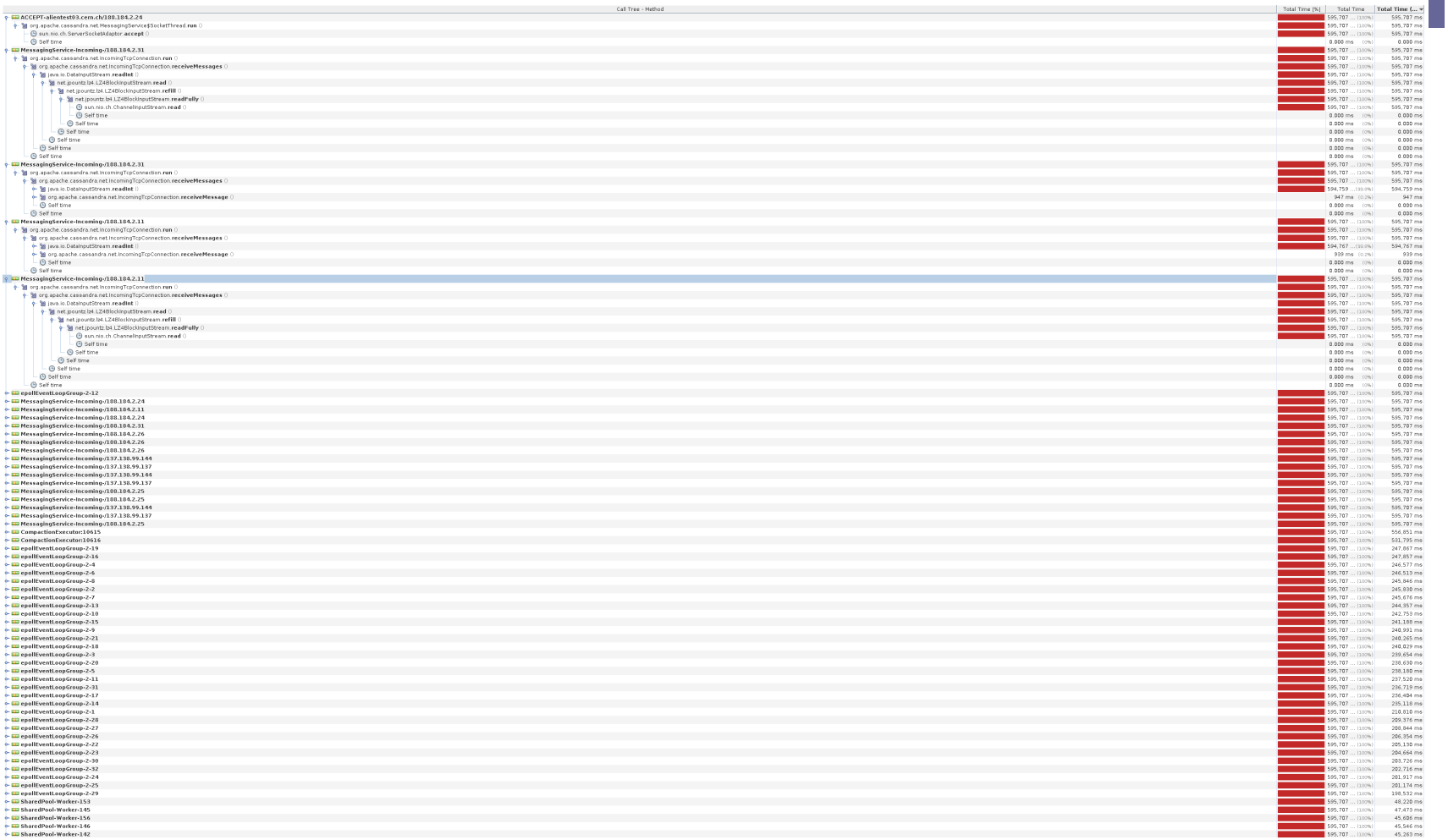
- Questions?

B slides - CPU debug: cpu sample



B slides - CPU debug: GC

B slides - CPU debug: methods/cpu usage



B slides - CPU debug: jstack info

```
"(ML ThP) [ util.process ] Worker 250448, started: Wed Mar 29 16:57:08 CEST 2017" #1096365 daemon prio=5 os_prio=0 tid=0x00007f58680c6800 nid=0x1547 waiting on condition
[0x00007f57632f1000]
   java.lang.Thread.State: TIMED_WAITING (parking)
        at sun.misc.Unsafe.park(Native Method)
        - parking to wait for  <0x00000006c0002870> (a java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject)
        at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215)
        at java.util.concurrent.locks.AbstractQueuedSynchronizer$ConditionObject.awaitNanos(AbstractQueuedSynchronizer.java:2078)
        at java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.poll(ScheduledThreadPoolExecutor.java:1129)
        at java.util.concurrent.ScheduledThreadPoolExecutor$DelayedWorkQueue.poll(ScheduledThreadPoolExecutor.java:809)
        at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1066)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1127)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
        at java.lang.Thread.run(Thread.java:745)
```