Multi-threading Architecture of MySQL Cluster

Mikael Ronström, Senior MySQL Architect @ Oracle

Purpose of this presentation

- See a real-world challenge and how one can solve it
- Describe a scalable and modular architecture

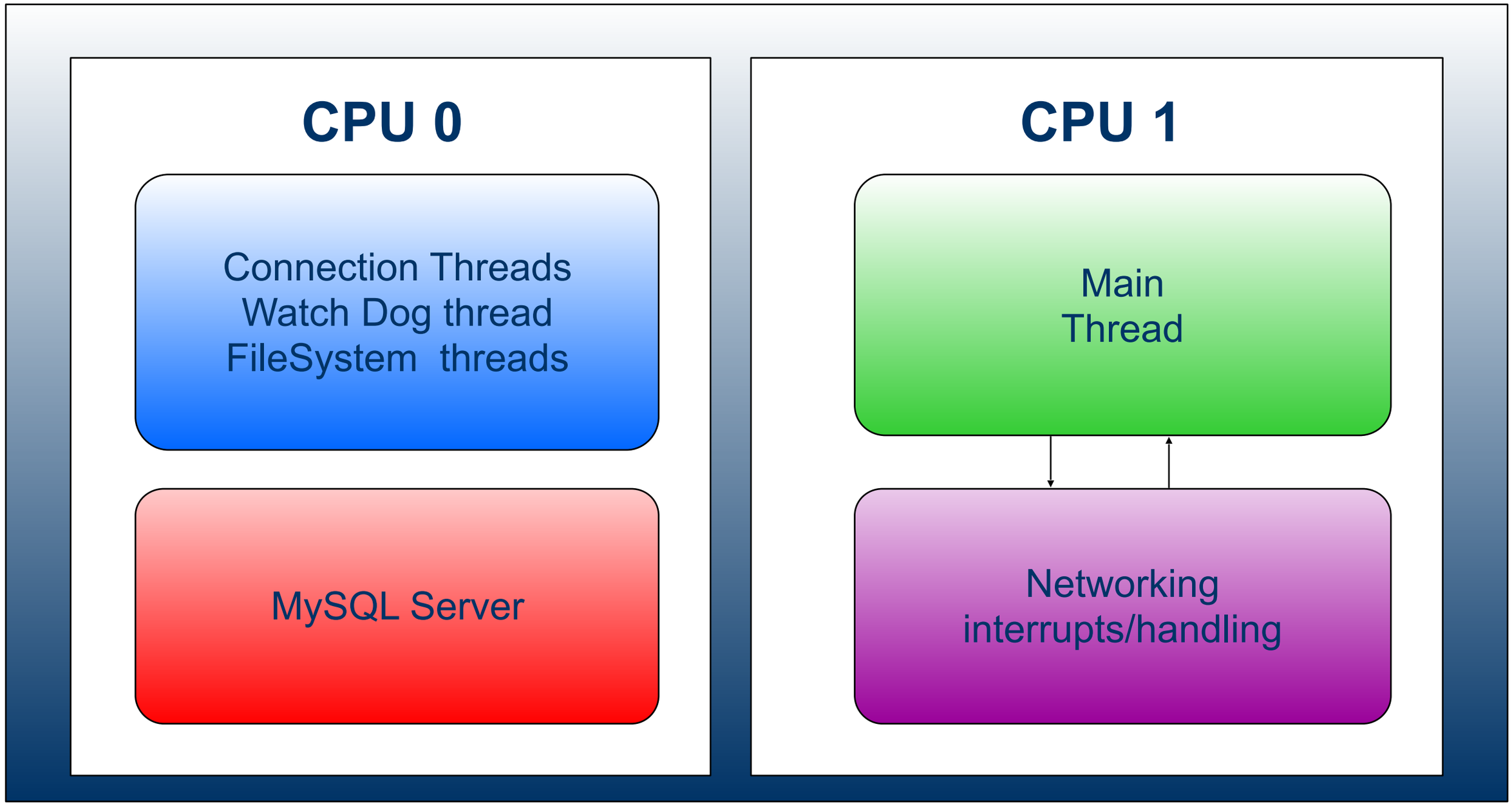
MySQL Cluster originated 1990s

- Single-threaded OS-like environment
- File System made asynchronous by separating into their own set of threads
- Threads to setup networking connections
- Watchdog thread to check for process being alive

MySQL Cluster originated in AXE architecture

- Code divided into blocks (== software modules)
- Blocks only communicate through signals (== messages)
- Blocks do not share data
- => Scalable architecture, modular architecture
- AXE was designed from the same ideas as HW (blocks == ICs, signals == circuits, well defined interfaces)

MySQL Cluster
Original Threading Architecture



Multi-core CPU revolution in 2000s

- Xeon 2006: Dual-core CPUs
- Xeon 2007: Quad-core CPUs
- Xeon 2008: 6-core CPUs
- Xeon 2010: 8-core CPUs
- Xeon 2011: 10-core CPUs
- Xeon 2013: 12-core CPUs
- Xeon 2014: 15-core CPUs
- In addition Hyperthreading and up to 8 sockets per off-the-shelf server
- => Up to 240 CPU threads per machine in 2014 from 4 in 2006

How do we scale our SW 60x in 8 years?

