



# Cassandra in Action

ApacheCon NA 2013

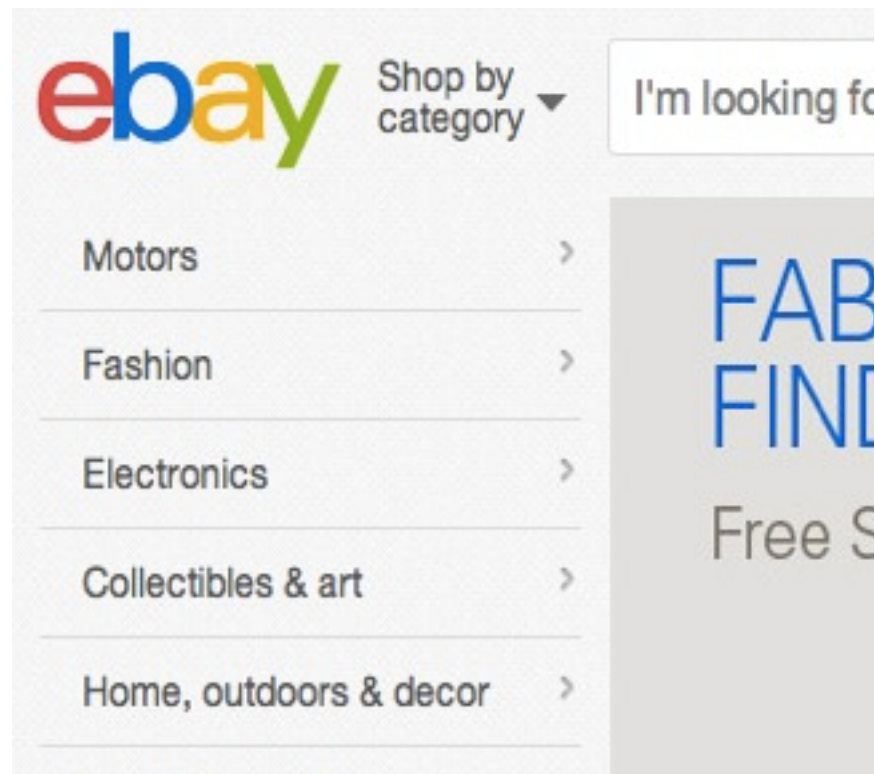
Yuki Morishita

Software Developer@DataStax / Apache Cassandra Committer





# eBay



## Application/Use Case

- Social Signals: like/want/own features for eBay product and item pages
- Hunch taste graph for eBay users and items
- Many time series use cases

## Why Cassandra?

- Multi-datacenter
- Scalable
- Write performance
- Distributed counters
- Hadoop support

# Time series data

**DataStax OPSCENTER**

Performance Metrics » Unsaved Preset

Cluster Write Operations (Edit Graph)

Cluster Read Operations (Edit Graph)

```
85.229.84.246 - - [07/Aug/2012:23:57:26 -0400] "GET /parties/bits.tcl?party_id=52375 HTTP/1.1" 304 0 0.002119 "user_id=272983" - "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20100101 Firefox/15.0"
67.188.234.120 - - [07/Aug/2012:23:57:26 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398245360 HTTP/1.1" 200 105 0.002393 "user_id=222276" - "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:14.0) Gecko/20100101 Firefox/14.0.1"
68.98.55.30 - - [07/Aug/2012:23:57:26 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398250124 HTTP/1.1" 200 106 0.002316 "user_id=272355" - "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.11 (KHTML, like Gecko) Chrome/20.0.1132.57 Safari/536.11"
216.8.163.19 - - [07/Aug/2012:23:57:26 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398247529 HTTP/1.1" 200 104 0.002349 "user_id=223" - "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.1 (KHTML, like Gecko) Chrome/21.0.1180.60 Safari/537.1"
70.140.188.229 - - [07/Aug/2012:23:57:27 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398232675 HTTP/1.1" 200 104 0.002660 "user_id=6027" - "Mozilla/5.0 (Windows NT 6.0; rv:14.0) Gecko/20100101 Firefox/14.0.1"
85.229.84.246 - - [07/Aug/2012:23:57:27 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398247167 HTTP/1.1" 200 104 0.002368 "user_id=272983" - "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20100101 Firefox/15.0"
75.132.185.30 - - [07/Aug/2012:23:57:27 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398260345 HTTP/1.1" 200 105 0.002335 "user_id=00201" - "Mozilla/5.0 (Windows NT 6.0; WOW64; rv:14.0) Gecko/20100101 Firefox/14.0.1"
74.69.240.195 - - [07/Aug/2012:23:57:27 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398250444 HTTP/1.1" 200 104 0.002330 "user_id=48390" - "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:14.0) Gecko/20100101 Firefox/14.0.1"
85.229.84.246 - - [07/Aug/2012:23:57:27 -0400] "POST /fight.tcl HTTP/1.1" 200 3226 0.070029 "user_id=272983" - "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20100101 Firefox/15.0"
67.172.29.94 - - [07/Aug/2012:23:57:27 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398244698 HTTP/1.1" 200 110 0.002622 "user_id=199060" - "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:10.0.6) Gecko/20100101 Firefox/10.0.6"
173.30.126.132 - - [07/Aug/2012:23:57:27 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398212035 HTTP/1.1" 200 109 0.003050 "user_id=93" - "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.0; WOW64; Trident/5.0)"
71.210.22.139 - - [07/Aug/2012:23:57:27 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398257721 HTTP/1.1" 200 111 0.002417 "user_id=267746" - "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:16.0) Gecko/16.0 Firefox/16.0"
85.229.84.246 - - [07/Aug/2012:23:57:28 -0400] "GET /parties/bits.tcl?party_id=10173 HTTP/1.1" 304 0 0.002540 "user_id=272983" - "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20100101 Firefox/15.0"
76.218.122.213 - - [07/Aug/2012:23:57:28 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398247843 HTTP/1.1" 200 111 0.002423 "user_id=890" - "Mozilla/5.0 (Windows NT 5.1; rv:14.0) Gecko/20100101 Firefox/14.0.1"
173.63.00.190 - - [07/Aug/2012:23:57:28 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398241271 HTTP/1.1" 200 109 0.002412 "user_id=272995" - "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:14.0) Gecko/20100101 Firefox/14.0.1"
75.166.203.171 - - [07/Aug/2012:23:57:28 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398247926 HTTP/1.1" 200 111 0.002284 "user_id=251543" - "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:14.0) Gecko/20100101 Firefox/14.0.1"
67.172.29.94 - - [07/Aug/2012:23:57:28 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398242252 HTTP/1.1" 200 112 0.002201 "user_id=297" - "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
141.155.116.204 - - [07/Aug/2012:23:57:28 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398245710 HTTP/1.1" 200 110 0.002267 "user_id=217405" - "Mozilla/5.0 (Windows NT 5.1; rv:14.0) Gecko/20100101 Firefox/14.0.1"
170.63.204.97 - - [07/Aug/2012:23:57:28 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398267007 HTTP/1.1" 200 109 0.002241 "user_id=51049" - "Mozilla/5.0 (X11; Linux x86_64; rv:14.0) Gecko/20100101 Firefox/14.0.1"
76.117.11.221 - - [07/Aug/2012:23:57:28 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398259710 HTTP/1.1" 200 111 0.002884 "user_id=162970" - "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20100101 Firefox/15.0"
85.229.84.246 - - [07/Aug/2012:23:57:28 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398245710 HTTP/1.1" 200 110 0.002267 "user_id=217405" - "Mozilla/5.0 (Windows NT 5.1; rv:14.0) Gecko/20100101 Firefox/14.0.1"
98.110.30.242 - - [07/Aug/2012:23:57:28 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398245710 HTTP/1.1" 200 110 0.002267 "user_id=217405" - "Mozilla/5.0 (Windows NT 5.1; rv:14.0) Gecko/20100101 Firefox/14.0.1"
74.69.240.195 - - [07/Aug/2012:23:57:28 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398245710 HTTP/1.1" 200 110 0.002267 "user_id=217405" - "Mozilla/5.0 (Windows NT 5.1; rv:14.0) Gecko/20100101 Firefox/14.0.1"
209.116.239.170 - - [07/Aug/2012:23:57:28 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398245710 HTTP/1.1" 200 110 0.002267 "user_id=230636" - "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:14.0) Gecko/20100101 Firefox/14.0.1"
71.94.234.47 - - [07/Aug/2012:23:57:28 -0400] "GET /ajax/sidebar-heartbeat.tcl?_1344398245710 HTTP/1.1" 200 110 0.002267 "user_id=271278" - "Mozilla/5.0 (Windows NT 6.1; WOW64; rv:14.0) Gecko/20100101 Firefox/14.0.1"
```

