MySQL Cluster

very big and fast databases on commodity hardware

PHP Conference | 11/09/2005 | Alex Aulbach

The speaker

- Alexander Aulbach
  - 37 years old
  - employee of Mayflower GmbH
  - web application development since 9 years
  - database design/development since 7 years

Mayflower GmbH

- founded 1997
- Munich and Würzburg, Germany
- Various projects for european companies like
    - Vaillant
    - Telefónica
    - HypoVereinsbank
- Heavy growth
- ThinkPHP bundles the PHP/LAMP activities of MAYFLOWER
    - Core Developer PHP and Apache
    - Lot of activities in the OpenSource community
        - PHPProjekt
        - lighttpd
        - PHP support

Agenda

- 17 slides
- Cluster: why and for whom is it suitable?
- Terminology
- Features
- How it works (explained hopefully easy)
- NDB storage engine
- Comparison to Oracle RAC
- Hardware requirements
- Limitations
- Practical show

MySQL Cluster wants to go far beyond...

Some look at MySQL's shared-nothing clustering as a viable enterprise-class feature. [Oracle CEO] Larry Ellison has said he'd be happy if 10 percent of customers adopted clustering. MySQL's [vice president of marketing] Zak Urlocker has said he's happy to pick up the other 90 percent.

(http://www.eweek.com/article2/0,1895,1605776,00.asp
Oracle VP: MySQL Cluster Not a Threat)

Key features

- HA (High Availability)
  through parallel server architecture availability of 99,999% (five-nines), i.e. Non-availability less than 5 minutes per year
- Dynamically scalable
  extending the cluster through more commodity boxes: NDB scales nearly linear
- High Performance
  100,000 transactions/s at less than 5ms response time with 4 CPUs. 380,000 write transactions/s, 1,5 mio. Read transactions/s (128 byte records) at 72 CPUs
- Low cost
  use cheap hardware („commodity“)
- Key: speed and availability at low hardware costs

Target groups

- Already existing users of MySQL
  Throughput of a business critical application is not enough.
- Telecommunication companies
  Replace commercial or self-written solutions
- Government
  cheap solutions for government and communes who want to use OpenSource Software
- Companies
  Every organisation that needs high availability for reducing the costs of breakdown
- Developers
  without paying, running a cluster and to see what happens if you cut the power.

Terminology

- NDB: network database
- Nodes: a node simply is a server process
- Differentiation between
  - MGM Node (management node),
  - Data Node (which saves and replicates, i.e. the core of the cluster) and
  - SQL Node (mysqld)
- Cluster: in our case the data nodes. Or, to explain it differently:
- Cluster: physical RAM of the NDB storage engine in MySQL.
- MySQL Cluster: a combination of MySQL and NDB storage engine

Features

- transactions
- synchronous replication
- auto-sync at startup of a NDB node
- restore of a checkpoint (if cluster cache is on)
- online backup (without shutting down before)
- reconstruction of changes at one row
- there are two indices (explicit hash and T-tree sort)
- index creation at runtime
- software update at runtime

How does the MySQL Cluster work?

- It has nothing to do with Replication!!
  (of course you can replicate with a cluster, it makes sense if you want to replicate the data on “slow upstreams” to an upstream server)
- nearly the same principle like RAID

