Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual

van der Veen et al 2012

Luca Rossetto
Introduction

- Large increase in sensor usage
- Leads to more sensor data
- Data has to be stored and processed
- How to do that best?
Sensor Data

- Generated by small microcontroller based devices
- Usually raw data from one or many different instruments
- Measurements taken periodically or on certain events
- Generates lots of small data packets
- Usually composed of
  - Instrument/Sensor/Node Id
  - Timestamp
  - Measured value
The Goal

- Compare three database systems on relative performance with sensor data, both natively and virtualized (using XenServer)
  - PostgreSQL
  - Cassandra
  - MongoDB
- Give suggestions which system to use for which kind of application
PostgreSQL

- SQL database
- Mostly SQL 2008 compliant
- Maintained by an open source community since 1997
- Primarily designed to run on one server
- Written in C
Apache Cassandra

- NoSQL database
- Structured key-value Store
- Designed to scale over very large amount of commodity servers
- Open Source since 2008
- Originally developed by Facebook, now maintained by Apache
- Written in Java
- Currently used by Twitter, Digg and Reddit
MongoDB

- NoSQL database
- Key-value store for BSON objects
- Designed for one or few commodity servers
- Developed since 2007, public since 2009
- Written in C++
The Setup

I’m up to 16 sensor nodes

24 CPU cores @ 2.2GHz
64GB RAM
2 x 2TB HDD
Scenarios

- Single client, single operation
  - One sensor node writes data whenever a new measurement is taken
  - One analyst accesses single entries

- Single client, multiple operations
  - One sensor node writes data in batches
  - One analyst accesses many data items at once

- Multiple clients, single operation
- Multiple clients, multiple operations
Use of indexes

- MongoDB and PostgreSQL can be used with or without indexes
- Cassandra automatically generates indexes
- “...The speed difference between writes with and without indexes is very low...”
- Read access with indexes is significantly higher
- Therefore, indexes are used in all three database systems
Single client, single operation

Write

Read
Single client, multiple operations

Write

Read
Multiple clients, single operation

Write

Read

29.11.2012
Luca Rossetto
Multiple clients, multiple operations

Write

Read

Luca Rosetto 29.11.2012
Recommendations

- Use Cassandra for large scale critical sensor applications
- Use MongoDB for small to medium sized non-critical sensor applications
- Use PostgreSQL for applications which require flexible or complex querying and a high read performance
Some comments

- Test setup does not accurately represent actual use case
  - No simulation of real network
  - Too few nodes
  - Inappropriate assumptions concerning node capabilities
- Too few measurements
- Only one server used
  - Cloud application planned as future work
- Results directly reflect properties of tested database systems
SELECT questions FROM audience

Thank you for your attention
Questions?