Time Series Data With Apache Cassandra

Berlin Buzzwords
May 27, 2014

Eric Evans
eevans@opennms.org
@jericevans

openNMS®

Open m&m's®

Open

Open

openNMS®

Open

Network

Management

System

openNMS®

OpenNMS: What It Is

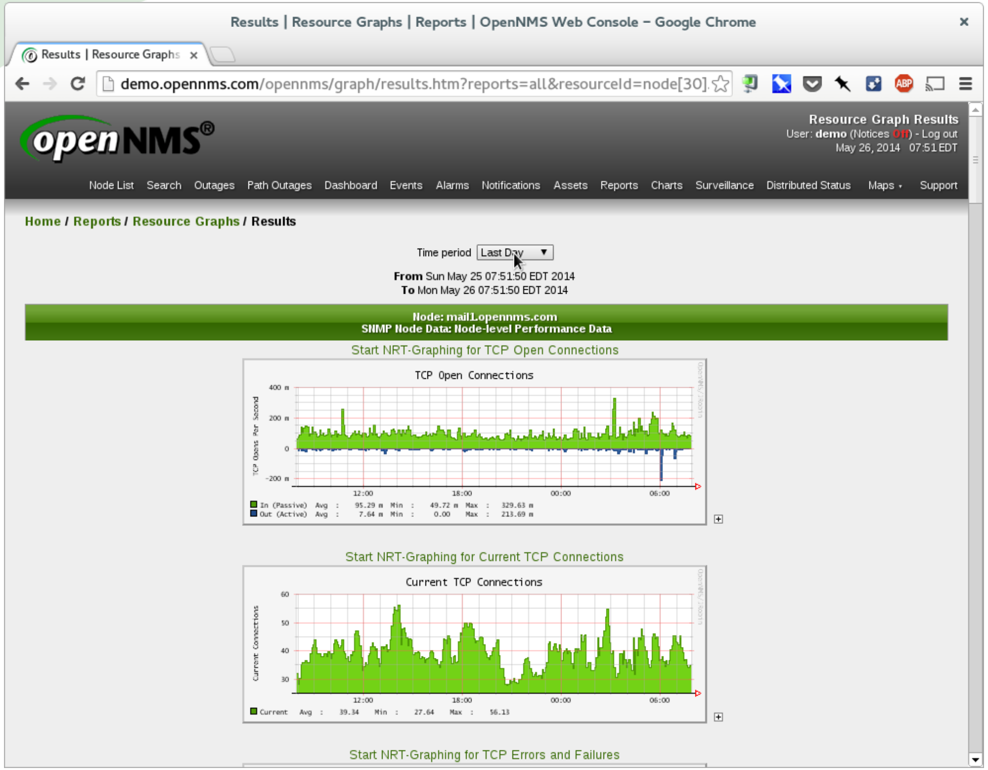
- Network Management System
    - Discovery and Provisioning
    - Service monitoring
    - Data collection
    - Event management, notifications
- Java, open source, GPLv3
- Since 1999

Time series: RRDTool

RRDtool
logging & graphing

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

… and oh yeah, graphing



Results | Resource Graphs | Reports | OpenNMS Web Console – Google Chrome

demo.opennms.com/opennms/graph/results.htm?reports=all&resourceId=node[30].

Resource Graph Results
User: demo (Notices Off) - Log out
May 26, 2014 07:51 EDT

Home / Reports / Resource Graphs / Results

Time period: Last Day

From Sun May 25 07:51:50 EDT 2014
To Mon May 26 07:51:50 EDT 2014

Node: mail1.opennms.com
SNMP Node Data: Node-level Performance Data

Start NRT-Graphing for TCP Open Connections

```
■ In (Passive)  Avg :   95.29 m  Min :   49.72 m  Max :  329.63 m
■ Out (Active)  Avg :    7.64 m  Min :    0.00    Max :  213.69 m
```

Start NRT-Graphing for Current TCP Connections

```
■ Current   Avg :   39.34   Min :   27.64   Max :   56.13
```

Start NRT-Graphing for TCP Errors and Failures

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

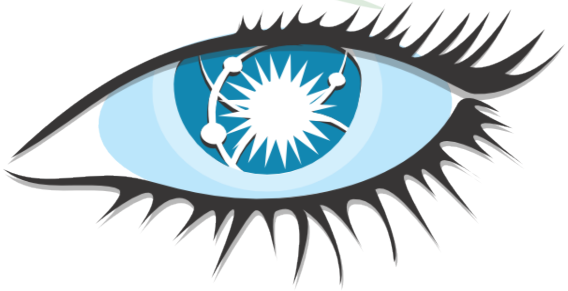
Metrics typically appear in groups that are accessed together.