- MySQL Cluster 6.3, 2008, can make use of 2 CPU threads per data node
- MySQL Cluster 7.0, 2009, use up to 8 CPU threads
- MySQL Cluster 7.2, 2012, use up to 32 CPU threads
- MySQL Cluster 7.3, 2013 use up to 64 CPU threads

MySQL Server Scaling

- MySQL Server 5.1, GA 2005, Scaled to 4 CPU threads
- MySQL Server 5.4, Preview 2009 (not released), Scaled to 16 CPU threads
- MySQL Server 5.5, GA 2010, Scaled to 32 CPU threads
- MySQL Server 5.6, GA 2013, Scaled to 60 CPU threads
- MySQL Server 5.7 Preview 2014, scales up to 80 CPU threads for use in MySQL Cluster

Combination of MySQL Server + Data Node

- MySQL 5.1 scaled to only 1-2 CPUs using MySQL Cluster. So MySQL Server + Data node on same machine runs optimally on 2 CPU threads (== 2006)

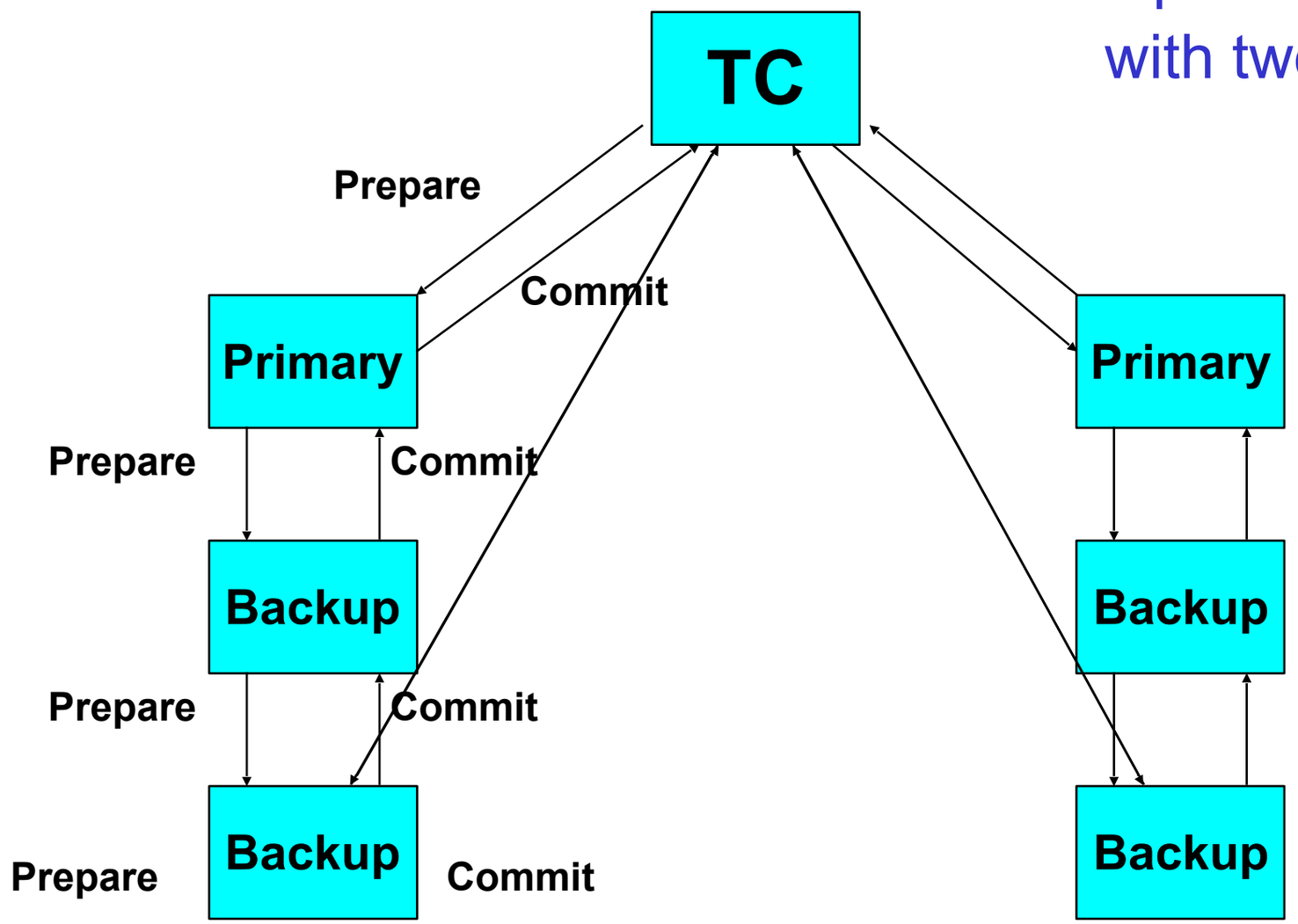
- MySQL Cluster 7.3 based on MySQL 5.6 can make use of 60 CPU threads for MySQL Server and 60 CPU threads for Data Node => 120 CPU threads