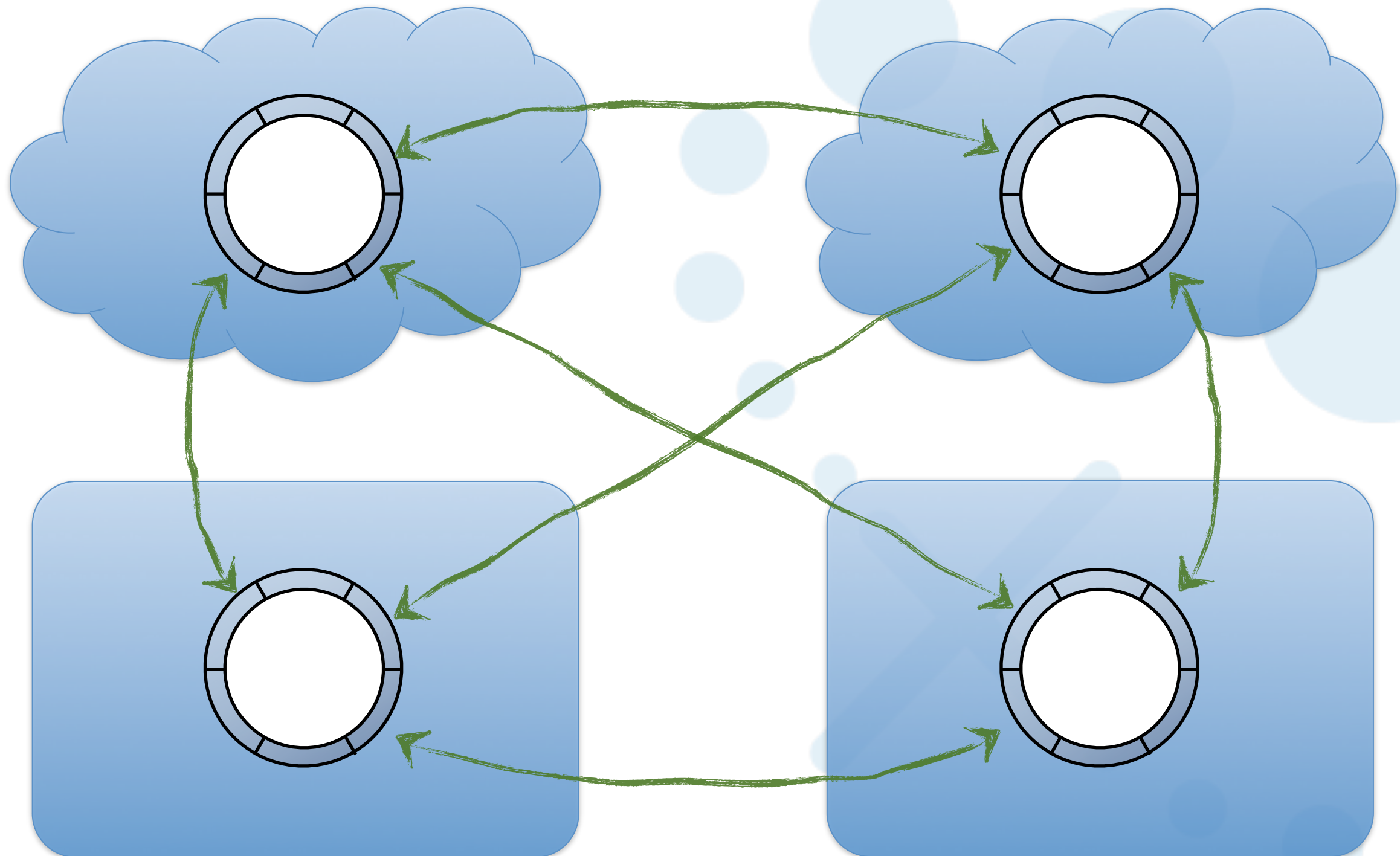
Zoom: 1d 5d 1m 3m 6m YTD 1y 5y 10y All

Aug 03, 2012 12:08 Price: 21.075 Vol: 307.73k

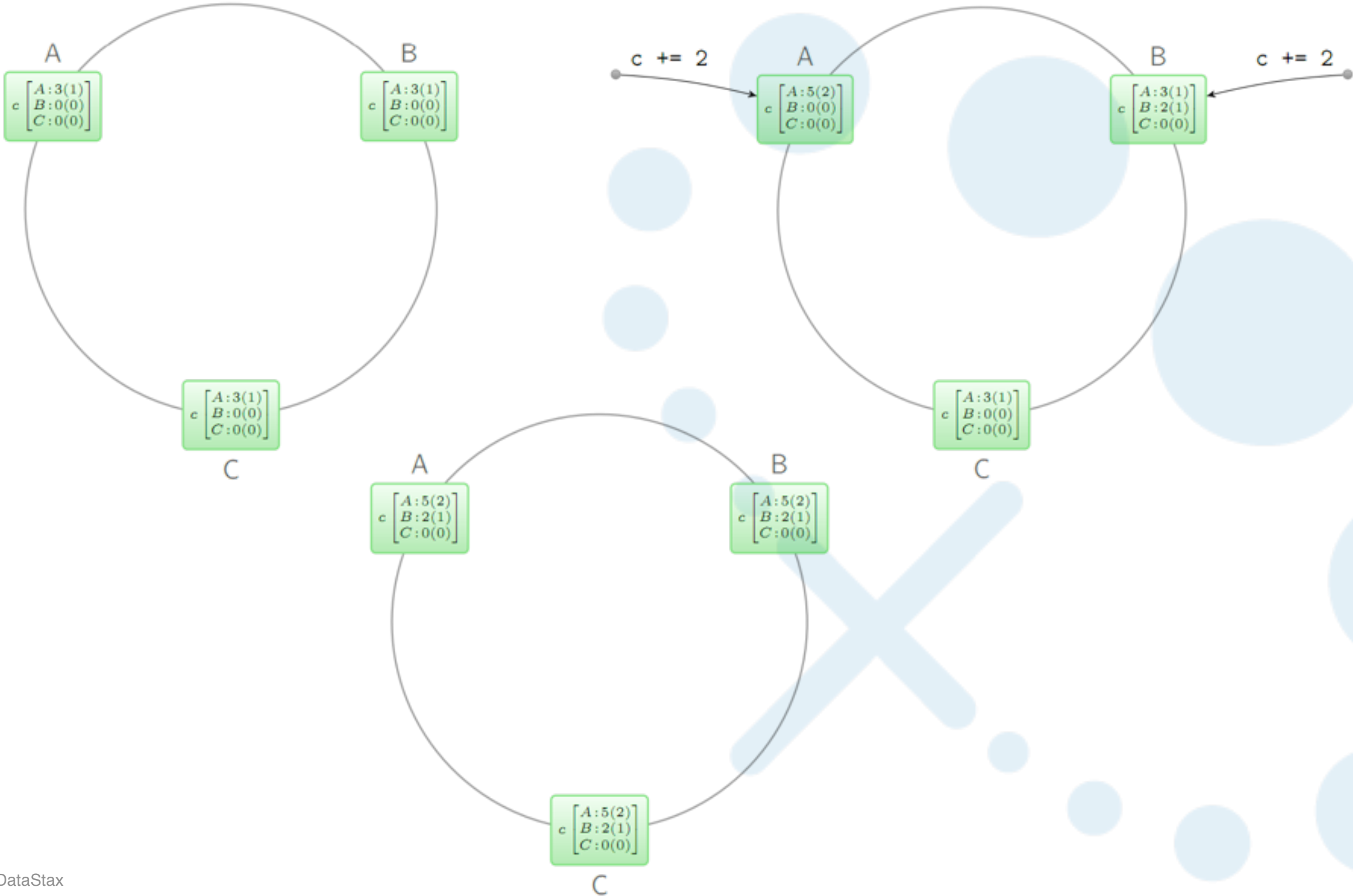
Volume (mil / 2min)



# Multi-Datacenter Support



# Distributed counters



# Hadoop support



# Disney



## Application/Use Case

- Meet the data management needs of user facing applications across The Walt Disney Company with a single platform

## Why Cassandra?

- DataStax Enterprise can tackle real-time and search functions in the same cluster
- Scalability
- 24x7 uptime



# Multitenancy



# Multi-tenancy

The screenshot displays the DataStax OpsCenter Enterprise interface. At the top left is the DataStax logo and 'OpsCenter Enterprise'. On the top right are links for 'Add a cluster' and 'Help'. Below the logo is a navigation bar with tabs for 'Overview', 'Development', 'Staging', and 'Production'. The 'Production' tab is active, indicated by a red badge with the number '2'. The main content area is titled 'Overview 3 Clusters: 13 nodes up'. Below this is a section for 'Active Alerts (2)' with a table listing alerts in effect, their descriptions, and the clusters they affect. The table has three columns: 'In effect', 'Alert', and 'Cluster'. The first alert is 'Write Request Latency at 20,150 μs/op - 101.202.203.105' on the 'Production' cluster, with an 'In effect' time of '8 minutes'. The second alert is 'Local Read Latency at 30,822 μs/op for Keyspace1.ColFam1 - 101.202.203.106' on the 'Production' cluster, with an 'In effect' time of '1 hour 16 minutes'. Below the alerts are three panels for the clusters: 'Development' (0 alerts, 1 node up), 'Staging' (0 alerts, 4 nodes up), and 'Production' (2 alerts, 8 nodes up). Each cluster panel shows 'All agents connected', 'Storage Capacity Used / Free' with a pie chart, 'Cluster Reads & Writes' with a line graph, 'Cluster Latency' with a line graph, and 'Disk Rates' with a line graph. The 'Disk Rates' graphs are labeled 'Reads / Writes'.

