Time Series Data With Apache Cassandra

Strangeloop
September 19, 2014

Eric Evans
eevans@opennms.org
@jericevans

OpenNMS®

Open m&m's

openNMS®

openNMS®

Open

openNMS®

Open

openNMS®

Network

Management

System

openNMS®

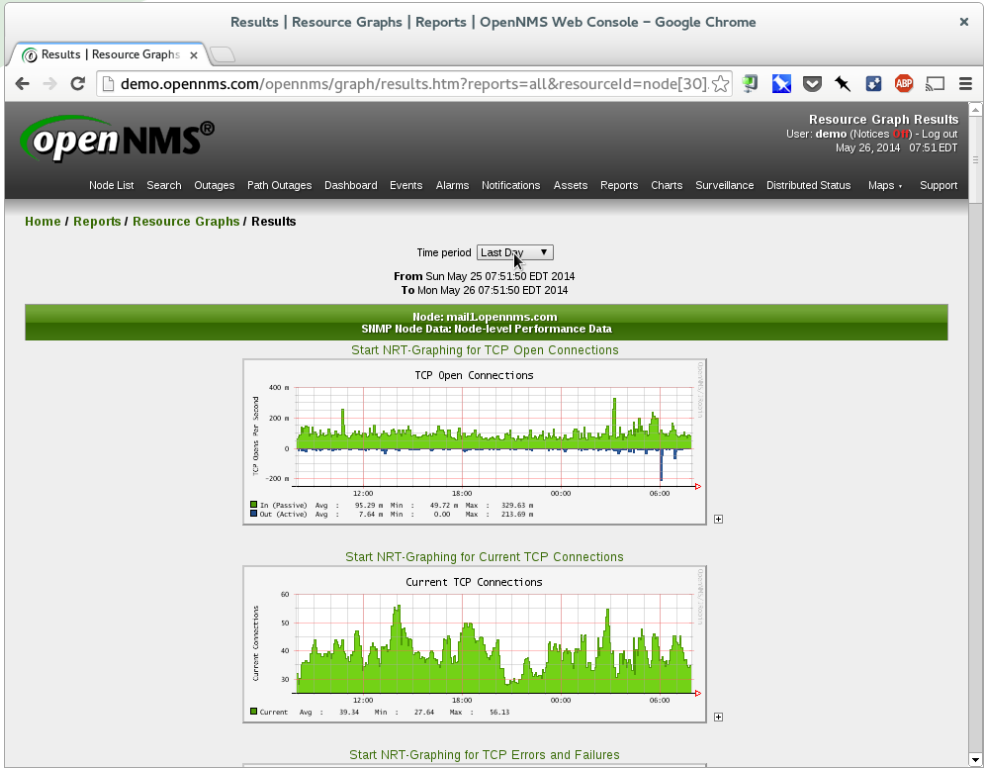
OpenNMS: What It Is

- Network Management System
  - Discovery and Provisioning
  - Service monitoring
  - Data collection
  - Event management, notifications
- Java, open source, GPLv3
- Since 1999

Time series: RRDTool

- Round Robin Database
- First released 1999
- Time series storage
- File-based, constant-size, self-maintaining
- Automatic, incremental aggregation

… and oh yeah, graphing



Consider

- 5+ IOPs per update (read-modify-write)!
- 100,000s of metrics, 1,000s IOPS
- 1,000,000s of metrics, 10,000s IOPS
- 15,000 RPM SAS drive, ~175-200 IOPS

Hmmm

We collect and write a great deal; We read (graph) relatively little.

So why are we aggregating everything?

Also

- Not everything is a graph
- Inflexible
- Incremental backups impractical
- Availability subject to filesystem access

TIL

Metrics typically appear in groups that are accessed together.

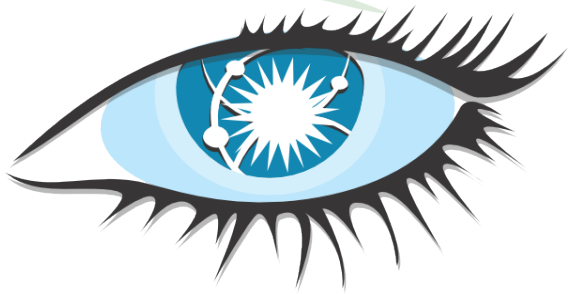
Optimizing storage for grouped access is a great idea!