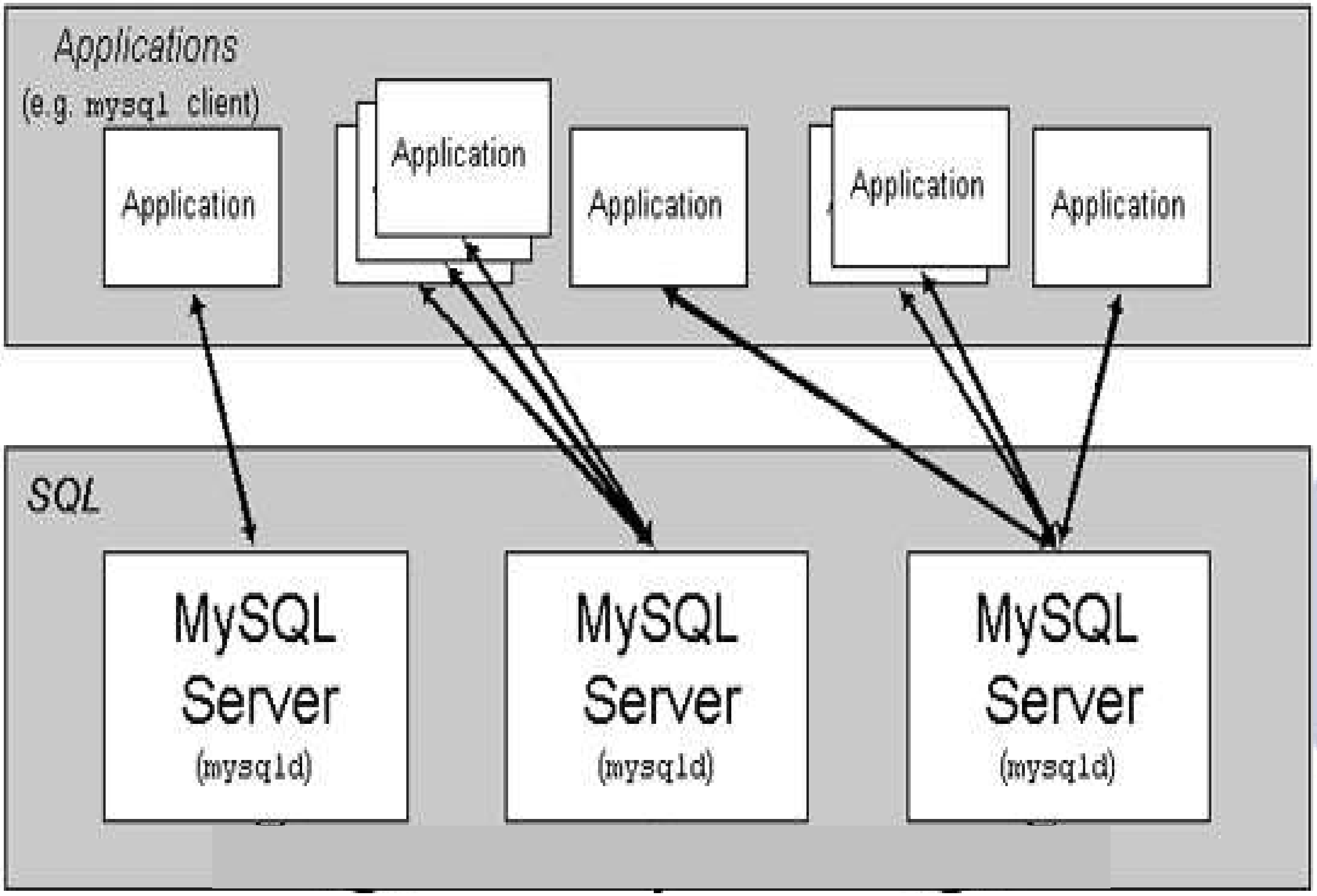
To remember:

- Raid 0: Striping (Concatenation)
  several physical HDDs will be connected to one large HDD
- Raid 1: Mirroring
  same content will be distributed to several HDDs so that the shutdown of one HDD will have no cause to the whole array

