The problem of clock synchronization in cloud storage system

DDN – PARIS
2017, Jan. 30th,

Alexey Romanenko

Clocks in real life

- **Non-atomic clocks**
  - Not precise
  - Depends on clock quality, weather conditions, power stability, etc
  - Quartz is better than mechanical ones
  - Drift can be in order of seconds per days
- **Atomic clocks**
  - Very precise
  - Used as primary standards to control:
    - Wave frequency of TV broadcast
    - In GPS
  - It uses the microwave signal that electrons in atoms emit when they change energy levels
  - Accuracy of 10^{-9} seconds per day

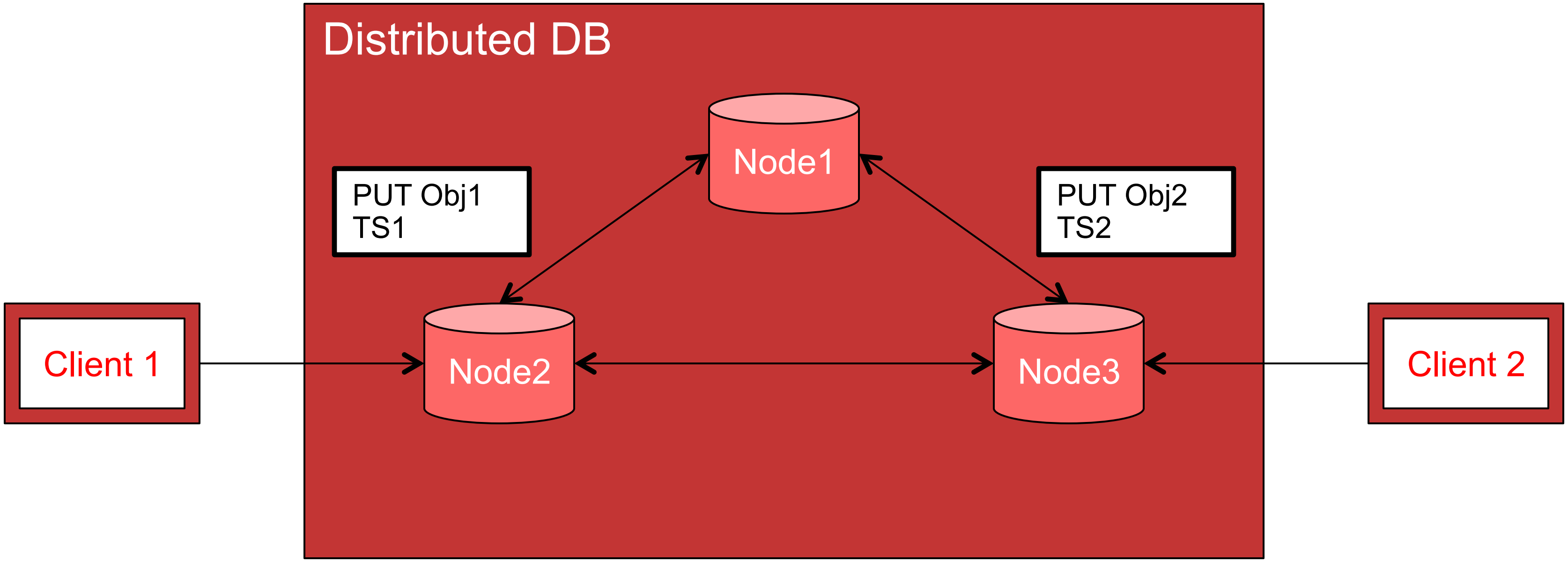
Clocks in computer

- **How it works in two words**
  - Quartz crystal generate oscillation with some frequency
  - Every oscillations are counted in register
  - Interruption is generated after several oscillations **clock tick**
  - Computer clock is incremented on each tick
- **Clock drift**
  - Not perfectly tuned crystal
  - External factors, like temperature or humidity, might have an influence
  - Computer clock differs from real time clock
- **Clock skew**
  - Two crystals are not identical
  - Two computers with different crystals have different internal time

Why should we care about clock sync?