**DATASTAX**  
OpsCenter Enterprise

[Add a cluster](#) | [Help](#)

**Overview** 3 Clusters: 13 nodes up

**Active Alerts (2)**

In effect ▲	Alert	Cluster
8 minutes	Write Request Latency at 20,150 μs/op - 101.202.203.105	Production
1 hour 16 minutes	Local Read Latency at 30,822 μs/op for Keyspace1.ColFam1 - 101.202.203.106	Production

**Development** 0 alerts

**1 node up**  
All agents connected

Storage Capacity Used / Free

Cluster Reads & Writes

Cluster Latency

Disk Rates  
Reads / Writes

**Staging** 0 alerts

**4 nodes up**  
All agents connected

Storage Capacity Used / Free

Cluster Reads & Writes

Cluster Latency

Disk Rates  
Reads / Writes

**Production** 2 alerts

**8 nodes up**  
All agents connected

Storage Capacity Used / Free

Cluster Reads & Writes

Cluster Latency

Disk Rates  
Reads / Writes

# Enterprise search

Cassandra ColumnFamily : Users

row_key	state	status
jake	CT	at work
jason	NY	at home
jonathan	TX	in bed

Documents in Solr Core : Users

Field	Value
user	jake
state	CT
status	at work

Field	Value
user	jason
state	NY
status	at home

Field	Value
user	jonathan
state	TX
status	in bed



# Netflix



## Application/Use Case

- General purpose backend for large scale highly available cloud based web services supporting Netflix Streaming

## Why Cassandra?

- Highly available, highly robust and no schema change downtime
- Highly scalable, optimized for SSD
- Much lower cost than previous Oracle and SimpleDB implementations
- Flexible data model
- Ability to directly influence/implement OSS feature set
- Supports local and wide area distributed operations, spanning US and Europe

# Optimized for SSD

The screenshot shows a Slideshare presentation page. The browser address bar displays [www.slideshare.net/rbranson/cassandra-and-solid-state-drives](http://www.slideshare.net/rbranson/cassandra-and-solid-state-drives). The Slideshare logo is in the top left, with a search bar and 'Upload' and 'Browse' buttons to its right. Below the navigation bar is a banner for 'Insperity. ExpensAble' with the text 'Automate your business expense reporting' and a 'Learn More' button. Below the banner are icons for 'Email', 'Favorite', 'Download', and 'Flag', and an 'Embed' link. The main content area features a large title 'CASSANDRA & SOLID STATE DRIVES' and the author 'Rick Branson, DataStax'. The background image shows a man sitting on a beach. On the left side, there is a vertical sidebar with social sharing options: 21 likes, Like button, 96 tweets, Tweet button, 15 shares, Share button, +1 button, Pin it button, and a WordPress icon.

# Open source

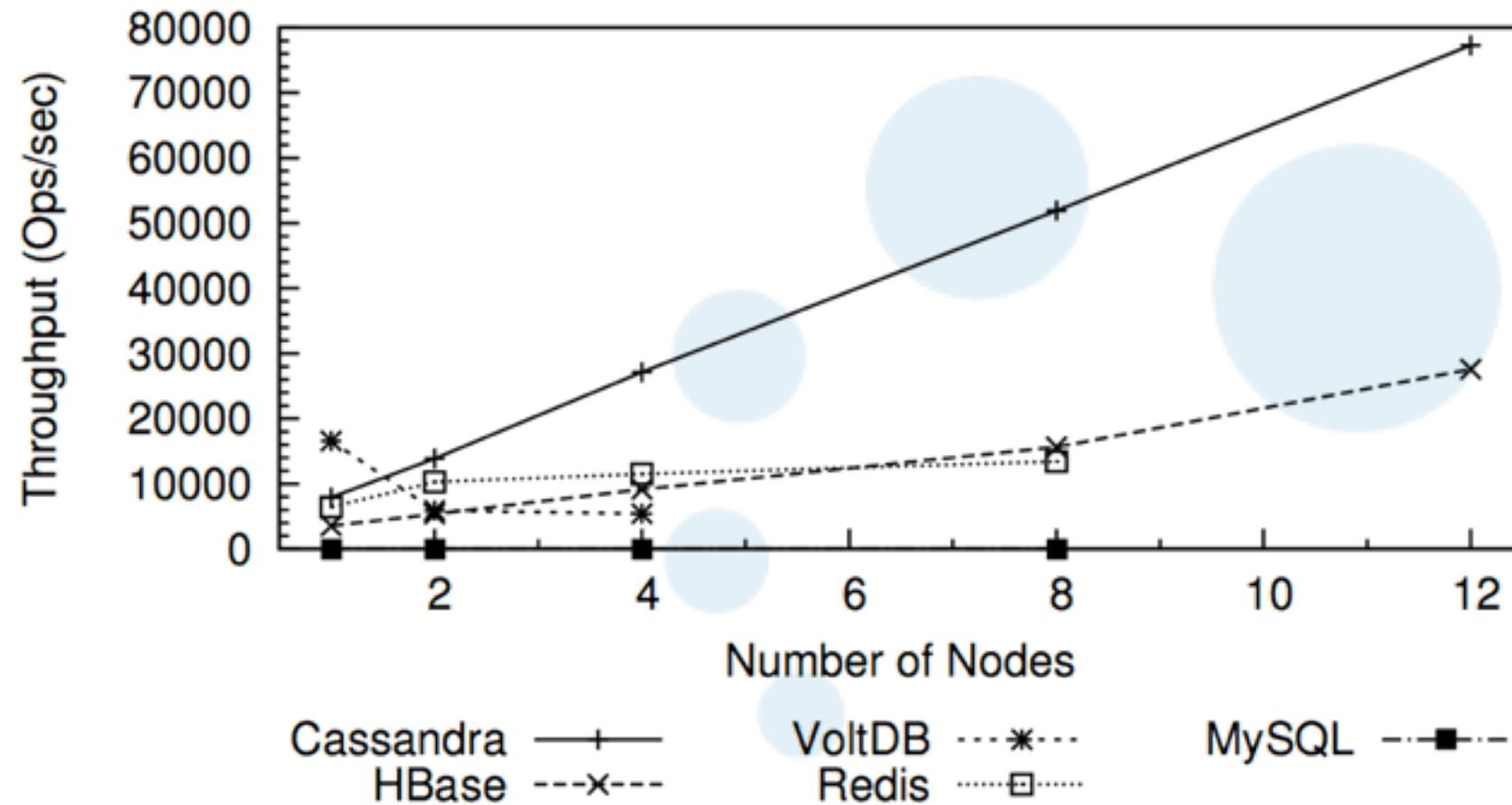




# Use case patterns

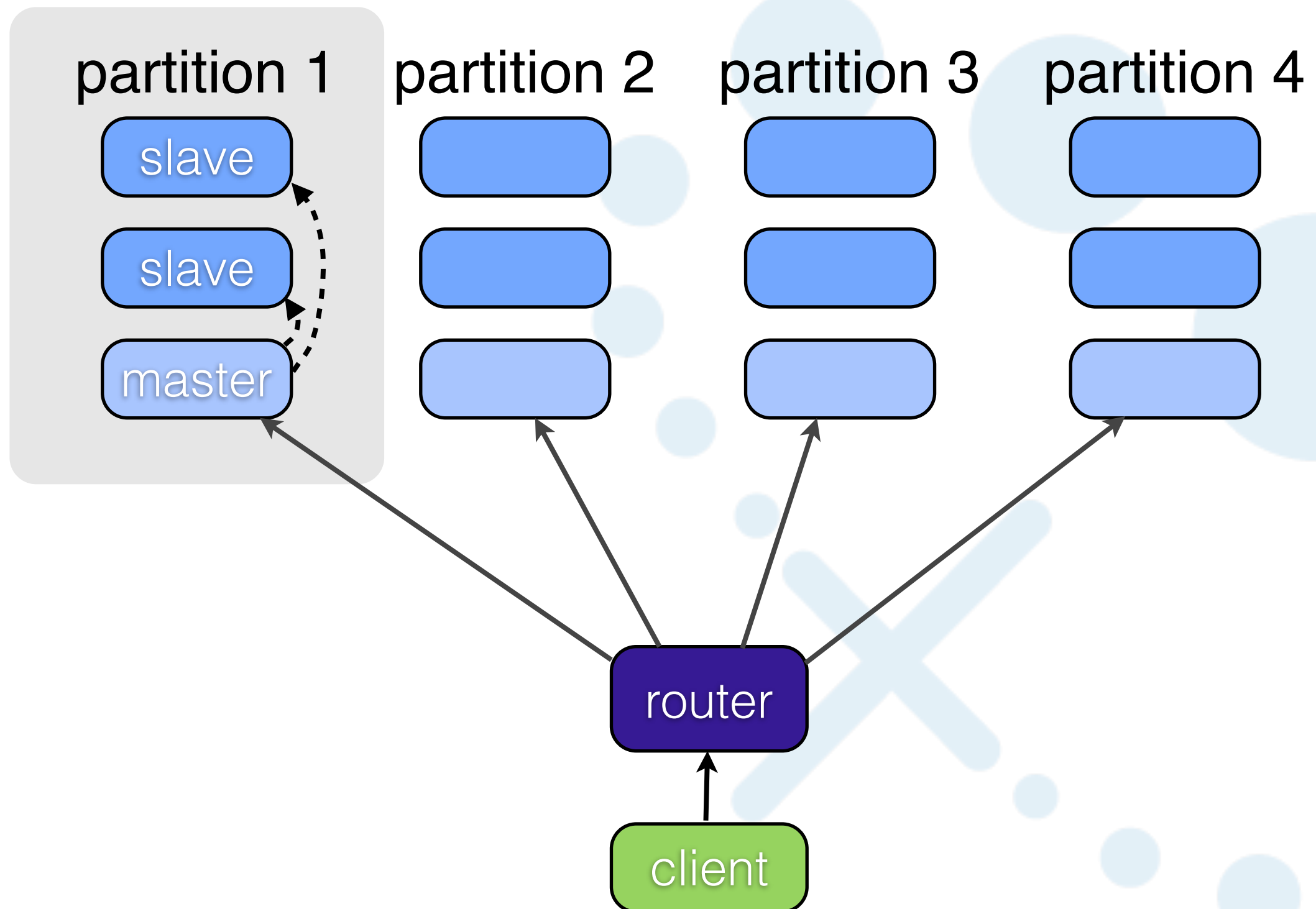
- High performance
- Massively Scalable
- Reliable/Available