Optimizing storage for grouped access is a great idea!

What OpenNMS needs:

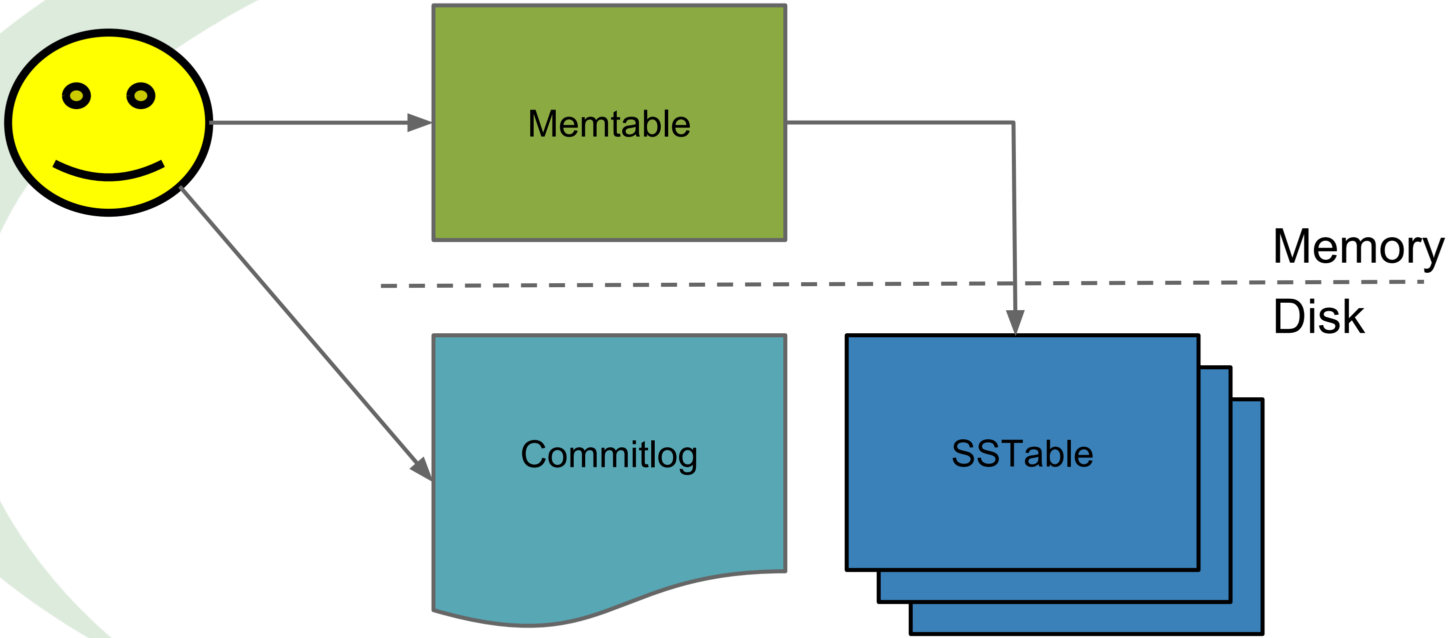
- High throughput
- High availability
- Late aggregation
- Grouped storage/retrieval

Cassandra



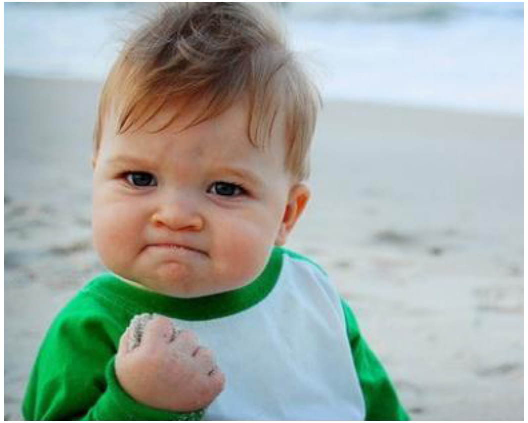
- Apache top-level project
- Distributed database
- Highly available
- High throughput
- Tunable consistency