- **Not a big deal for single machine, but…**
- **In distributed environment it might be very important**
  - Ordering of concurrent requests in distributed systems
    - Example: two clients send requests to update data on different cluster nodes almost in the same time (microseconds difference)
  - Transactions in distributed databases
  - Data replication between two geo-distributed sites
  - Time synchronization between senders and receivers.

Concurrent requests in distributed systems



Here is a problem:
TS1 > TS2 or **TS1 < TS2**
???

Distributed Databases, HBase

- **Column-Oriented data storage (Hadoop Database)**
  - Based on Google BigTable architecture
- **Horizontal scalability**
  - Automatic sharding
- **Write and read operation are strongly consistent**
- **Automatic fail-over**
- **Support random real time CRUD operations**
- **Distributed system designed for large tables**
  - Billions of rows and millions of columns
- **Works on commodity hardware cluster**
- **Open-source, written in Java, Apache project**
- **NoSQL**
  - No SQL-access
  - Doesn’t provide relation model (only limited part)

HBase architecture

- **<u>Table</u> is split into <u>regions</u>**
- **<u>Region</u> is group of <u>rows</u> that stored together**
  - Unit of **sharding**
- **<u>Region server</u> is daemon which is responsible for one or several regions**
  - One region is linked to only one region server
- **<u>Master server</u> (HMaster) is daemon which manage all region servers**

HBase Data Model

- **Data is stored in table**
- **Tables contains rows**
  - Access to row by unique key
    - Key – byte array
    - Everything can be a key
  - Rows are sorted in lexicographical order of keys
- **Rows are grouped by columns in column families**
- **Data values are stored in cells**
  - Access to cell by row : column-family : column
  - Values are stored as byte array

HBase Timestamps

- **Values in columns have versions**
  - Hbase keeps several versions of values
  - New dimension for data
  - Timestamp
    - Set implicitly by RegionServer during write operation
    - Can be set explicitly by client
  - Versions are stored in descending order of ts
    - Last written value will be read at first
- **Value = Table + RowKey + Family + Column + Timestamp**

Cloud metadata in HBase

Row Key	Timestamp	CF: "Core Data"		CF: "Meta Data"	
		UserID	ObjectID	Size	Date
object1	t1	1234	aaa111	1234	123401
	t2	1234	aaa112	1234	123410
	t3	1234	aaa113	1234	123421
object2	t1	1221	ccc331	2345	123765
	t2	1221	ccc332	2345	123765