What OpenNMS needs:

- High throughput
- High availability
- Late aggregation
- Grouped storage/retrieval

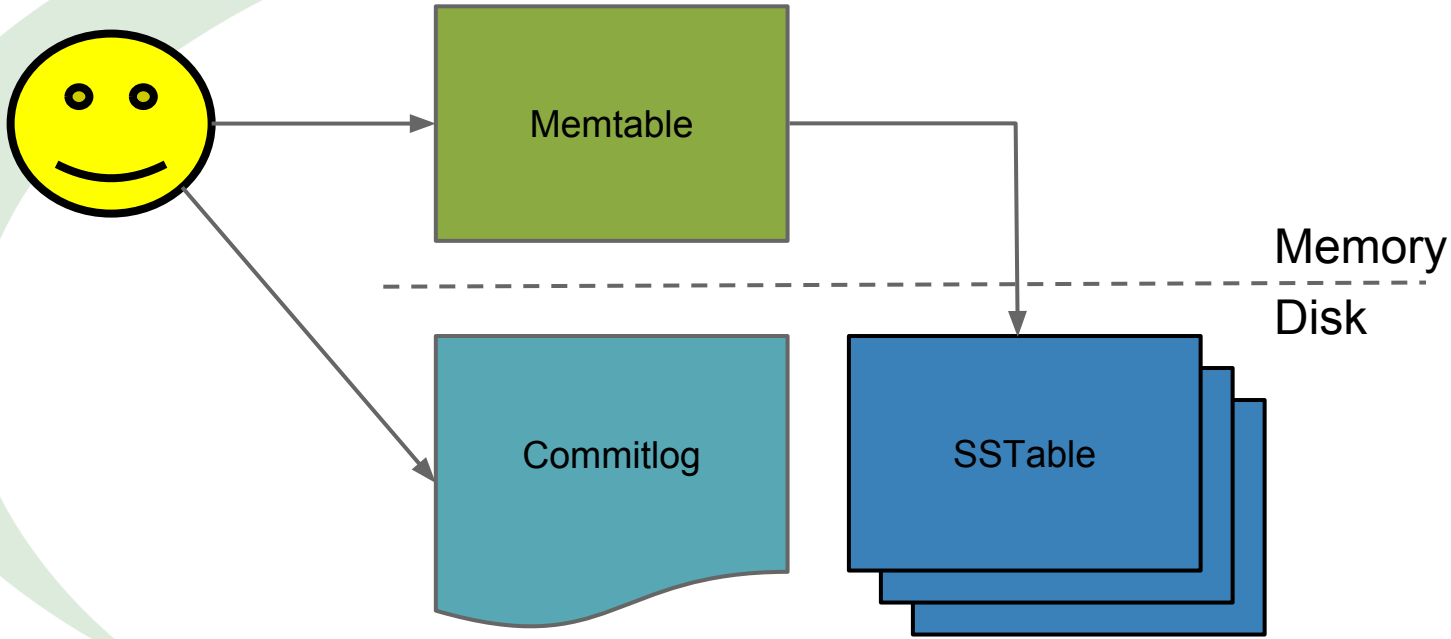
openNMS®

Cassandra



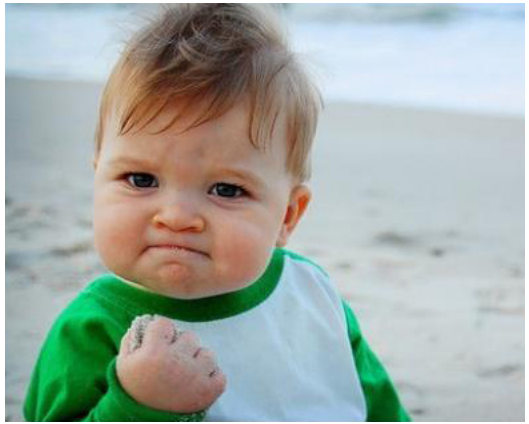
- Apache top-level project
- Distributed database
- Highly available
- High throughput
- Tunable consistency