# Cassandra Scalability



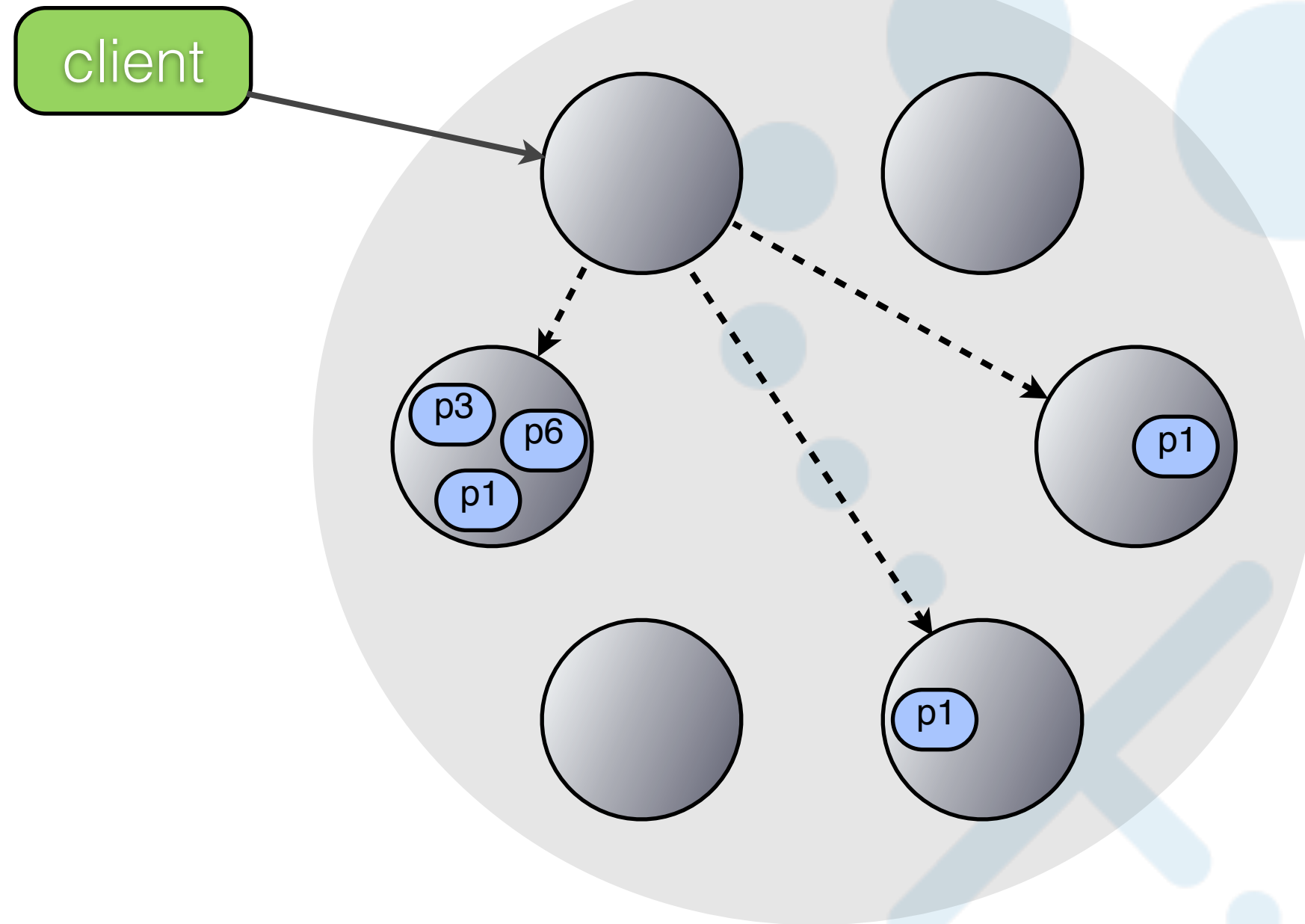
“In terms of scalability, there is a clear winner throughout our experiments. Cassandra achieves the highest throughput for the maximum number of nodes in all experiments with a linear increasing throughput from 1 to 12 nodes.”

# Classic partitioning with SPOF



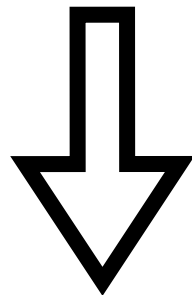


# Fully distributed, no SPOF



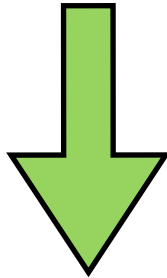
# Partitioning

Primary Key as  
“Partition Key”



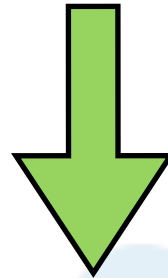
<b>jim</b>	age: 36	car: camaro	gender: M
<b>carol</b>	age: 37	car: subaru	gender: F
<b>johnny</b>	age:12	gender: M	
<b>suzy</b>	age:10	gender: F	

