Introduction to Cassandra.yaml

& friends

THE LAST PICKLE

Hi, I'm Edward Capriolo.

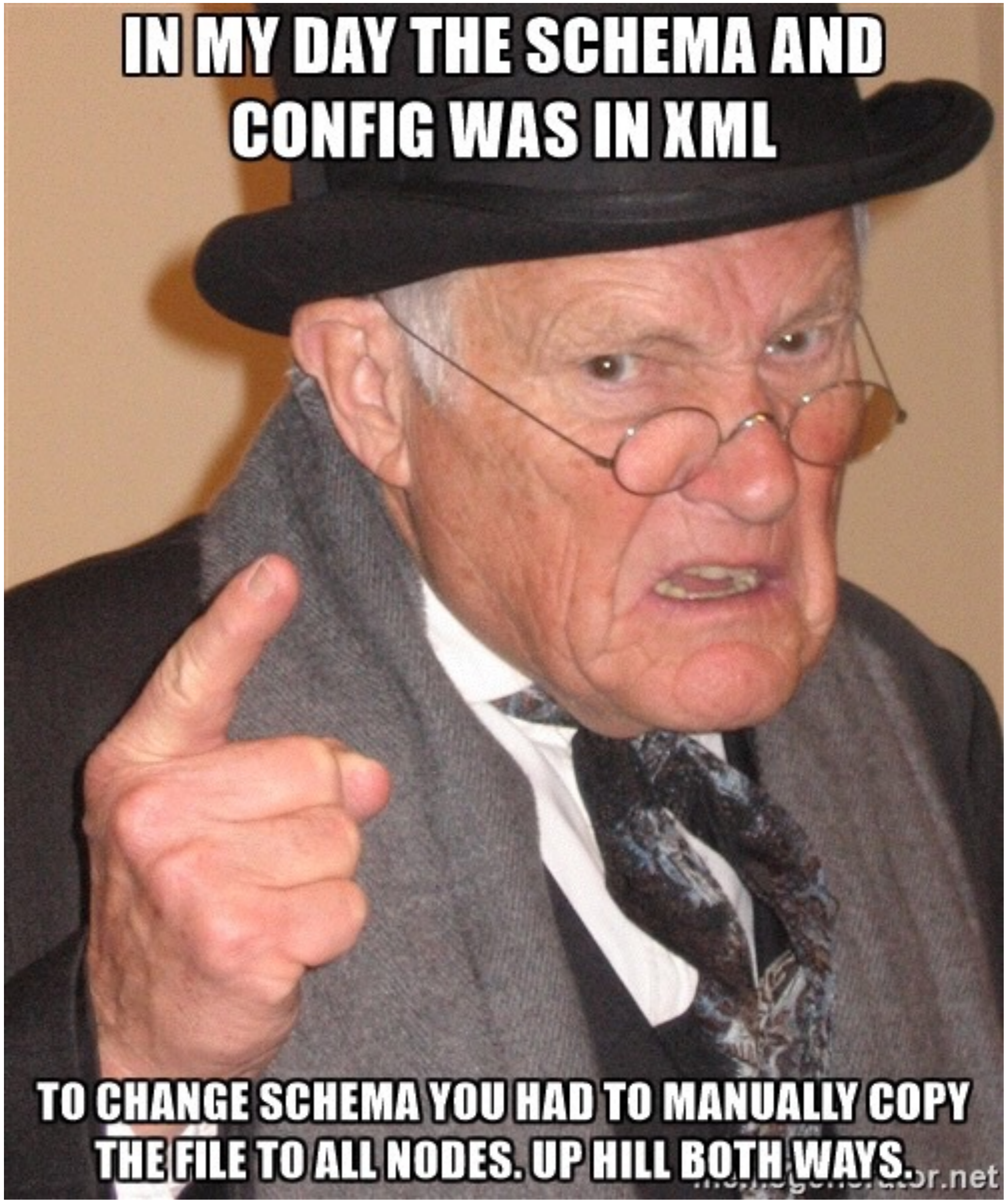
@edwardcapriolo
https://www.linkedin.com/in/edwardcapriolo
http://www.slideshare.net/edwardcapriolo

Consultant
The Last Pickle

Cassandra user since (v 0.6.5)
White Plains, NY USA

improve Apache based solutions.