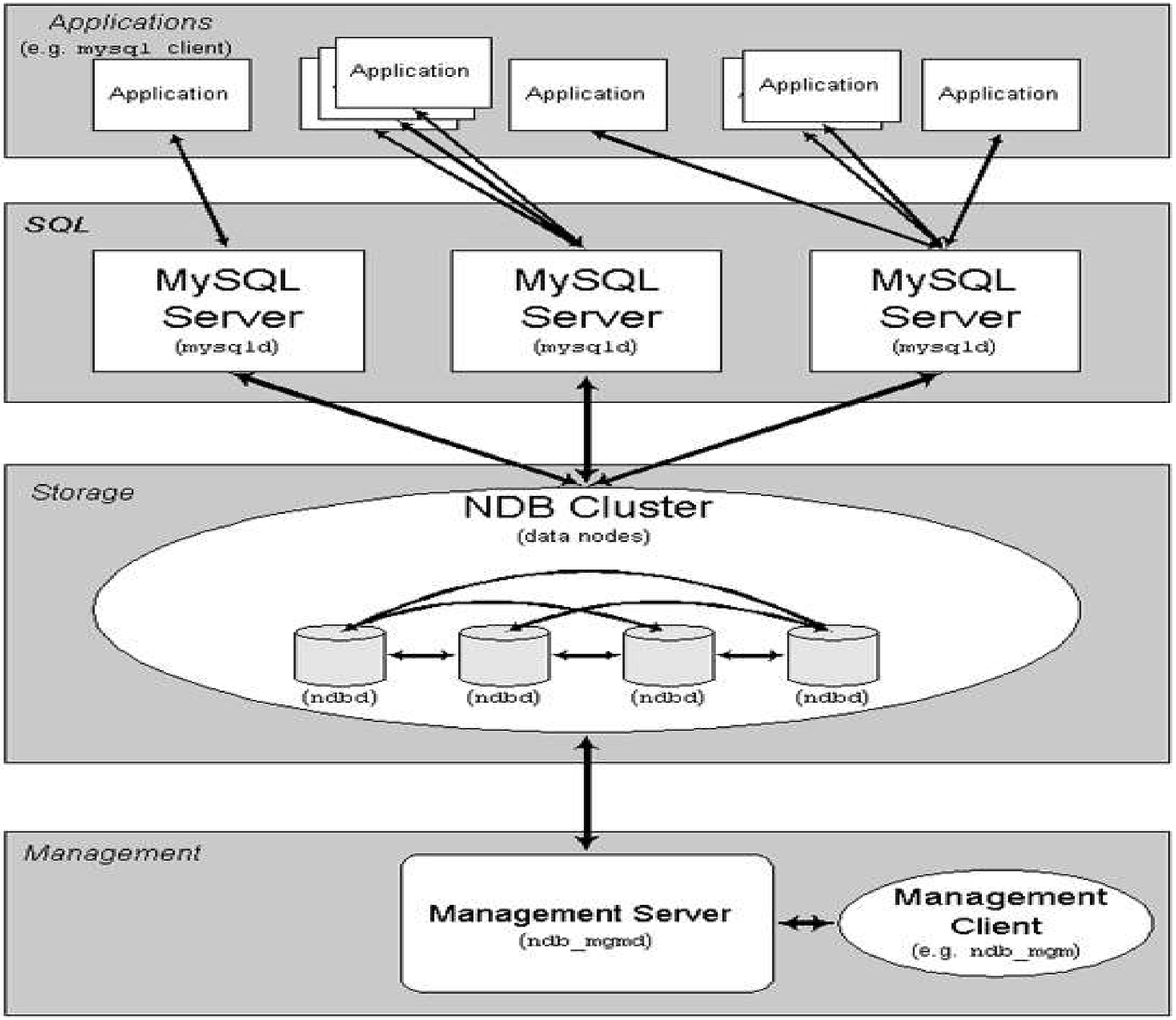
Analogy: MySQL-Cluster to RAID

- MySQL Cluster works in the same way
- replace “HDD” by “data node” (or by “CPU process”)
- replace “RAID” by „NDB Cluster“
- RAID 0 or RAID 1 won't be differentiated here, the coherence will alway be obvious if several nodes will run on different servers.
- Finally, there's a “superboss” (MGMD) which can manage the breakdown of one or more servers and handles the commands to the data nodes.

Conventional database installation



MySQL Cluster



NDB Storage Engine

- Now common at MySQL: just a new storage engine
- The engine is connecting automagically to the cluster (configured by the MGM node)
- Queries to the MySQL server which are directed to a NDB table will be handled by the NDB cluster. The MySQL server is just acting like a proxy
- Possible: NDB cluster as a independant database. MySQL is not neccessary, but useful to have it :-)
- Cluster manages the data nodes completely on its own
- MGM is – after start of the database cluster – not neccessary in theory
- But it handles breakdowns: if a part of the cluster is breaking down or it can't see other party of the cluster (some kind of a watch dog)

What's not possible (yet)?