PK



<b>jim</b>
<b>carol</b>
<b>johnny</b>
<b>suzy</b>

Hashed Value

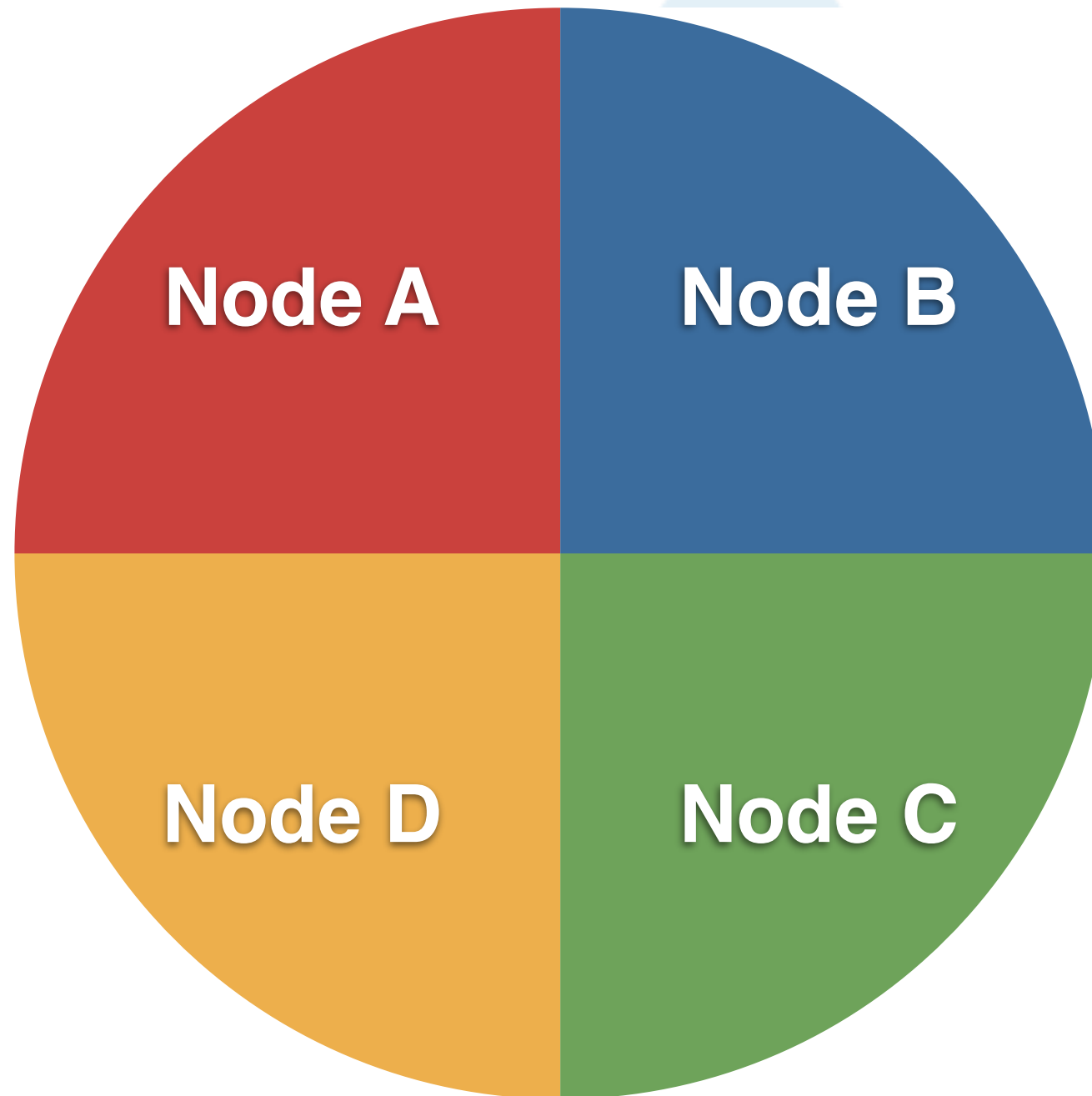


5e02739678...
a9a0198010...
f4eb27cea7...
78b421309e...

MD5\* hash operation yields a 128-bit number for keys of any size.



# The “Token Ring”



	Start	End
<b>A</b>	0xc000000000..1	0x0000000000..0
<b>B</b>	0x0000000000..1	0x4000000000..0
<b>C</b>	0x4000000000..1	0x8000000000..0
<b>D</b>	0x8000000000..1	0xc000000000..0

<b>jim</b>	5e02739678...
<b>carol</b>	a9a0198010...
<b>johnny</b>	f4eb27cea7...
<b>suzy</b>	78b421309e...

	Start	End
<b>A</b>	0xc000000000..1	0x0000000000..0
<b>B</b>	0x0000000000..1	0x4000000000..0
<b>C</b>	0x4000000000..1	0x8000000000..0
<b>D</b>	0x8000000000..1	0xc000000000..0

<b>jim</b>	5e02739678...
<b>carol</b>	a9a0198010...
<b>johnny</b>	f4eb27cea7...
<b>suzy</b>	78b421309e...





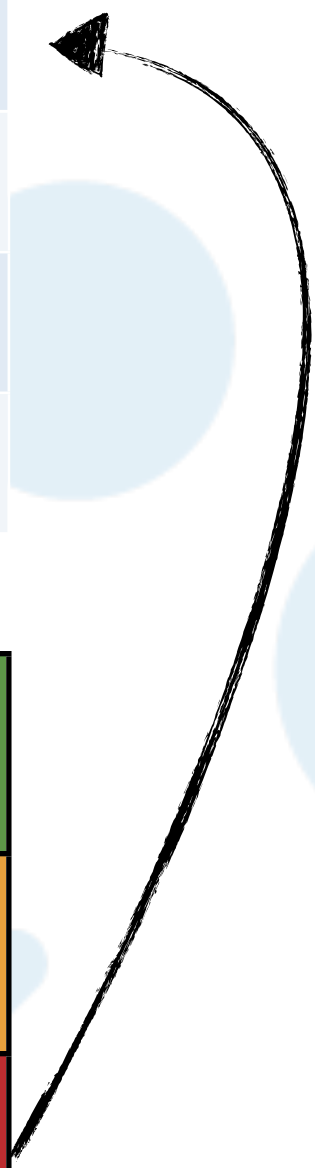
	Start	End
<b>A</b>	0xc000000000..1	0x0000000000..0
<b>B</b>	0x0000000000..1	0x4000000000..0
<b>C</b>	0x4000000000..1	0x8000000000..0
<b>D</b>	0x8000000000..1	0xc000000000..0

<b>jim</b>	5e02739678...
<b>carol</b>	a9a0198010...
<b>johnny</b>	f4eb27cea7...
<b>suzy</b>	78b421309e...



	Start	End
<b>A</b>	0xc000000000..1	0x0000000000..0
<b>B</b>	0x0000000000..1	0x4000000000..0
<b>C</b>	0x4000000000..1	0x8000000000..0
<b>D</b>	0x8000000000..1	0xc000000000..0

<b>jim</b>	5e02739678...
<b>carol</b>	a9a0198010...
<b>johnny</b>	f4eb27cea7...
<b>suzy</b>	78b421309e...

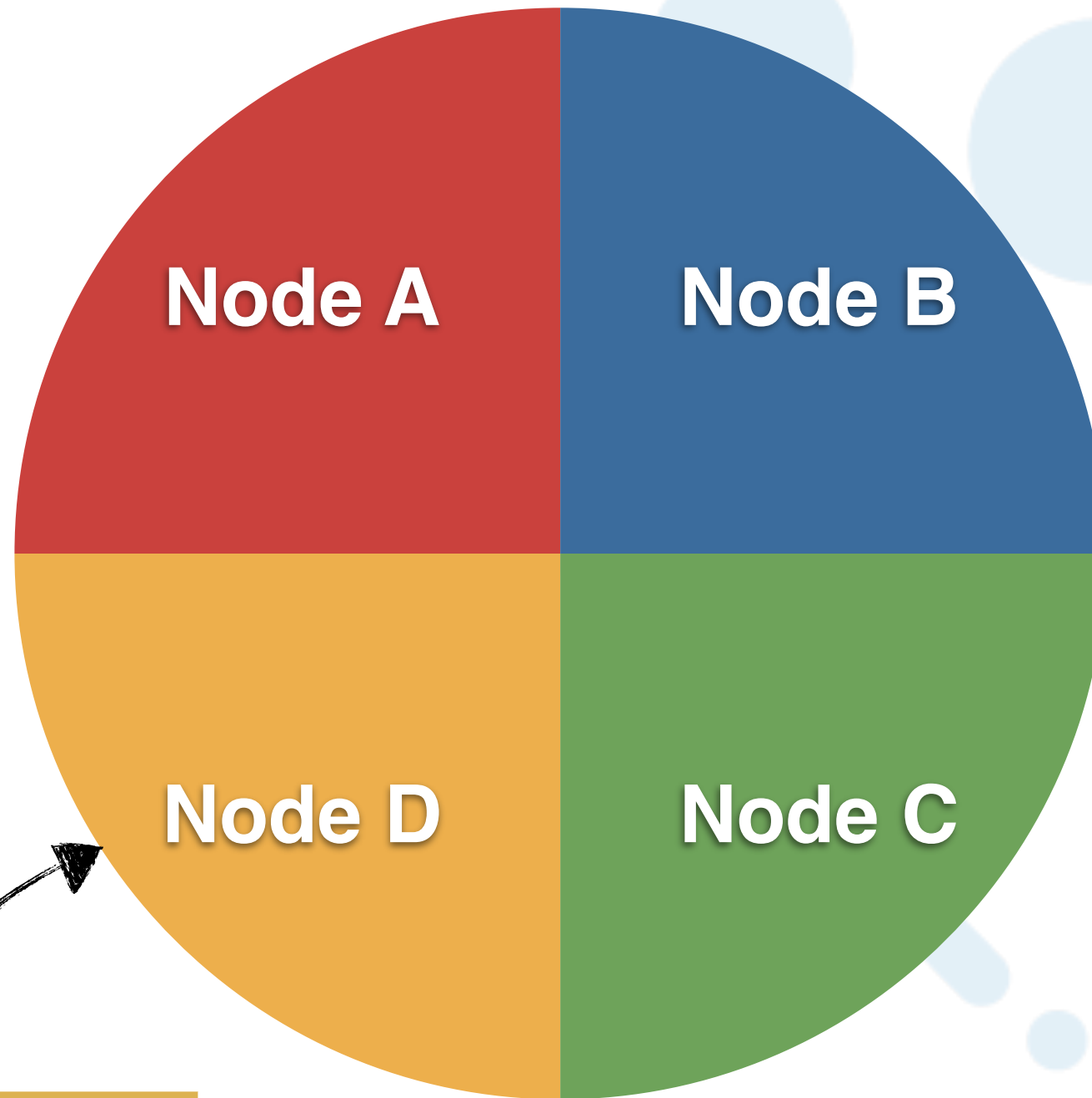


	Start	End
<b>A</b>	0xc000000000..1	0x0000000000..0
<b>B</b>	0x0000000000..1	0x4000000000..0
<b>C</b>	0x4000000000..1	0x8000000000..0
<b>D</b>	0x8000000000..1	0xc000000000..0

<b>jim</b>	5e02739678...
<b>carol</b>	a9a0198010...
<b>johnny</b>	f4eb27cea7...
<b>suzy</b>	78b421309e...