THE LAST PICKLE


IN MY DAY THE SCHEMA AND CONFIG WAS IN XML
TO CHANGE SCHEMA YOU HAD TO MANUALLY COPY THE FILE TO ALL NODES. UP HILL BOTH WAYS.
memegenerator.net

This talk is the ‘gateway’ talk…

Many ‘picklers’ (TLP staff) are covering some points I will quickly cover over in depth in other talks.

Section Overview

1. Key configuration settings
2. Configuration outside of the yaml
3. Multi-system configuration settings
4. Advanced settings
5. Exotic settings

Basic setup

1. $ wget <apache-cassandra*.tar.gz>
2. $ tar -xf <apache-cassandra*.tar.gz>
3. $ apache-cassandra*/bin/cassandra

Result:

Web scale distributed storage

Drop Mic.

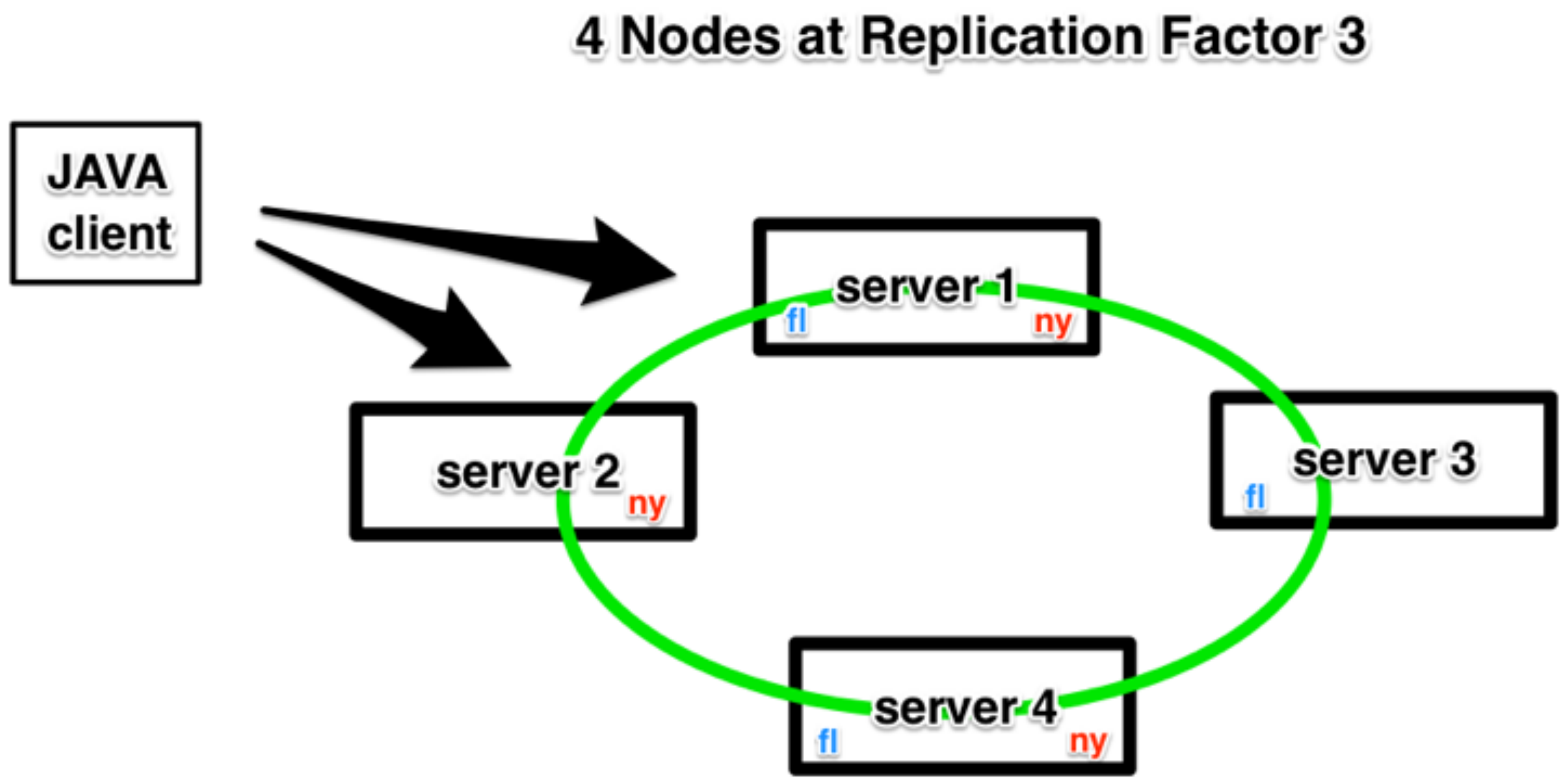
Well almost…

We have to do a bit of configuration.

Before we dive into config

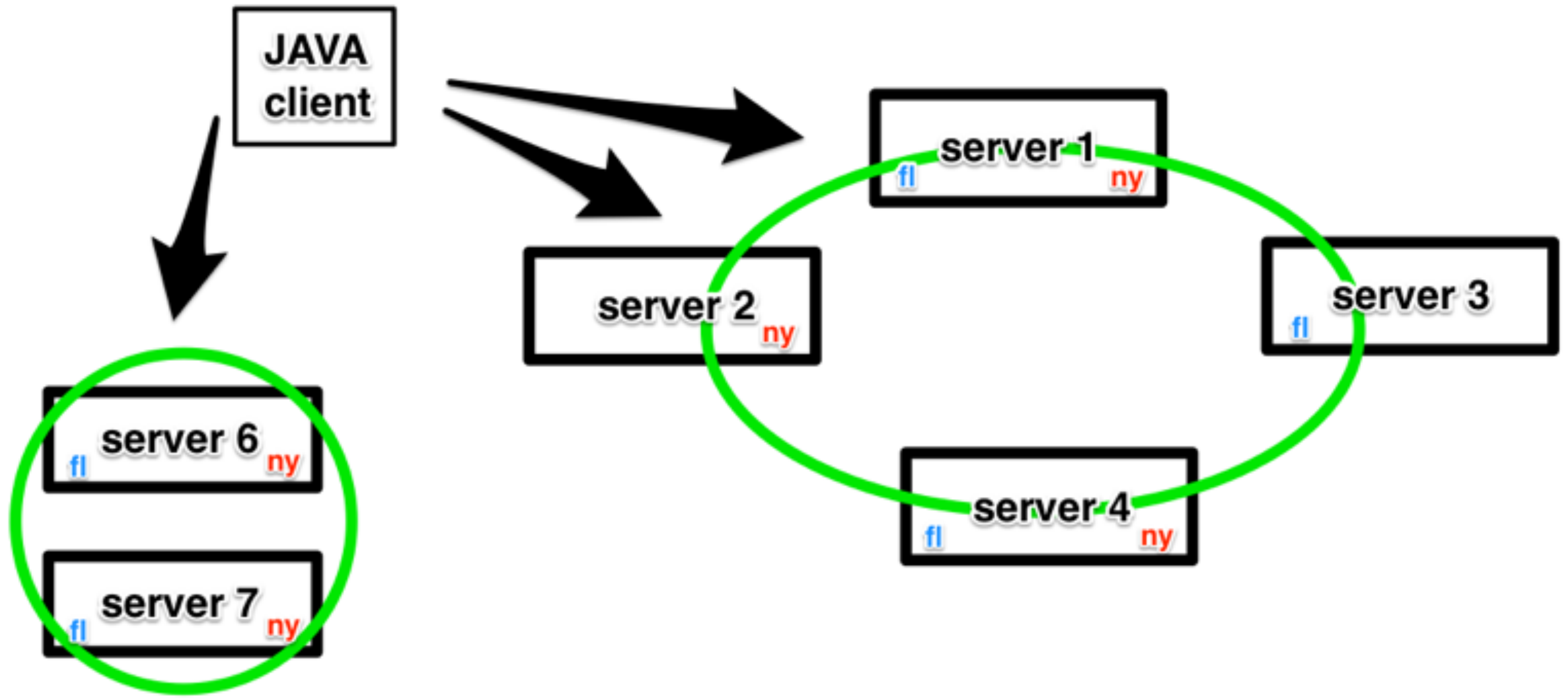
```
cqlsh> CREATE KEYSPACE test WITH replication =
{'class': 'SimpleStrategy', 'replication_factor' : 1};
cqlsh> USE test;
cqlsh:test> CREATE COLUMNFAMILY trip (src varchar,
...  dest varchar, PRIMARY KEY (src,dest));

cqlsh:test> INSERT INTO trip (src, dest) VALUES ('ny', 'ca');

cqlsh:test> SELECT * FROM trip;

 src | dest
-----+------
  ny |   ca


cqlsh:test> INSERT INTO trip (src, dest) VALUES ('fl', 'ca');
cqlsh:test> SELECT * FROM trip;

 src | dest
-----+------
  fl |   ca
  ny |   ca
```