Writes

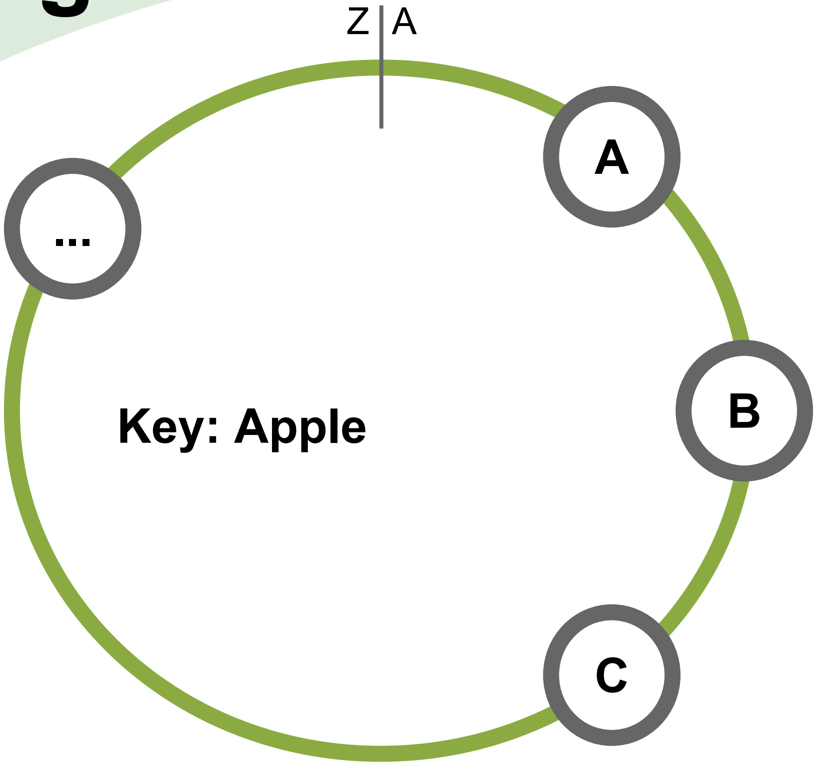


Write Properties

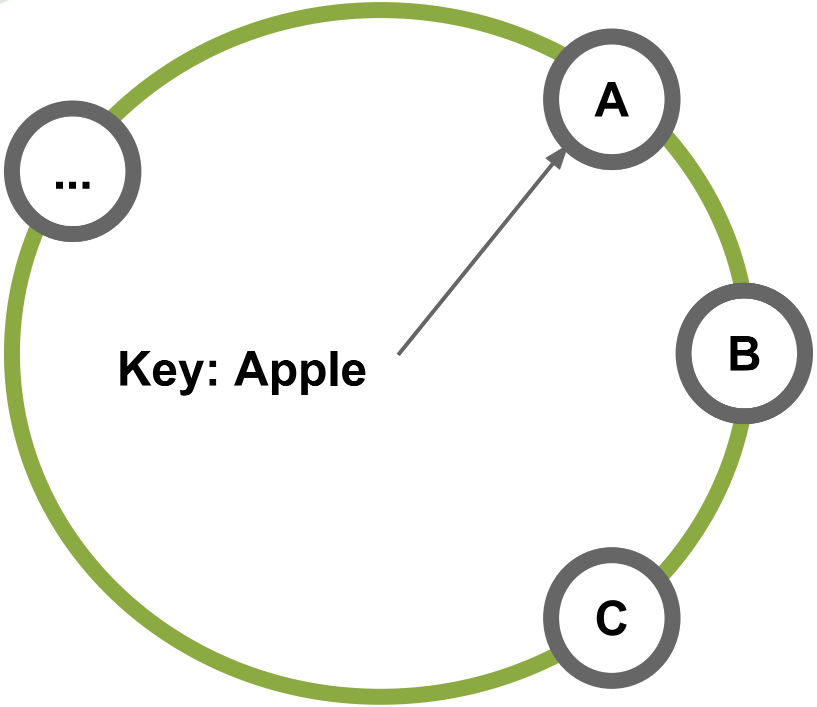
- Optimized for write throughput
- Sorted on disk
- Perfect for time series!