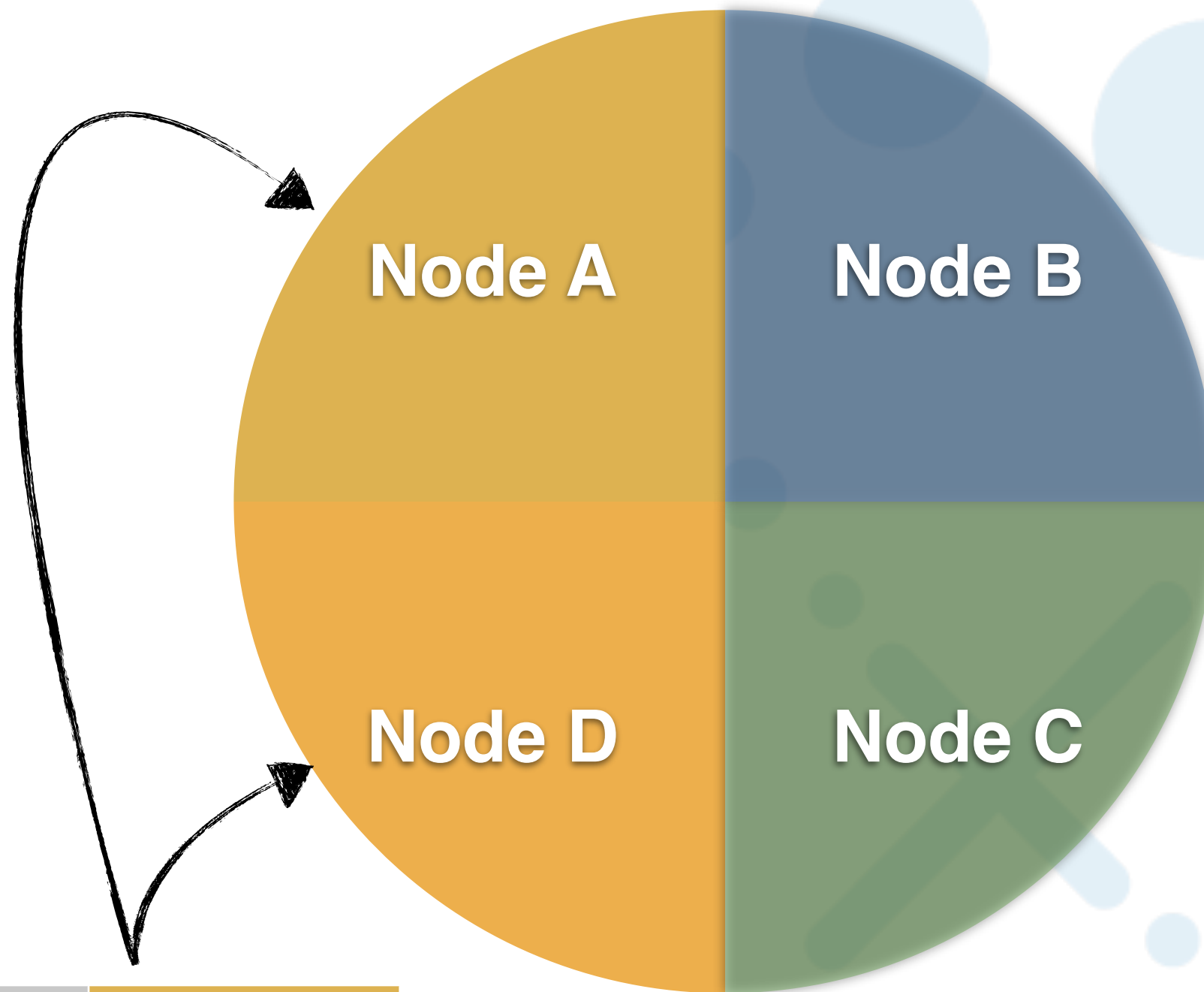
# Replication



**carol** a9a0198010...

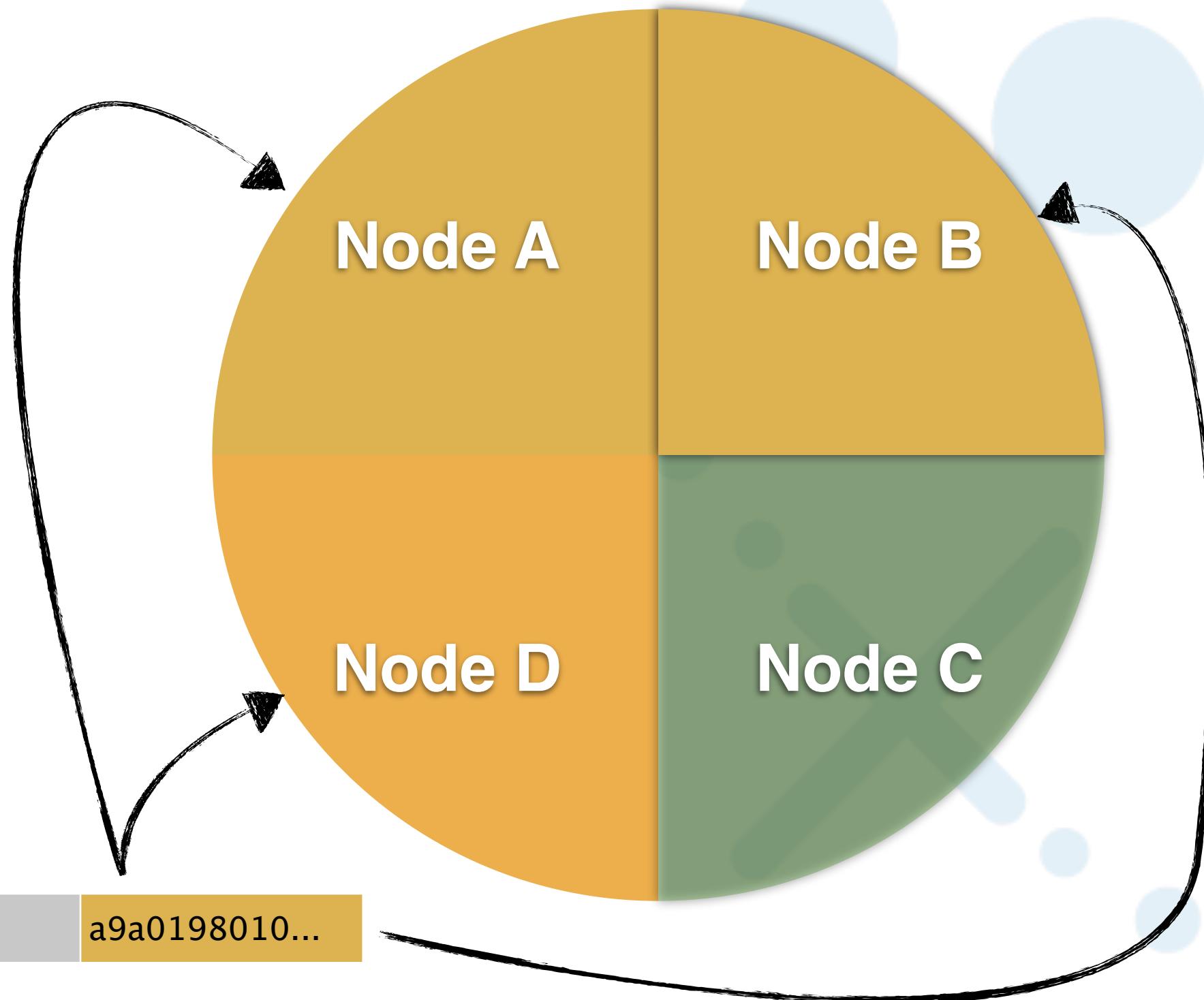


# Replication Factor = 2



carol a9a0198010...