Single Data Center



Multiple Data Center

DC1: 4 Nodes at Replication Factor 3
DC2: 2 Nodes at Replication Factor 2



Where does the data go?

```
data_file_directories:
    - /var/lib/cassandra/data
```

1. User data is stored in all listed directories
2. Do: fast seek’ing storage (SSD)
3. Do: ample free space (30% overhead)
4. Don’t: Store on a SAN

Commit log storage

commitlog_directory:
  - /var/lib/cassandra/commitlog

1. Stores unflushed mutations (write/deletes)
2. Don't: Assume these are log4j type logs
3. Do: use a dedicated disk if possible
4. Do: provide at least 10GB (write velocity)

Ok we now where (most of) the data goes…

How do clients connect?

Default port binding

```
$ netstat -nl
tcp        0      0 127.0.0.1:7000          0.0.0.0:*               LISTEN
tcp6       0      0 127.0.0.1:9042          :::*                    LISTEN
```

1. Cassandra does not bind to 0.0.0.0
2. 127.0.0.1 not web scale
3. 7000 is the “Storage Port” inter node traffic
4. 9042 is the “Native Port” client traffic

Native transport

```
start_native_transport: true (default)
native_transport_port: 9042 (default)
listen_address: localhost
```

1. Change listen_address to a client-reachable address
2. Do: consider transport security
3. Do: consider network routing performance
4. Don’t: put nodes on a public network. EVAR

Outside the yaml file…

cassandra-env.sh (& friends)

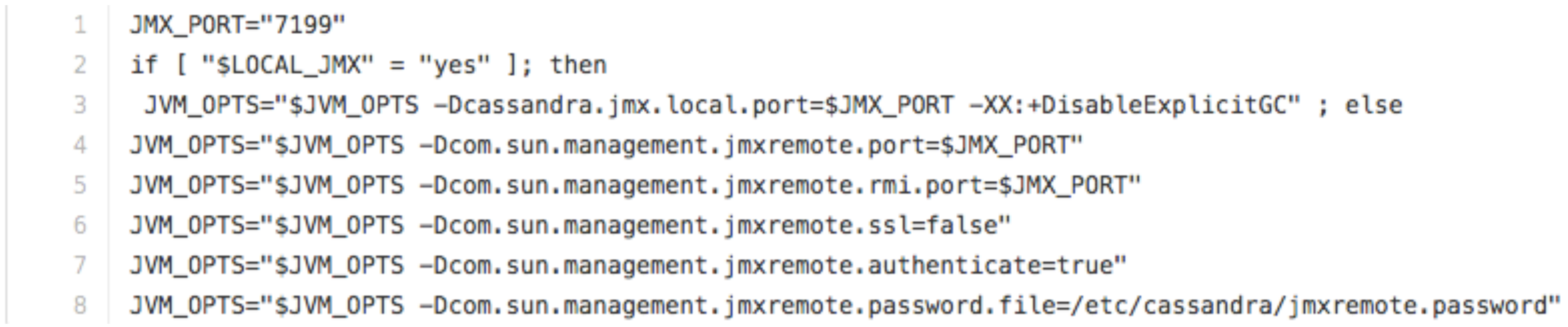
1. JVM and startup params defined outside the YAML
2. Newer version of c* use jvm.options