- Absolutely failsafe – 100% safety on power breakdown of all cluster nodes is not possible
- Not ready yet for large databases (several hundreds of GB) because nearly everything is handled in RAM (should be fixed with MySQL 5.1)

MySQL Cluster vs Oracle RAC?

- Oracle RAC needs sufficient hardware
- Investmenets in SAN (storage area network, very expensive network)
- Dedicated specialists who are able to setup the cluster
- MySQL cluster is not targeted at the very special requirements of some special customers
- MySQL cluster is a product for the masses
- NDB cluster is already there with MySQL-4.1-max! You only have to configure it
- “just” 1,5 years after release: really stable?!
- will be developed further
- Customers don't often want to invest more than neccessary
- We can advise the MySQL cluster for 90% of all cases

Hardware requirements (1)

- Currently everything will be stored in RAM. You need at least as much RAM as the size of the database:

  *(SizeofDatabase * NumberOfReplicas * 1.1) / NumberOfDataNodes*

  It's hard to estimate exact numbers (ie. primary key will be saved additionally as hash etc.), so you should calculate a bit more RAM.
- To have a speedy cluster you should have server like 2 x Xeon, 16 GB Ram, 4 x 73 GB Raid, Gigabit Ethernet per server
- To increase the throughput you can package more RAM, more CPUs or more commodity hardware. (2, 4, 8, 16 etc.)

Hardware requirements (2)

- A good cluster setup begins with 4 data nodes
- You need at least 2 data nodes (otherwise setting up a cluster would be nonsense) plus a third server for the management node (which can run on another server because it doesn't need that much CPU)
- So you need at least three (3) servers to setup a failsafe cluster!
- You need at least 100 mbit network connection. It's better to use Gbit network, SCI (Scalable Coherent Interconnect) or other high-speed connections.
- The cluster network itself should run on its own separated network connections.

Planning?

- You can't setup the cluster “just in five minutes”, also not for playing around! You should at least calculate ½ to 1 day except if you want to create the playground on just one server which may be senseless
  You need to plan thoroughly!
- The hardware has to harmonise (I/O throughput)
- Check for a backup concept – neccessary? How important is the data?
- What advantage should a cluster have if the power connectivity is not very good?
- The concept of a cluster is speeding up or breaking down with the speed of the network connection.

Further limitations (only most important listed here)

- Every table needs a primary key (i.e. automagically generated)
- FULLTEXT and prefix indices are not yet indexable
- All character sets and collations will be supported with V5.0+
- Spatial extensions (i.e. GIS datatypes) won't be supported
- no partial rollbacks
- A maximum of 128 attributes per table row; attribute name not longer than 31 chars; maximum length of database name + table name 122 chars; maximum 1792 tables per cluster DB
- Maximum size of a table row ist 8 KiB (without BLOBs)
- No foreign keys
- No query caching (of course)
- Tables will be saved only at fixed-length

Other limitations

- Because everything will be handled sequentially, searches for spans (i.e. with BETWEEN) are slow
- The query optimizer is not yet working, because “records in range” does not work yet. You could workaroung by using “USE/FORCE INDEX”
- All NDB nodes have to have the same architecture (BIG/LITTLE endian!)
- Cluster has to be restarted on every change of the nodes (online adding/dropping is not working)

Live usage of Cluster

- Test-Setup
  - 3 virtual linux servers (VM-Ware)
  - One as a MGM node and two as data node plus MySQL node
- Configuration files
- Startup of the services (MGMD, data nodes, databases)
- Shutdown and startup again of one data node
- Radically breakdown of a node and re-introduction into the cluster
- No performance tests (don't work in this special configuration because VMWare uses the same underlying hardware physically)
- If there's time: integration of a fourth server as it's own MySQL cluster (shutdown of the other two)
- „play around“

Consulting

- If you need consulting, we can help you!
-
- Cluster training
- How to setup the cluster
- How it fits into your organisation
-
- Contact info@mayflower.de for more details

think php

by MAYFLOWER

Thank you for your attention

Alex Aulbach

Mayflower GmbH
Pleichertorstr. 2
97070 Würzburg

+49 (931) 35 9 65 - 23

aulbach@mayflower.de