Writes

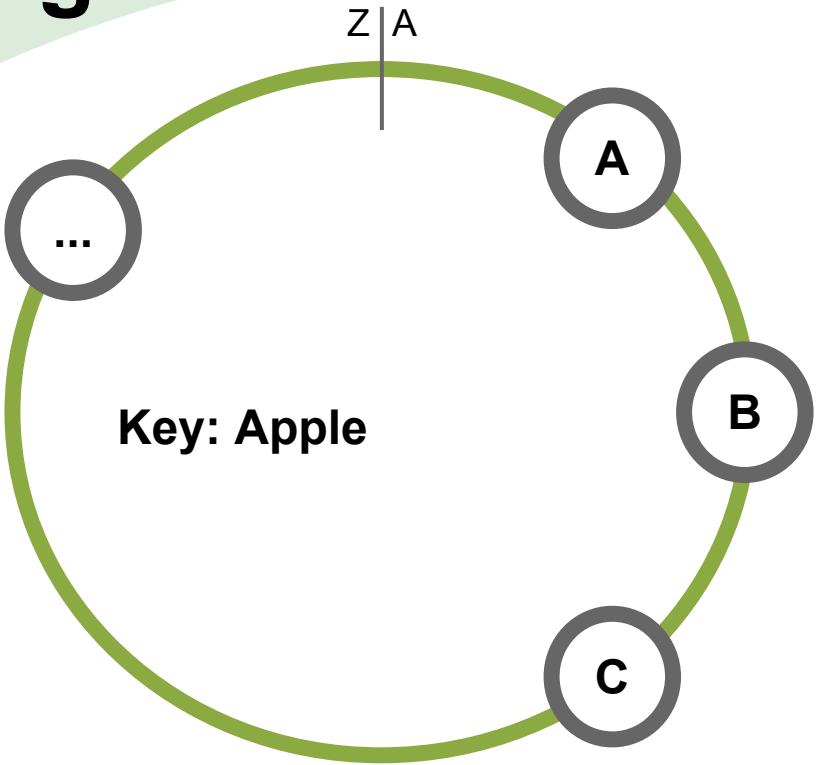


Write Properties

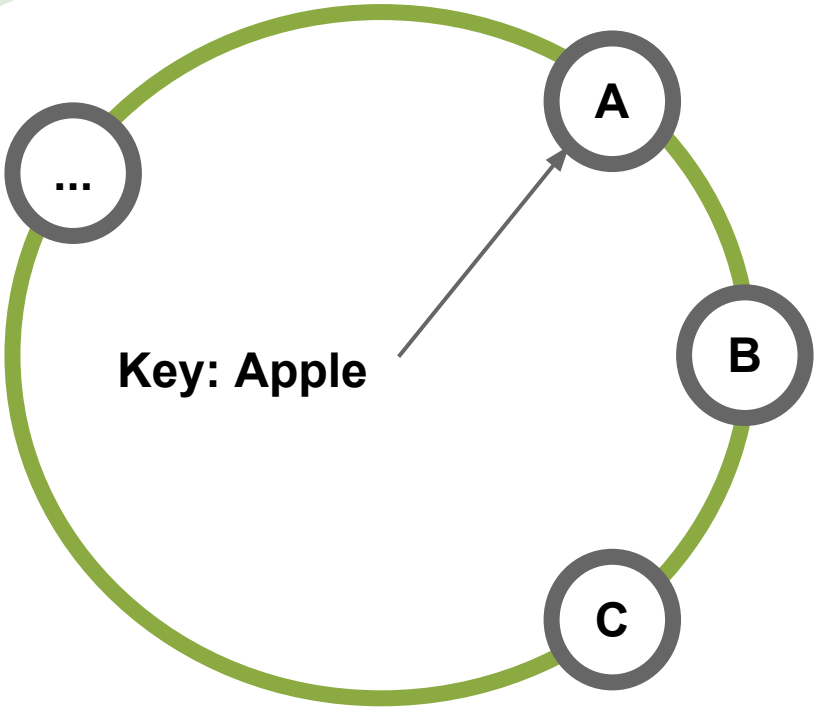
- Optimized for write throughput
- Sorted on disk
- Perfect for time series!