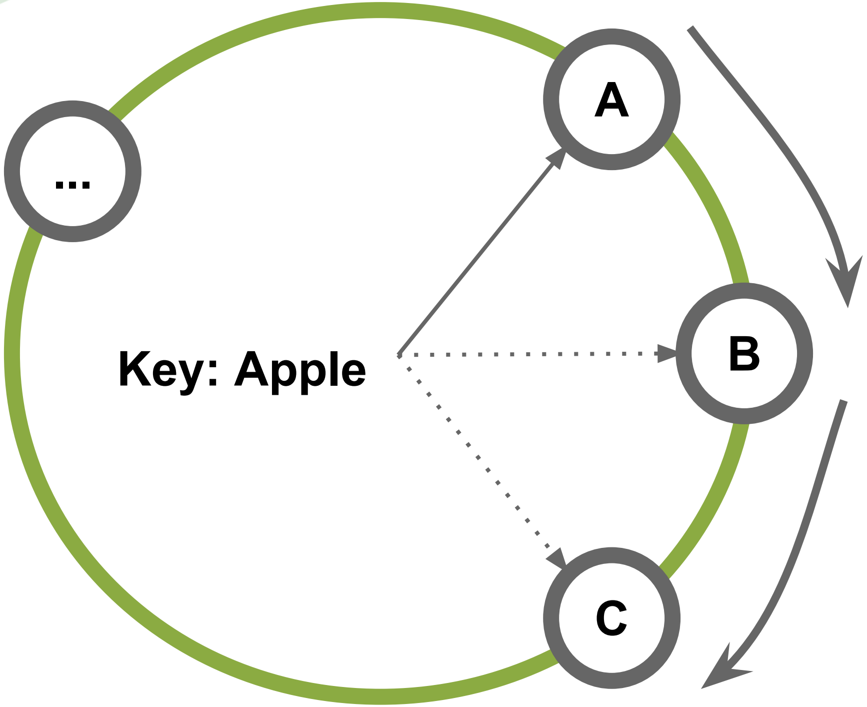
Partitioning



Placement



Replication



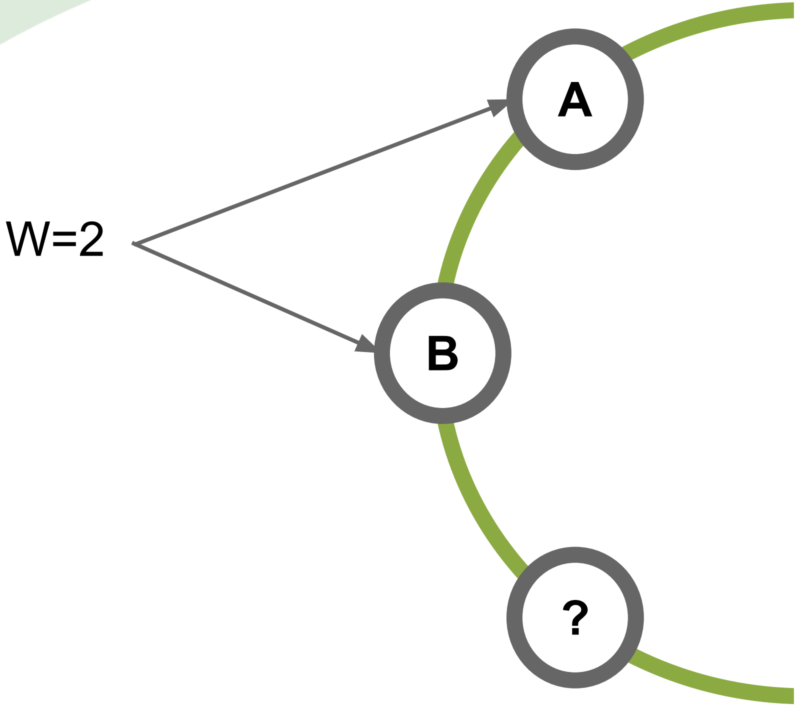
CAP Theorem

Consistency

Availability

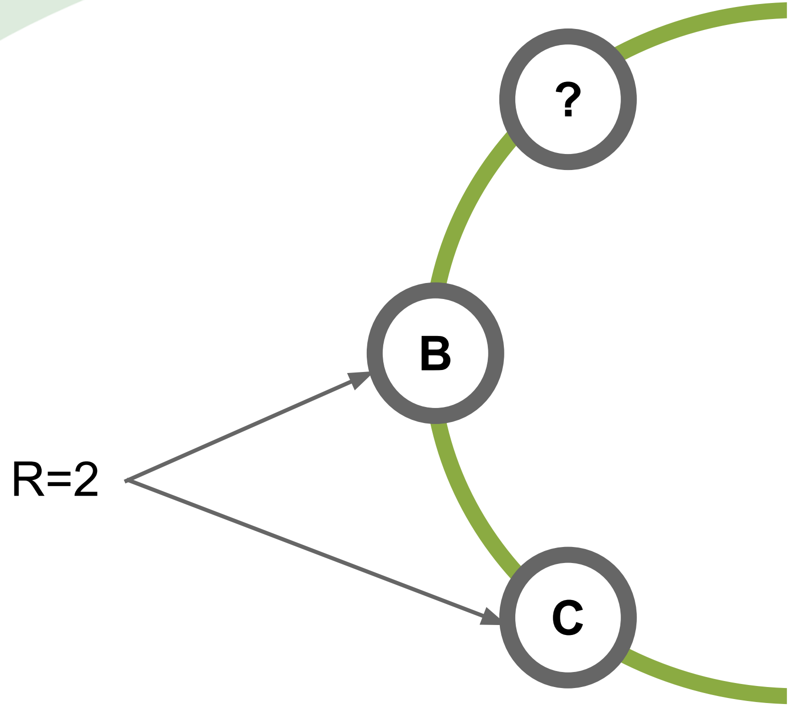