Node1

Node2

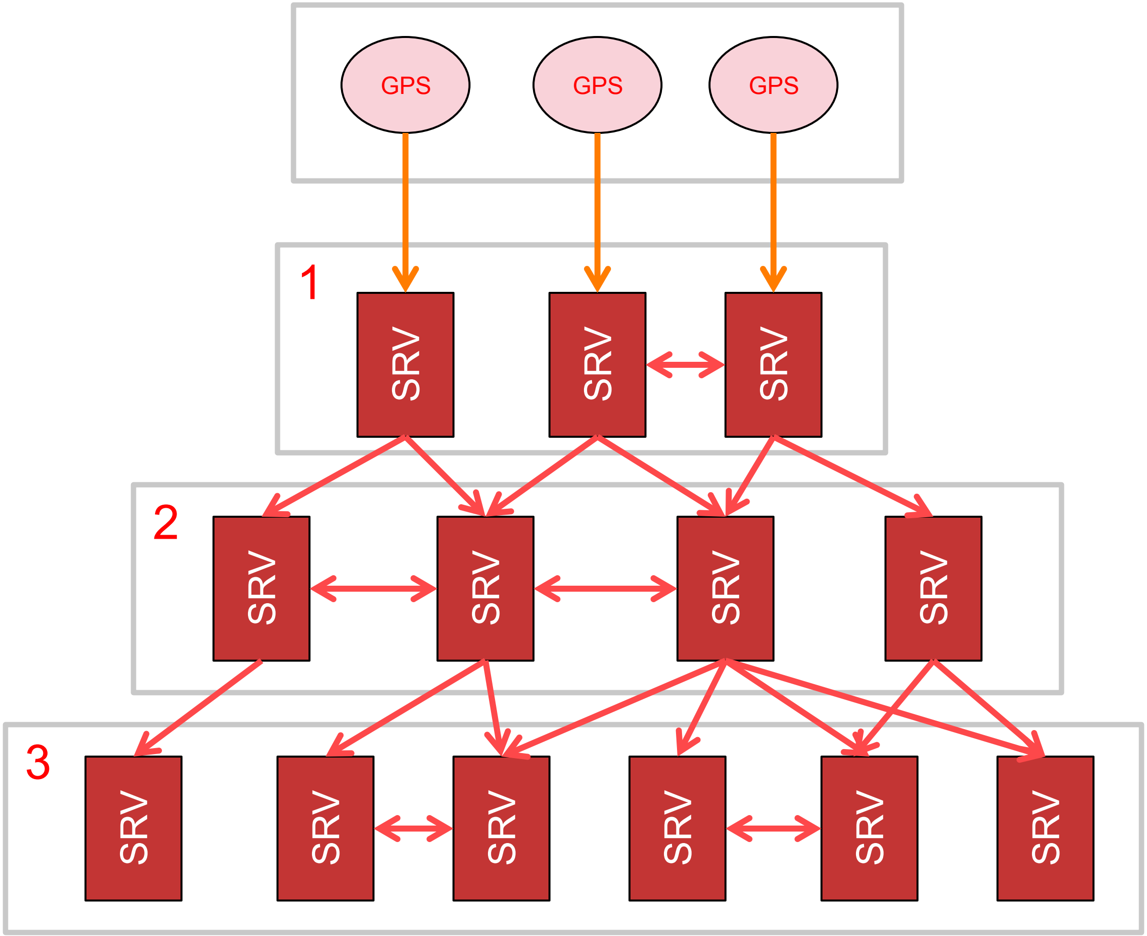
Possible solutions

- **Global Positioning System**
  - The accuracy of GPS time signals is ±10 ns
  - Based on atomic clocks
  - Second after the atomic clocks
- **Network Time Protocol (NTP)**
  - The state of the art in distributed time synchronization protocols for unreliable networks.
  - The order of a few milliseconds over the public Internet, and to sub-millisecond levels over local area networks.
- **Precision Time Protocol (PTP)**
  - Designed to fill a niche between NTP and GPS
- **Logical clock**
  - Mechanism for capturing chronological and causal relationships in a distributed system

NTP – Network Time Protocol

- **Network Time Protocol (NTP)**
  - Internet protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks.
  - Since 1985, designed by David L. Mills of the University of Delaware
- **NTP features**
  - NTP needs some reference clock that defines the true time to operate
    - NTP uses UTC
    - Universal Time Coordinated is an official standard for the current time
  - NTP is a fault-tolerant protocol and scalable
  - NTP can select the best candidates to build its estimate of the current time.
- **Accuracy**
  - About one millisecond accuracy in local area networks under ideal conditions
  - Tens of milliseconds over the public Internet
  - 100 milliseconds or more with asymmetric routes and network congestion