- So we achieved a 60x scaling of our SW from 2008 to 2013

Synchronous Replication:

Low failover time

messages = 2 x operations x (replicas + 1)

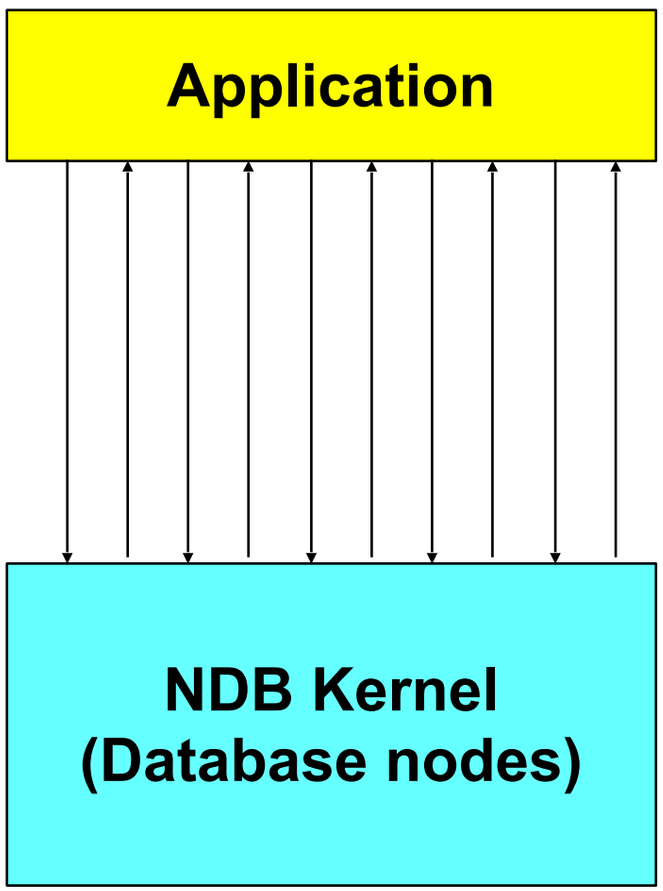


Example showing transaction with two operations using three replicas

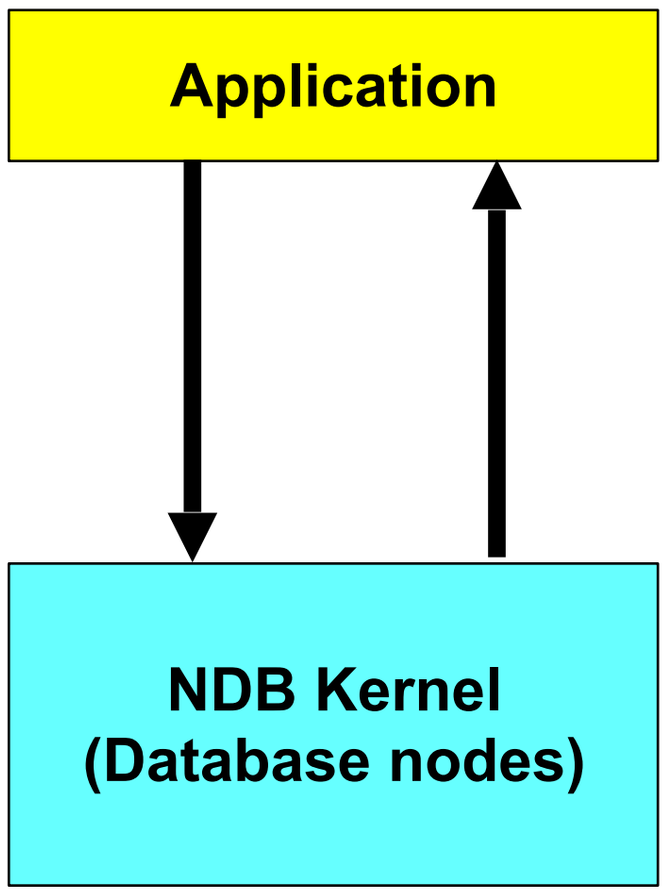
1. Prepare F1
2. Commit F1
1. Prepare F2
2. Commit F2