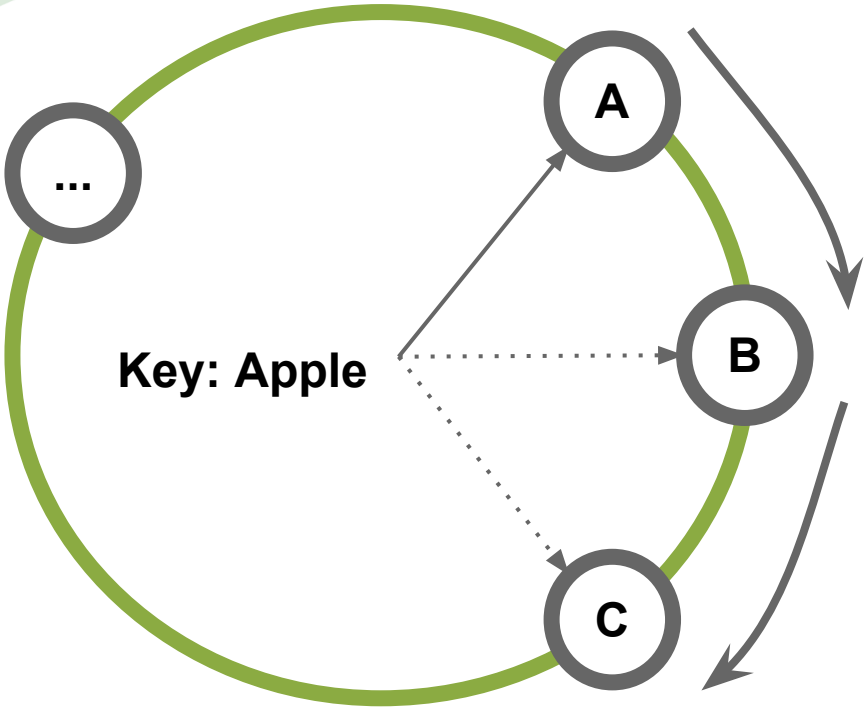
Partitioning



Placement



Replication



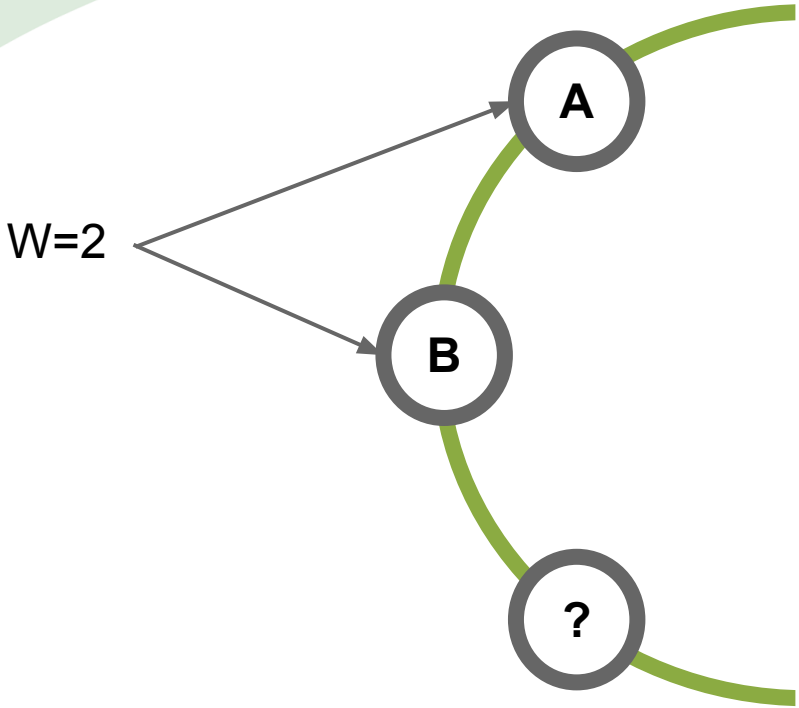
CAP Theorem

Consistency

Availability

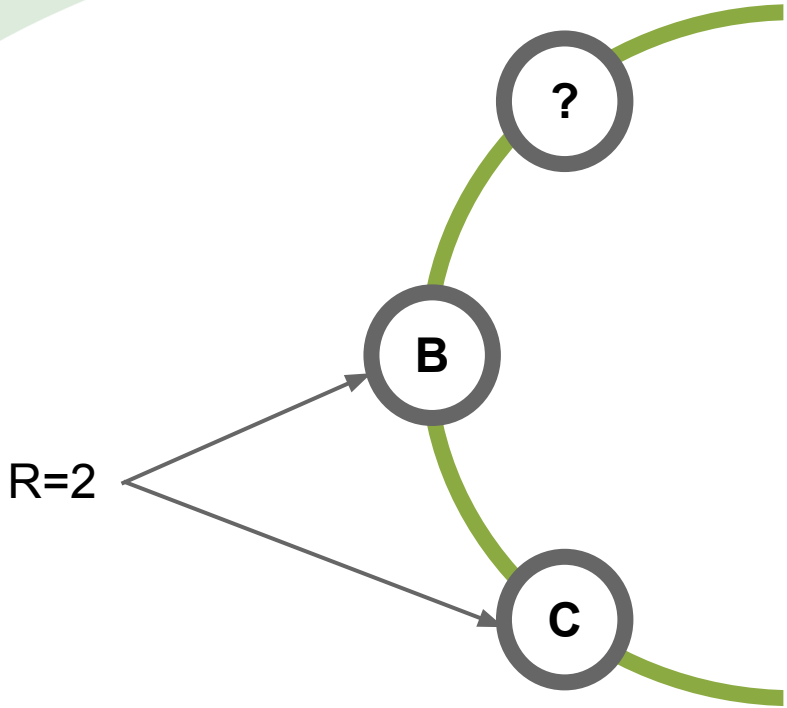
Partition tolerance

Consistency



Consistency

R+W > N



openNMS®

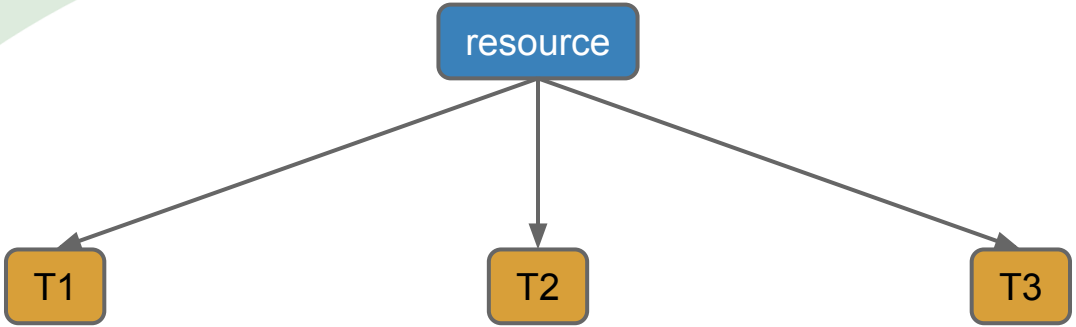
Distribution Properties

- Symmetrical
- Linearly scalable
- Redundant
- Highly available

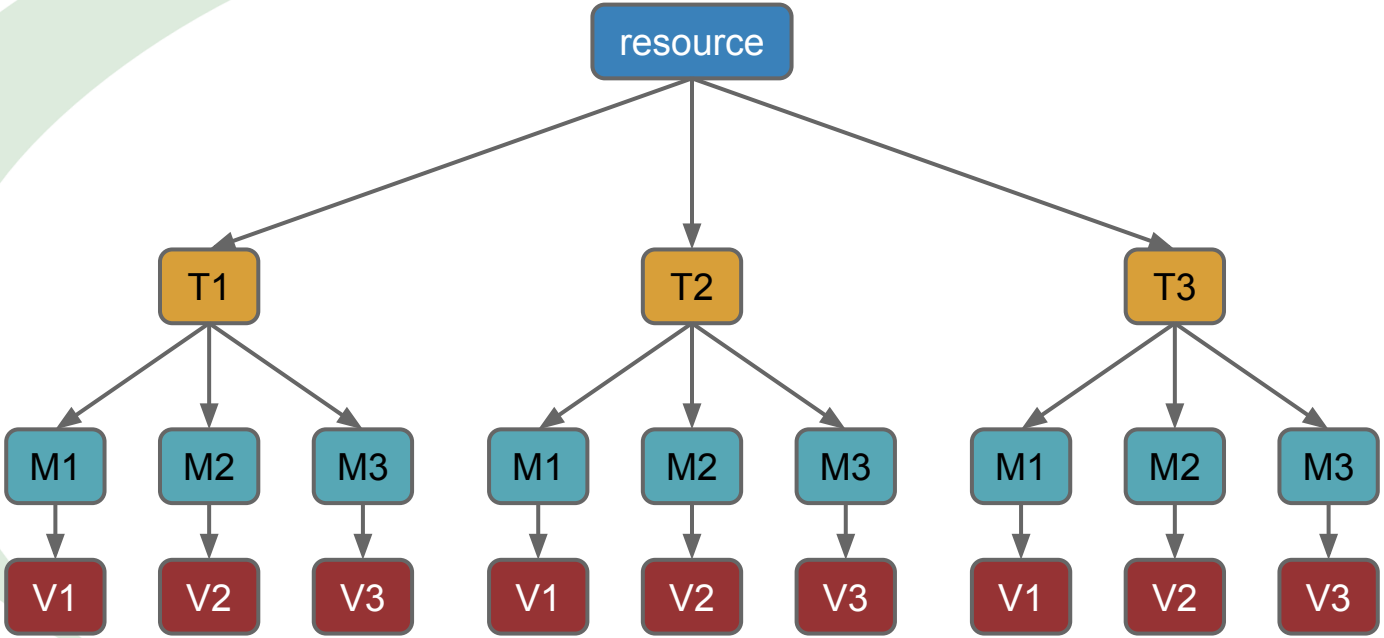
OpenNMS®

Data Model

Data Model