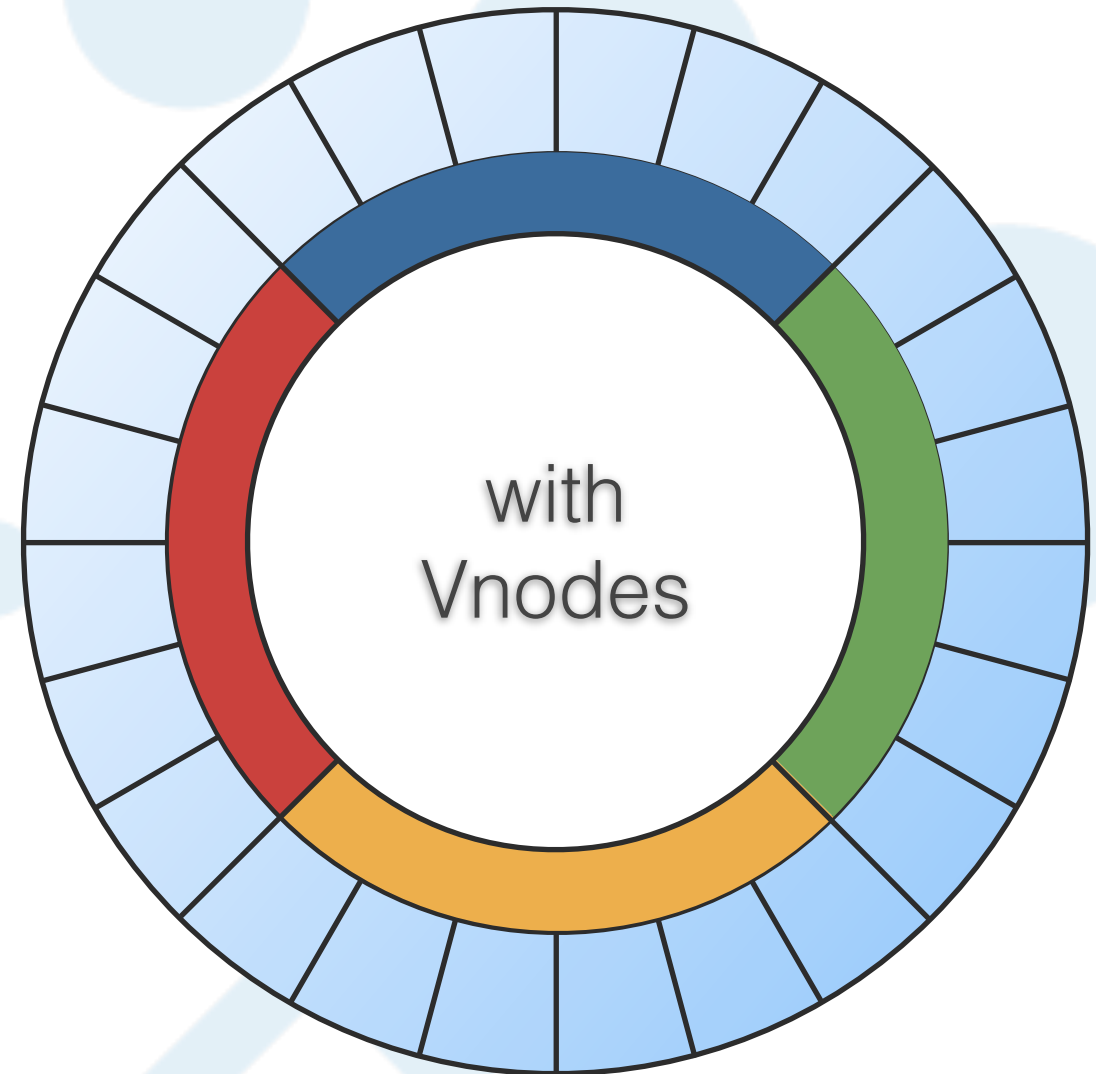
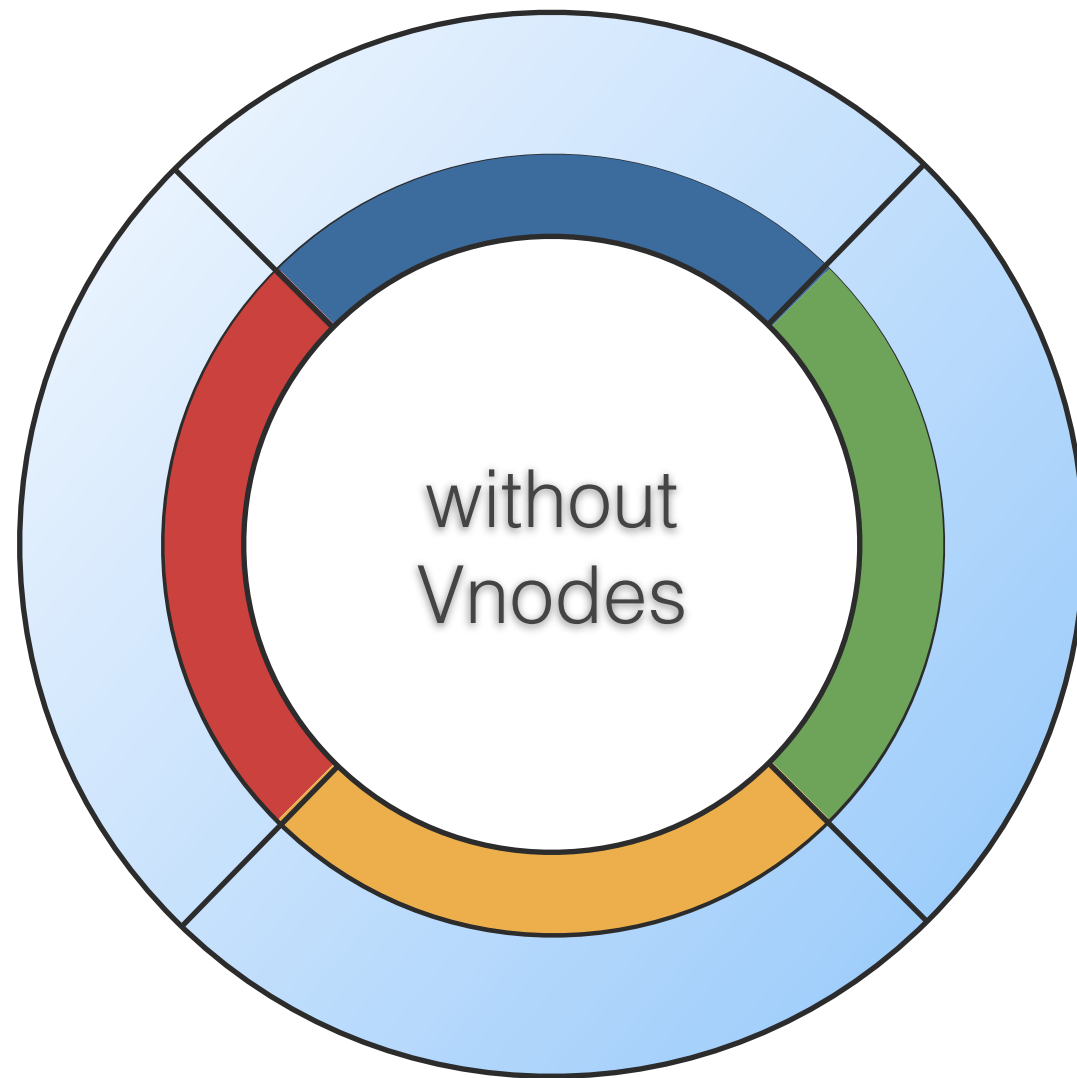
# Replication Factor = 3