Partition tolerance

Consistency



Consistency

R+W > N



Distribution Properties

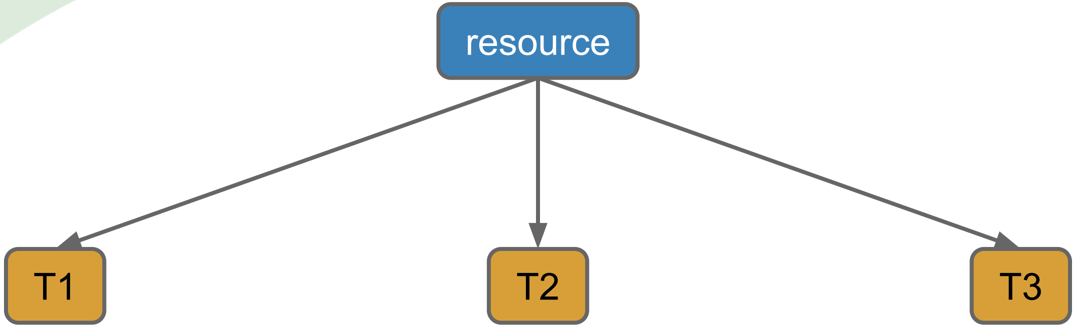
- Symmetrical
- Linearly scalable
- Redundant
- Highly available