NP architecture

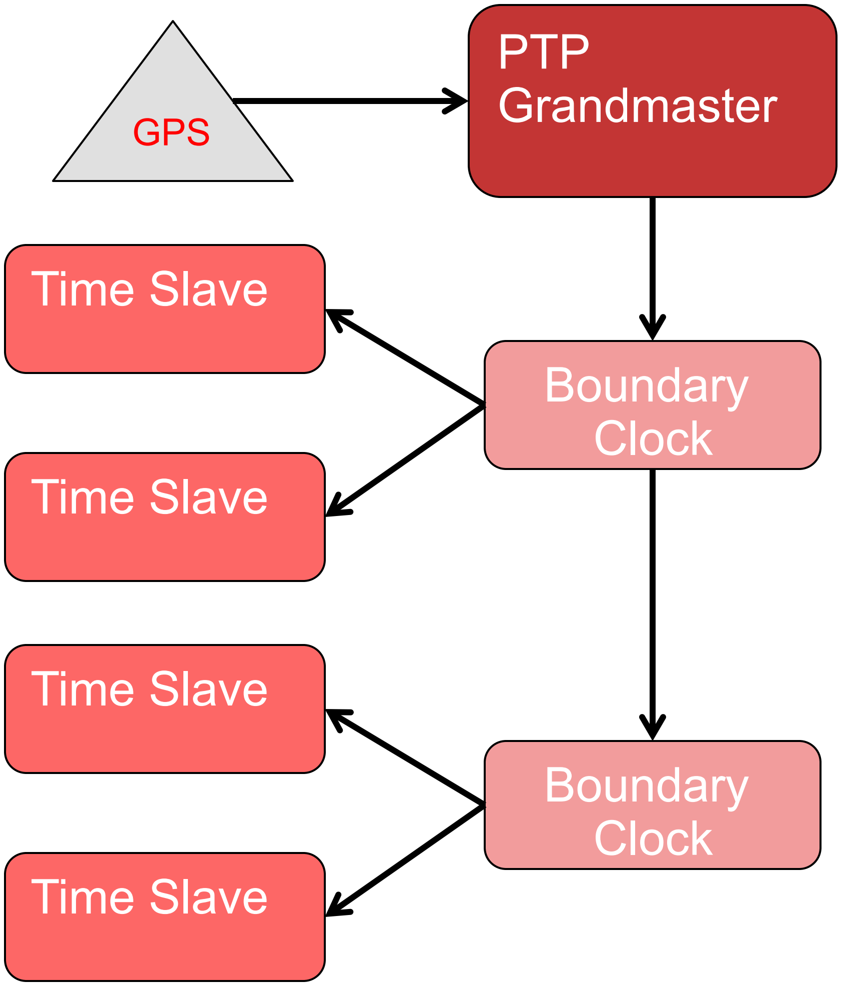


- **Stratum 0**: high-precision timekeeping devices (GPS, atomic, radio clocks)
- **Stratum 1**: synchronized to within a few microseconds to Strata 0
- **Stratum 2**: query several Stratum 1 servers

PTP - Precision Time Protocol

- **PTP**
  - PTP is used to synchronize clocks in a computer network with high accuracy
  - Designed to fill a niche between NTP and GPS
  - When used in conjunction with hardware support, PTP is capable of sub-microsecond accuracy

PTP architecture



- Clocks synchronization are organized in a master-slave hierarchy
- Slaves are synchronized to their masters
- Best master clock (BMC) algorithm, which runs on every clock.
  - One port – master or slave (ordinary clock - OC)
  - Two ports - master and slave (boundary clock - BC)
- Master can be slaves for their own masters
- The top-level master is called the **grandmaster clock**
  - synchronized by using **GPS**

PTP vs. NTP

- **NTP pros**
  - Easier to implement
  - More cheaper, no special switches are required
- **PTP pros**
  - Much better accuracy then with NTP
  - One of the main advantages is hardware support present in various network interface controllers (NIC) and network switches.
    - PTP accounts for delays in message transfer which improves accuracy
    - Possible to use non-PTP hardware but not recommended

Logical clock

- Logical clock was proposed in 1978 by Lamport as a way of timestamping and ordering events in a distributed system.
- Doesn’t depend on physical time
- Allows global ordering on events from different processes in distributed system
- In logical clock systems each process has two data structures:
  - **logical local time** - used by the process to mark its own events
  - **logical global time** - local information about global time
- **Hybrid Logical Clocks** is based on idea of combining logical clock and physical time
  - Substitutable for physical time (NTP clocks) in any application.
  - Resilient and monotonic and can tolerate NTP kinks.
  - Can be used to return a consistent snapshot at any given T
  - Useful as a timestamping mechanism in distributed databases

Some conclusions

- **Clock synchronization** is very important question in distributed systems
- No silver bullet (as usually)
- The choice of the algorithms/protocols depends on application needs and requirements
  - **NTP** – easy to use, good in cases when accuracy is not very important
  - **PTP** – requires additional hardware and support by NIC, very high time accuracy
  - **HLC** – requires application changes, no need hardware support

Used URLs

- http://www.ntp.org/ntpfaq/NTP-s-def.htm
- https://en.wikipedia.org/wiki/Network_Time_Protocol
- https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/ch-Configuring_PTP_Using_ptp4l.html
- http://muratbuffalo.blogspot.fr/2014/07/hybrid-logical-clocks.html

Questions?