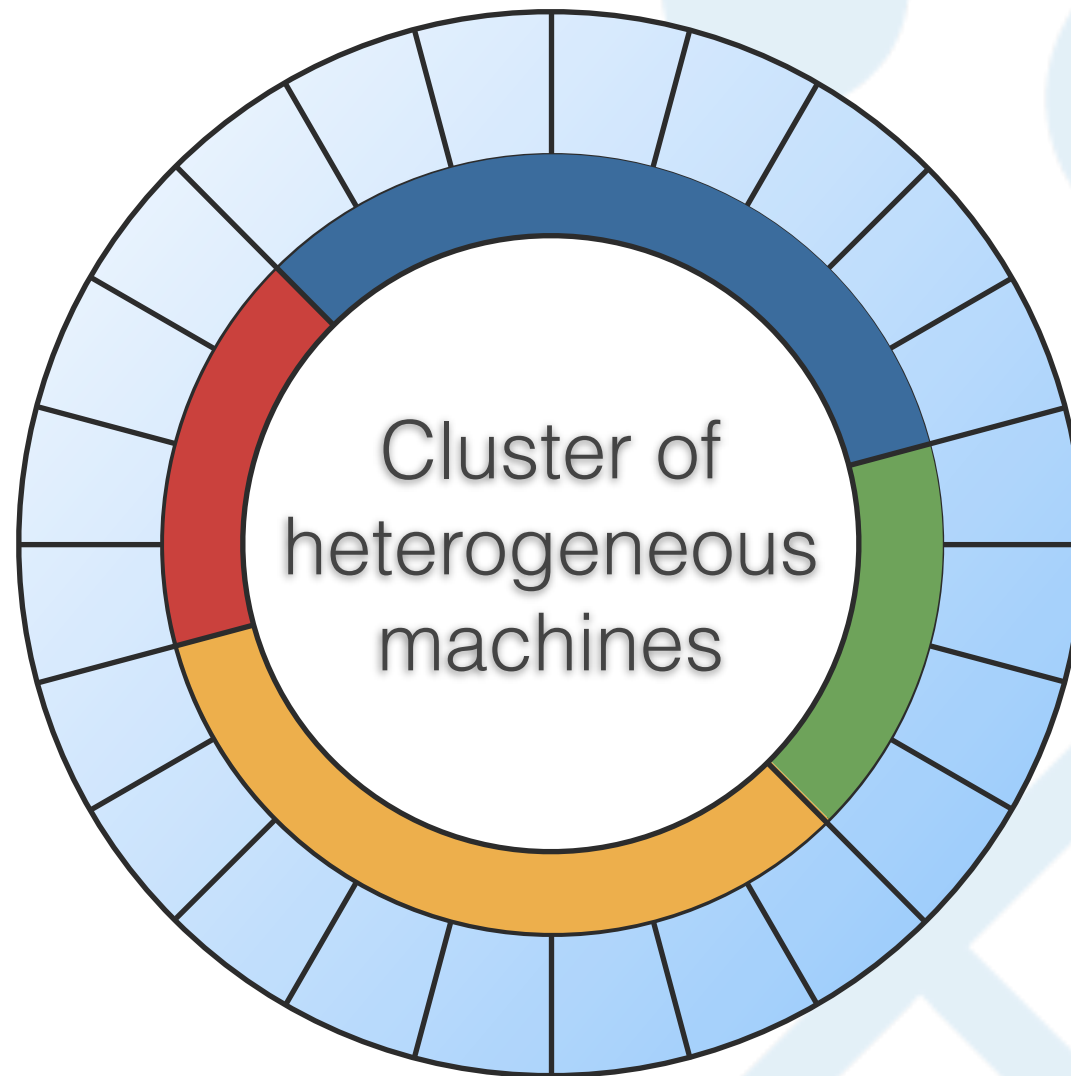
# Tunable Consistency

- Consistency Level
  - READ
    - ONE, QUORUM, LOCAL\_QUORUM, EACH\_QUORUM, ALL
  - WRITE
    - ANY, ONE, QUORUM, LOCAL\_QUORUM, EACH\_QUORUM, ALL

# Virtual Nodes

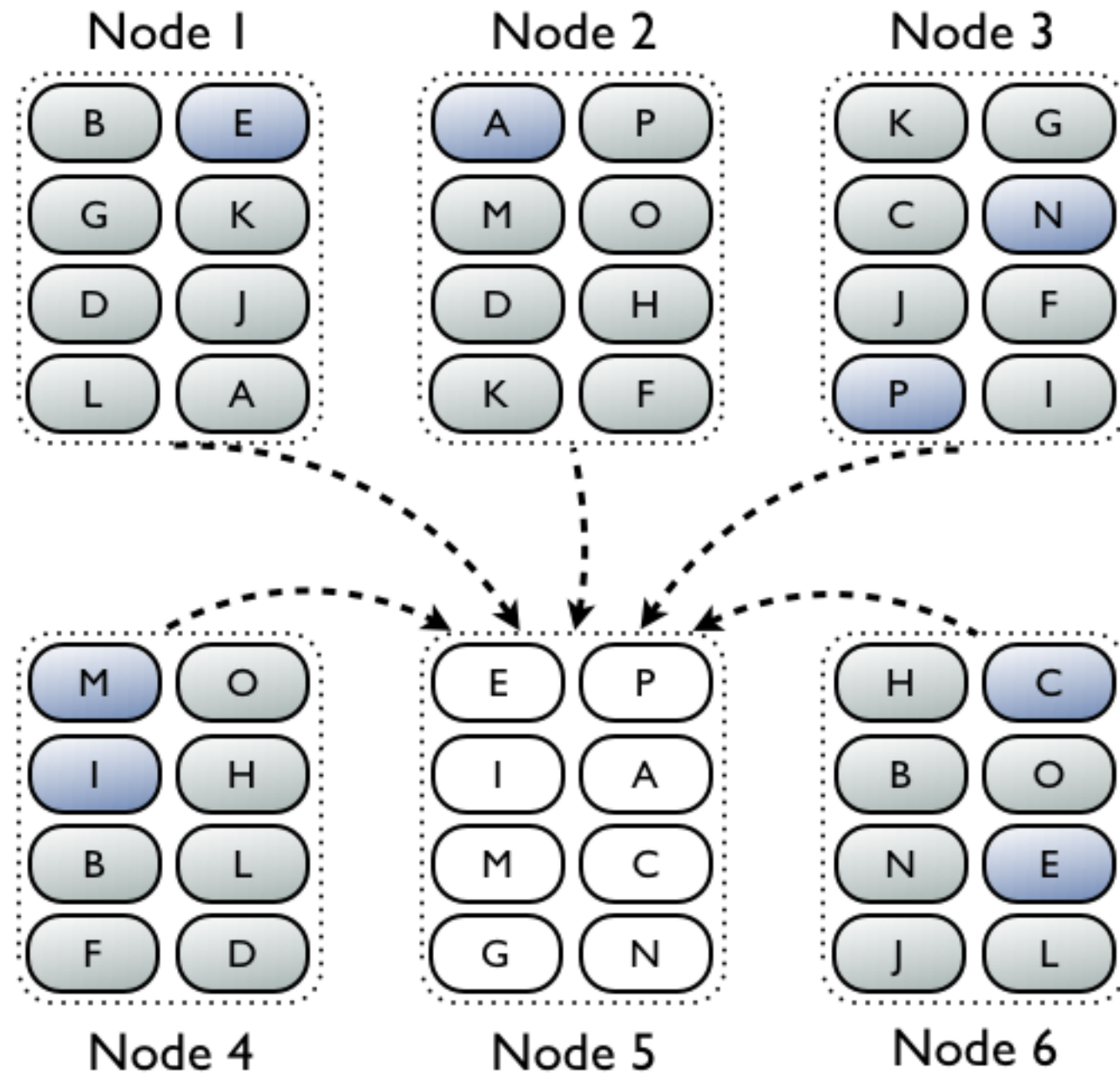
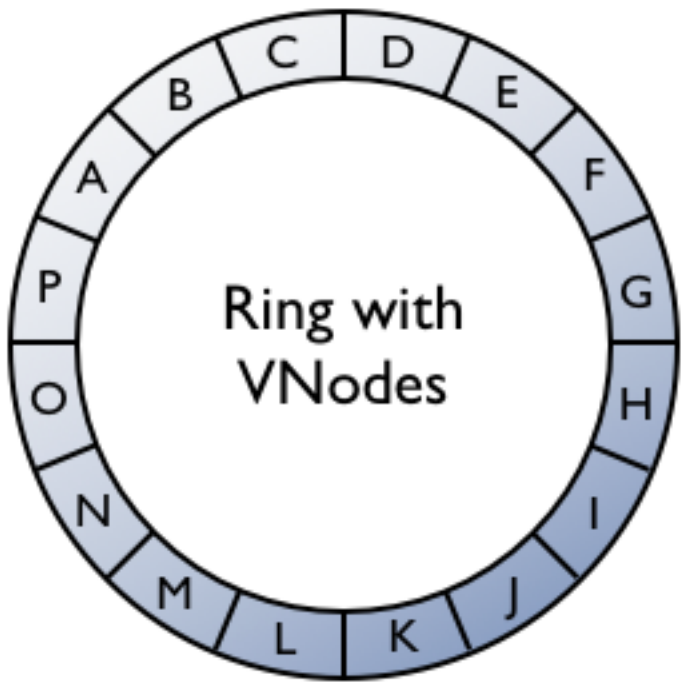


# Virtual Nodes





# Virtual Nodes



# CQL - Cassandra Query Language

```
CREATE TABLE users (  
  id uuid PRIMARY KEY,  
  name text,  
  state text,  
  birth_date int  
);
```

```
INSERT INTO users (id, name, state, birth_date)  
VALUES ('49290170-817f-11e2-9e96-0800200c9a66',  
      'john', 'Texas', 1990);
```

```
SELECT * FROM users WHERE state='Texas' AND birth_date > 1950;
```

# Strictly “realtime” focused

- No joins
- No subqueries
- No aggregation functions\* or GROUP BY
- ORDER BY?

# Collections

```
CREATE TABLE users (  
  id uuid PRIMARY KEY,  
  name text,  
  state text,  
  birth_date int  
);
```

```
CREATE TABLE users_addresses (  
  user_id uuid REFERENCES users,  
  email text  
);
```

```
SELECT *  
FROM users NATURAL JOIN users_addresses;
```



# Collections

```
CREATE TABLE users (  
  id uuid PRIMARY KEY,  
  name text,  
  state text,  
  birth_date int  
);
```

```
CREATE TABLE users_addresses (  
  user_id uuid REFERENCES users,  
  email text  
);
```

```
SELECT *  
FROM users NATURAL JOIN users_addresses;
```

# Collections

```
CREATE TABLE users (  
  id uuid PRIMARY KEY,  
  name text,  
  state text,  
  birth_date int,  
  email_addresses set<text>  
);
```

```
UPDATE users  
SET email_addresses = email_addresses + {'jbellis@gmail.com',  
jbellis@datastax.com'};
```

# Question ?

Feel free to contact me later if you have one  
[yuki@datastax.com](mailto:yuki@datastax.com)  
yukim (IRC, twitter)

DATASTAX