Data Model

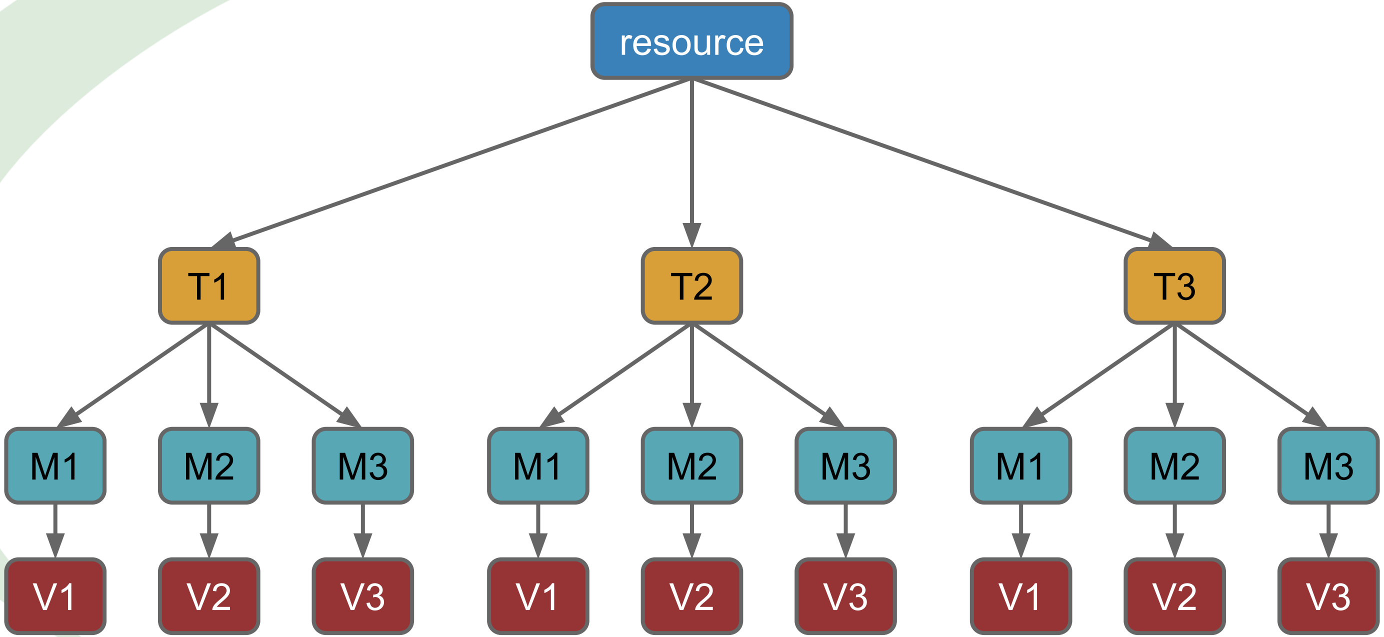
Data Model

resource

Data Model



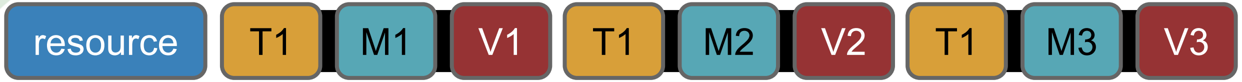
Data Model


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

Data Model

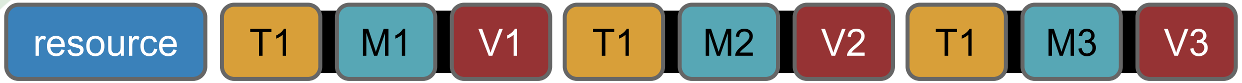
```
CREATE TABLE samples (
    T timestamp,
    M text,
    V double,
    resource text,
    PRIMARY KEY(resource, T, M)
);
```