Memory usage

```
#MAX_HEAP_SIZE="1G"
#HEAP_NEWSIZE="100M"
```

1. max(min(1/2 ram, 1024MB),
   min(1/4 ram, 8GB))
2. Do: set lower when experimenting with workstation
3. Do: leave ample free memory for disk cache

JMX



```
JMX_PORT="7199"
if [ "$LOCAL_JMX" = "yes" ]; then
 JVM_OPTS="$JVM_OPTS -Dcassandra.jmx.local.port=$JMX_PORT -XX:+DisableExplicitGC" ; else
JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.port=$JMX_PORT"
JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.rmi.port=$JMX_PORT"
JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.ssl=false"
JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.authenticate=true"
JVM_OPTS="$JVM_OPTS -Dcom.sun.management.jmxremote.password.file=/etc/cassandra/jmxremote.password"
```

1. bin/nodetool uses JMX to administer Cassandra
2. All management tools require password if set

Check out Nate's talk on Securing Cassandra to learn more

Multi-node configurations

Phi convict threshold

```
# phi_convict_threshold: 8
```

1. Threshold for failure detector
2. False positives make nodes appear down to peers
3. Do: Raise for flaky WAN networks 10 - 12

Defining network topology

```
# endpoint_snitch: SimpleSnitch
```

1. Snitch with config data determines topology
2. Do: use SimpleSnitch for single switch/LAN
3. Consider: Multi DC to start

Gossiping Property File Snitch

```
conf/cassandra-rackdc.properties
dc=dc1
rack=rack1
```

1. Information is propagated around the cluster
2. DC may not be physical but is a replication unit
3. Rack has impact on replication copies
4. Don’t: Change rack unless you understand the impact

Internode communications

```
internode_compression: all | dc | none
inter_dc_tcp_nodelay: false
```

1. WAN can benefit from reduced size

```
server_encryption_options:
    internode_encryption: none
internode_authenticator:
o.a.c.auth.AllowAllInternodeAuthenticator
```

2. Settings which server nodes use to communicate

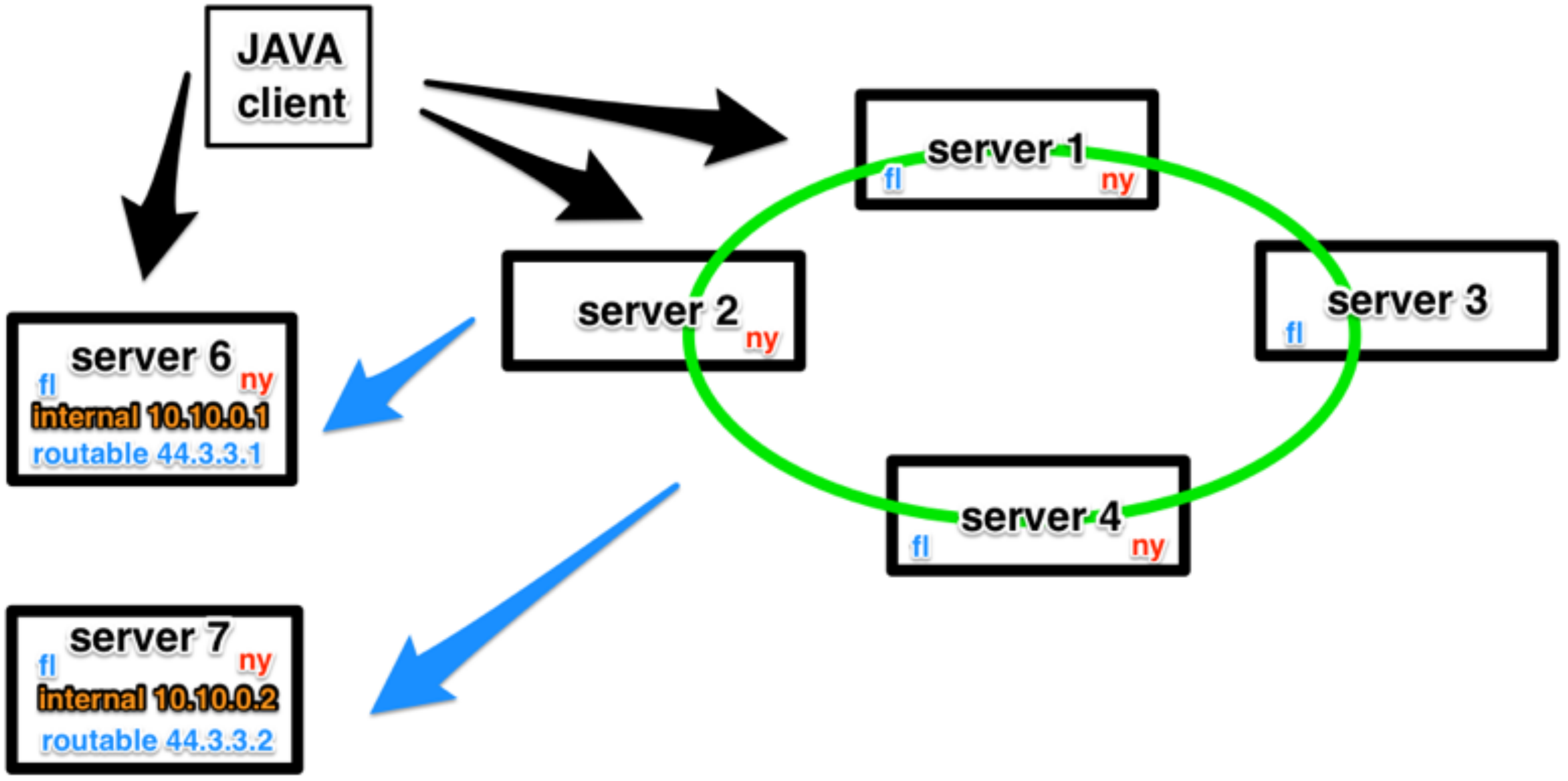
Broadcast address

```
broadcast_address: 1.2.3.4
listen_on_broadcast_address: false
broadcast_rpc_address: 1.2.3.4
```