Looking at performance

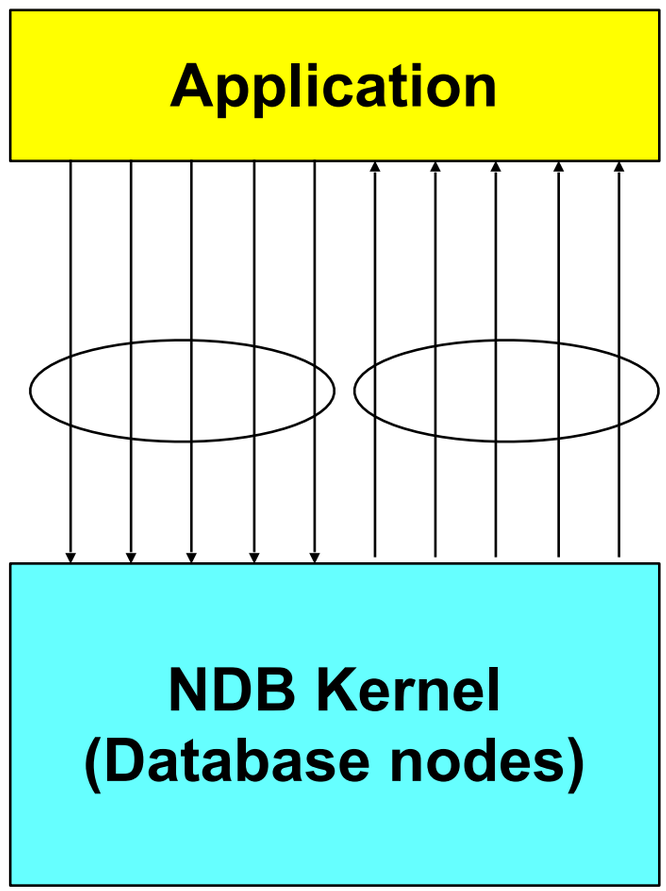
Five synchronous insert transactions (10 x TCP/IP time)



Five inserts in one synchronous transaction (2 x TCP/IP time)



Five asynchronous insert transactions (2 x TCP/IP time)



Characteristics of MySQL Cluster (1)

- Access through MySQL Servers as usual
- Transaction Support
- High Availability
- Synchronous Replication
- Automatic synch at node restart
- Distributed Main Memory Database
- Disk Data-support for non-indexed data
- Cluster Recovery from Disk Checkpoint
- Scalable Performance

Characteristics of MySQL Cluster(2)

- Tables are automatically partitioned over the Data Nodes in the cluster
- Separate configuration file for the cluster
- Separate Management Server(s) for the cluster
- Separate Management Client for the cluster
- Automatic discovery of node failures through heartbeat mechanisms
- Row locks

Characteristics of MySQL Cluster(3)

- Don’t support LOCK/UNLOCK TABLES
- Automatic Distribution of Table changes to all MySQL Servers in the cluster
- Automatisk Distribution of new Databases in the cluster
- No automatic distribution of Views, Stored Procedures and Triggers

Characteristics of MySQL Cluster(4)

- The cluster needs to be in a protected network without direct connection to internet
- MySQL Replication works between clusters (including Active-Active replication)
- WHERE-condition can be pushed down to the Data Nodes
- Certain types of join operations can be pushed down to the Data Nodes
- Updates from one MySQL Server is seen by all MySQL Servers immediately

On-line Characteristics in MySQL Cluster

- On-line Backup
- On-line Add/Drop Index
- On-line Add Column
- On-line Add Node(s)

Indexing in MySQL Cluster

- Primary Key always a distributed hash index
- Unique index separate table using distributed hash index
- Ordered index (T-Tree) locally in each partition of the table
- Definition of primary key and unique key automatically also defines an ordered index unless USING HASH is used in the index definition

Fixed Row Format in MySQL Cluster

- Hash Index overhead ~15 bytes per row
- Ordnat Index overhead ~10 byte per row
- Header Info overhead ~20 bytes per row

(Transaktionsinfo, Index info, NULL bitar, Checksumma, timestamp)

- All fields aligned on 4 bytes

Dynamic Row Format in MySQL Cluster

- VARCHAR stored as 2 bytes length + data
- Only for Main memory data
- Supports Dynamic fields (fields not specifically there is NULL, makes on-line Add Column possible)