Data model



resource T1 M1 V1 T1 M2 V2 T1 M3 V3

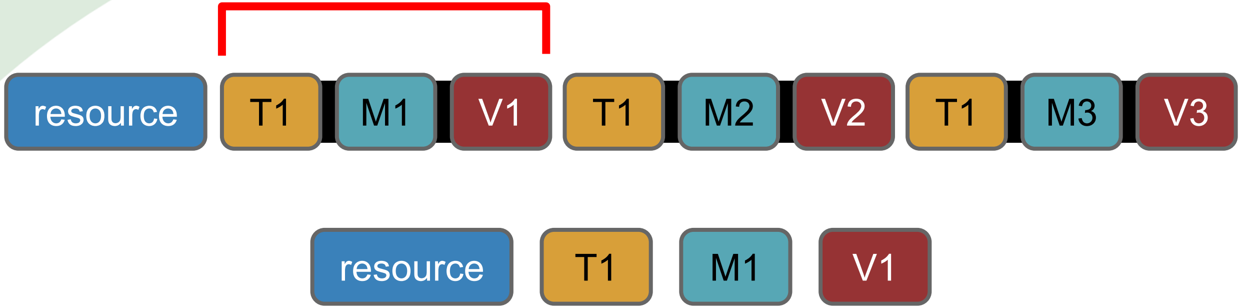
Data model



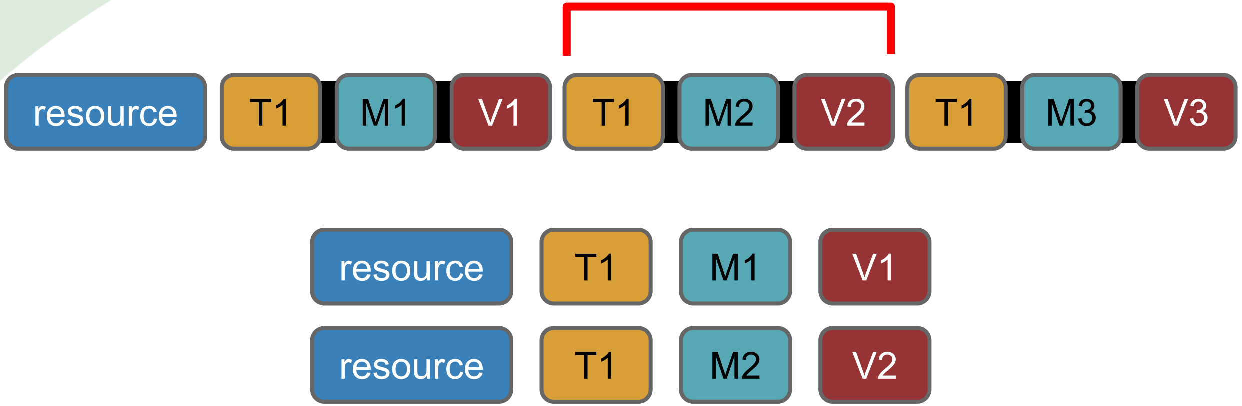
resource T1 M1 V1 T1 M2 V2 T1 M3 V3

```
SELECT * FROM samples
WHERE resource = ‘resource’
AND T = ‘T1’;
```

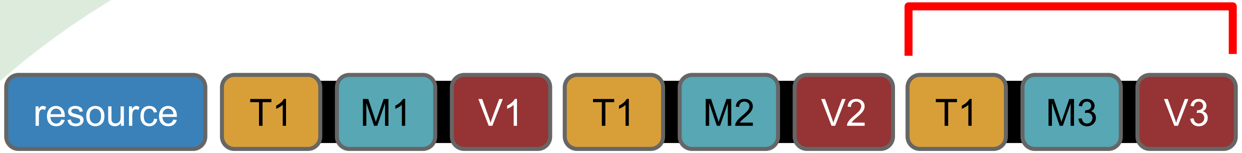
Data model



Data model


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

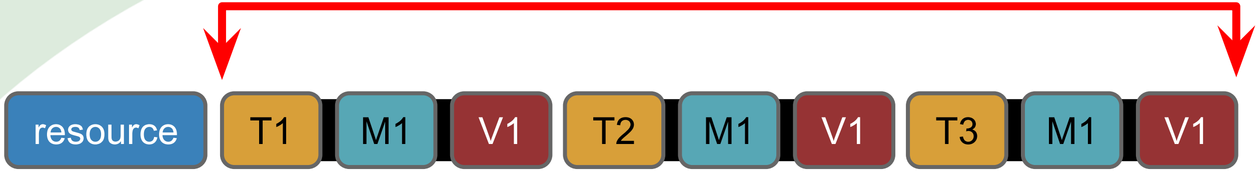
Data model



resource T1 M1 V1 T1 M2 V2 T1 M3 V3

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

Data model



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

Newts

- Standalone time series data-store
- Raw sample storage and retrieval
- Flexible aggregations (computed at read)
  - Rate (counter types)
  - Functions pluggable
  - Arbitrary calculations
- Cassandra-speed

Newts

Fork me on GitHub

- Java API
- REST interface
- Apache licensed
- Github (http://github.com/OpenNMS/newts)

openNMS®

Fin