1. Gossip a specific address (not bind address)
2. Useful in NAT and cloud environments

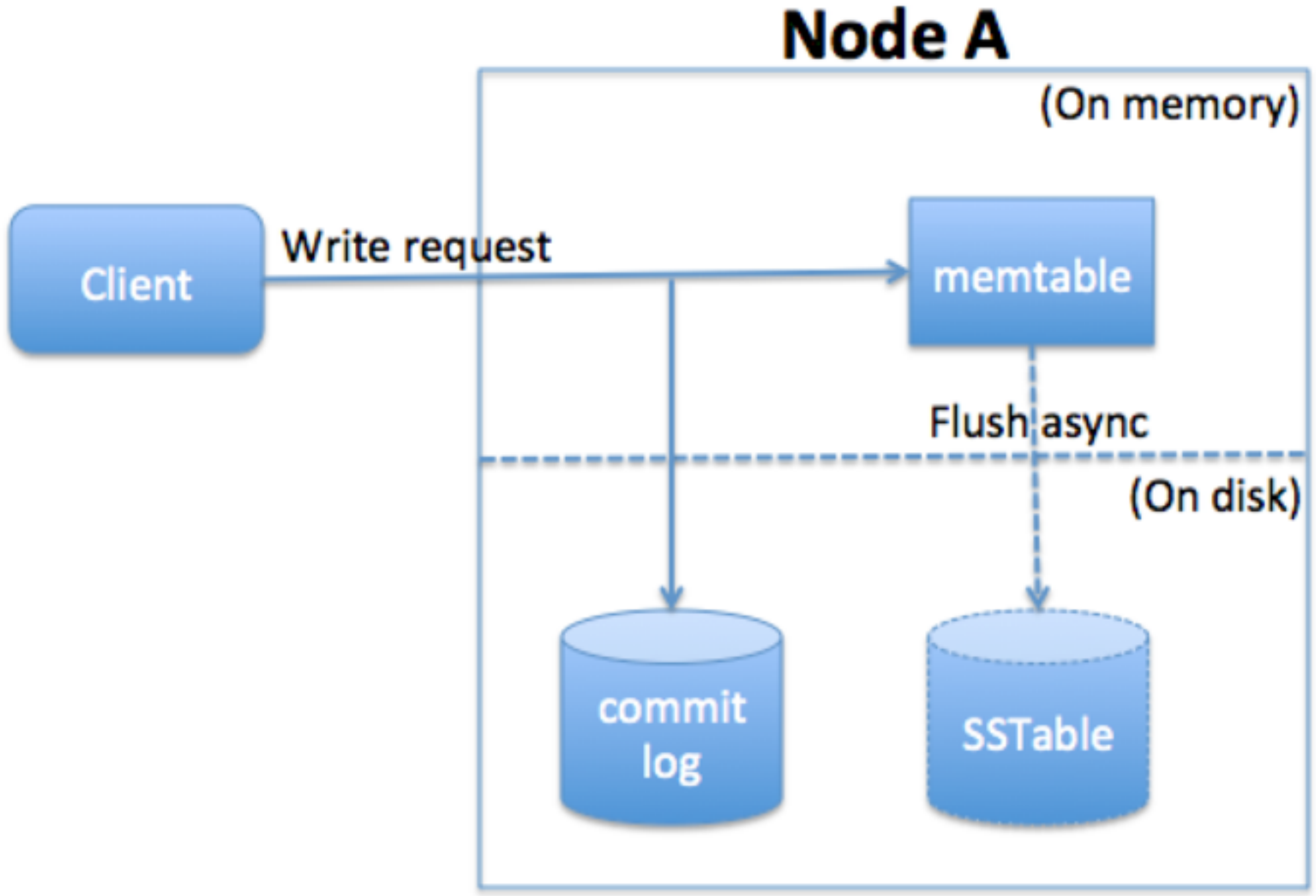
Broadcast address

DC1: 4 Nodes at Replication Factor 3
DC2: 2 Nodes at Replication Factor 2



Advanced settings

Write path



http://www.toadworld.com/platforms/nosql/w/wiki/11621.an-introduction-to-apache-cassandra

Memtables

```
#memtable_flush_writers: 1
```

1. Default One per data directory

```
# memtable_cleanup_threshold
defaults to 1 /
(memtable_flush_writers +
#memtable_cleanup_threshold: 0.11
```

2. 1 / (1 + 1) = .5

.5 of what you ask?

```
#If omitted, both set to 1/4 the heap
#memtable_heap_space_in_mb: 2048
#memtable_offheap_space_in_mb: 2048
```

1. Depending on the next setting dictates how much of each memory type is used

```
#heap_buffers: on heap nio buffers
#offheap_buffers: off heap nio buffers
#offheap_objects: off heap objects
#memtable_allocation_type: heap_buffers
```