Data Model



Data Model

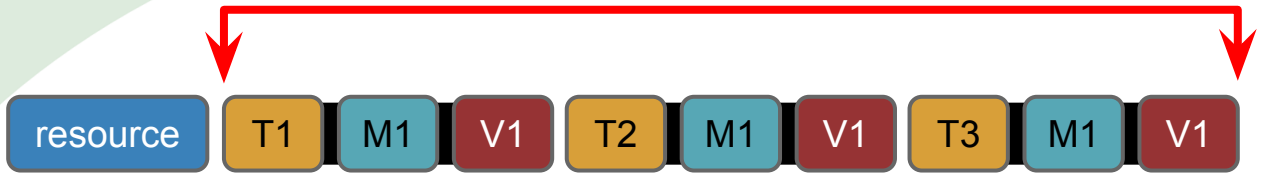
```
CREATE TABLE samples (
    T timestamp,
    M text,
    V double,
    resource text,
    PRIMARY KEY(resource, T, M)
);
```

Data model

Data model

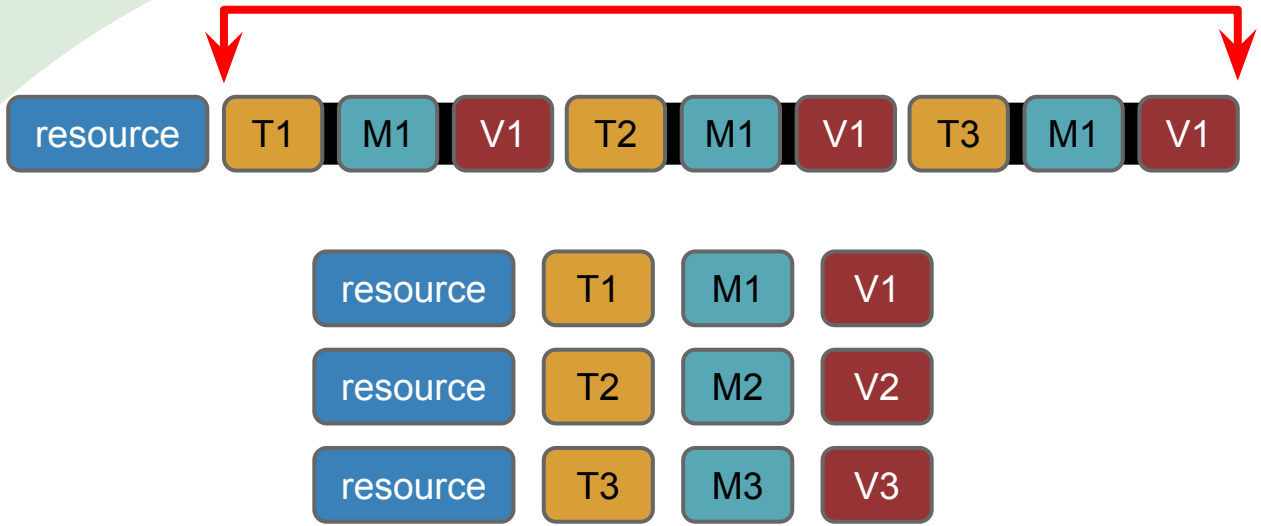

```
SELECT * FROM samples
WHERE resource = 'resource'
AND T >= 'T1' AND T <= 'T3';
```

Data model




```
SELECT * FROM samples
WHERE resource = 'resource'
AND T >= 'T1' AND T <= 'T3';
```

Data model


resource
T1
M1
V1
T2
M1
V1
T3
M1
V1
resource
T1
M1
V1
resource
T2
M2
V2
resource
T3
M3
V3

Newts

- Standalone time series data-store
  - Java API
  - REST interface
- Raw sample storage and retrieval
- Flexible aggregations (computed at read)
  - Rate (from counter types)
  - Pluggable aggregation functions
  - Arbitrary calculations

Newts

- Cassandra-speed
- Resource search indexing (preliminary)
- Approaching “1.0”
- Apache license
- Github (http://github.com/OpenNMS/newts)
- http://newts.io

Fin

openNMS®