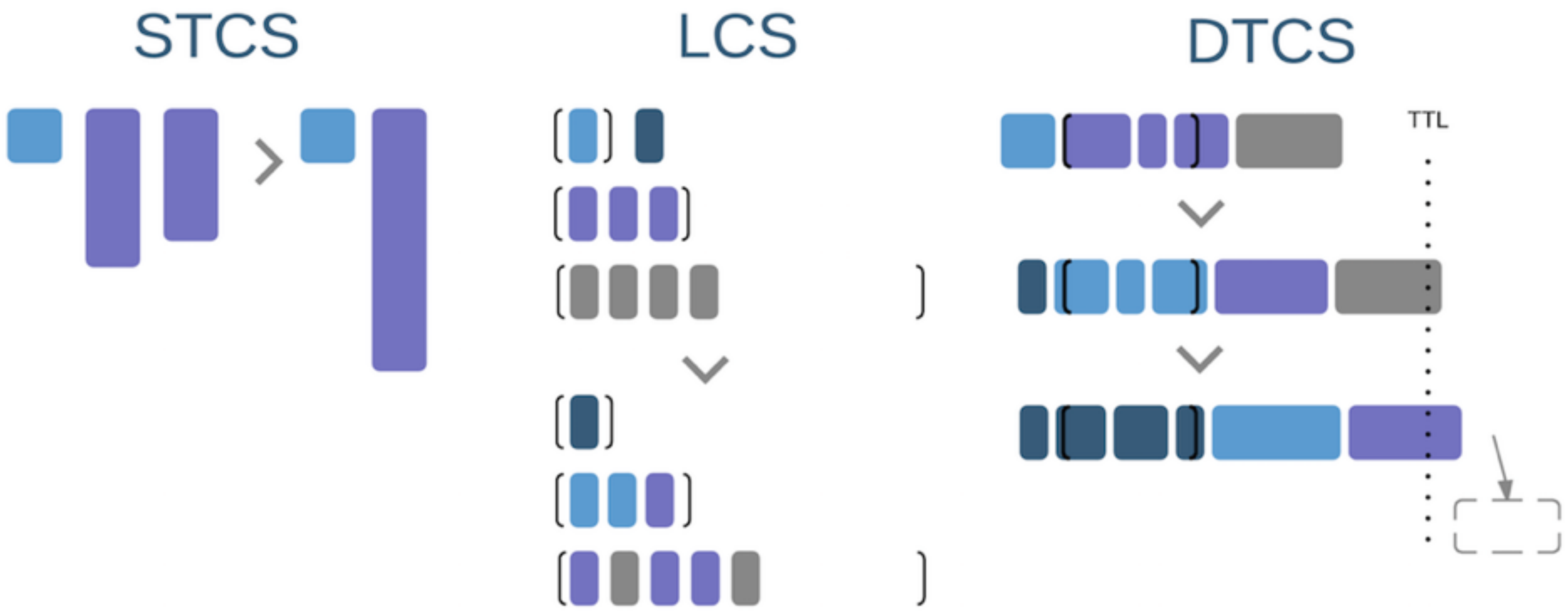
2. Based on column value buffers vs objects may be better

Trickle fsync

```
trickle_fsync: false
trickle_fsync_interval_in_kb: 10240
```

1. Optimization to periodically f-sync large files
2. Designed to prevent latency spikes in read path

Compaction



https://www.instaclustr.com/blog/2016/01/27/apache-cassandra-compaction/

Compaction

```
concurrent_compactors: 1
compaction_throughput_mb_per_sec: 16
```

1. Control resources used by compaction
2. Compaction throughput can be changed at runtime
3. Generally concurrent_compactors < 8 and > 1

Disk Failure settings

```
disk_failure_policy: stop
commit_failure_policy: stop
```

1. stop_paranoid: shut down gossip and client transports even for single-sstable errors, kill the JVM for errors during startup
2. die: shut down gossip and Thrift and kill the JVM, so the node can be replaced

Hints

```
hinted_handoff_enabled: true
max_hint_window_in_ms: 10800000
hinted_handoff_throttle_in_kb: 1024
max_hints_delivery_threads: 2
hints_directory: /var/lib/cassandra/hints
hints_flush_period_in_ms: 10000
max_hints_file_size_in_mb: 128
hints_compression: LZ4Compressor
```

1. Hints recently redesigned, again again
2. Don’t: tune high and overwhelming recovering node
3. Don’t: tune low and have out of sync data

Disk optimization strategy

```
#disk_optimization_strategy: ssd
```

1. Tip for those with rota

Exotic settings

Auto bootstrap

```
auto_bootstrap : true(hidden variable)
```

1. “Bootstrapping” here means: Should the node joining attempt to acquire data from other nodes or startup empty
2. Can be used when bringing on new datacenter
3. Can be used when streaming/ join issues

Backup*Ish options

```
incremental_backups: false
snapshot_before_compaction: false
auto_snapshot: true
```

1. Enable with external backup like tools
2. Creates hard link files operator must clean up
3. Enabling and not cleaning will cause disk fill up
4. Truncate/drop makes snapshot

Per operation default timeouts

```
read_request_timeout_in_ms: 5000
write_request_timeout_in_ms: 2000
request_timeout_in_ms: 10000
```

1. Each operation type has different timeout
2. Applied on the coordinator not the client
3. Previously was only global rpc_timeout

Commit Log sync

```
commitlog_sync: periodic
commitlog_sync_period_in_ms: 10000
commitlog_segment_size_in_mb: 32
```

1. Alternative batch mode blocks ack to clients
2. Commit logs persist until Memtable’s flush

Thanks!

@edwardcapriolo

THE LAST PICKLE