Analýza clusterového řešení MySQL databáze

1. prostudování možností MySql Clusteru
2. obeznámení se se stávajícím řešením replikovaných databází (radius)
3. navrhnout nové řešení pro zmíněnou databázi
    - MySql cluster s NDB úložištěm
    - Fibre Chanell - RAW přístup / clusterový filesystém
4. Replikace Master-Master, šifrování

Roman Kuneš

A08N0038P

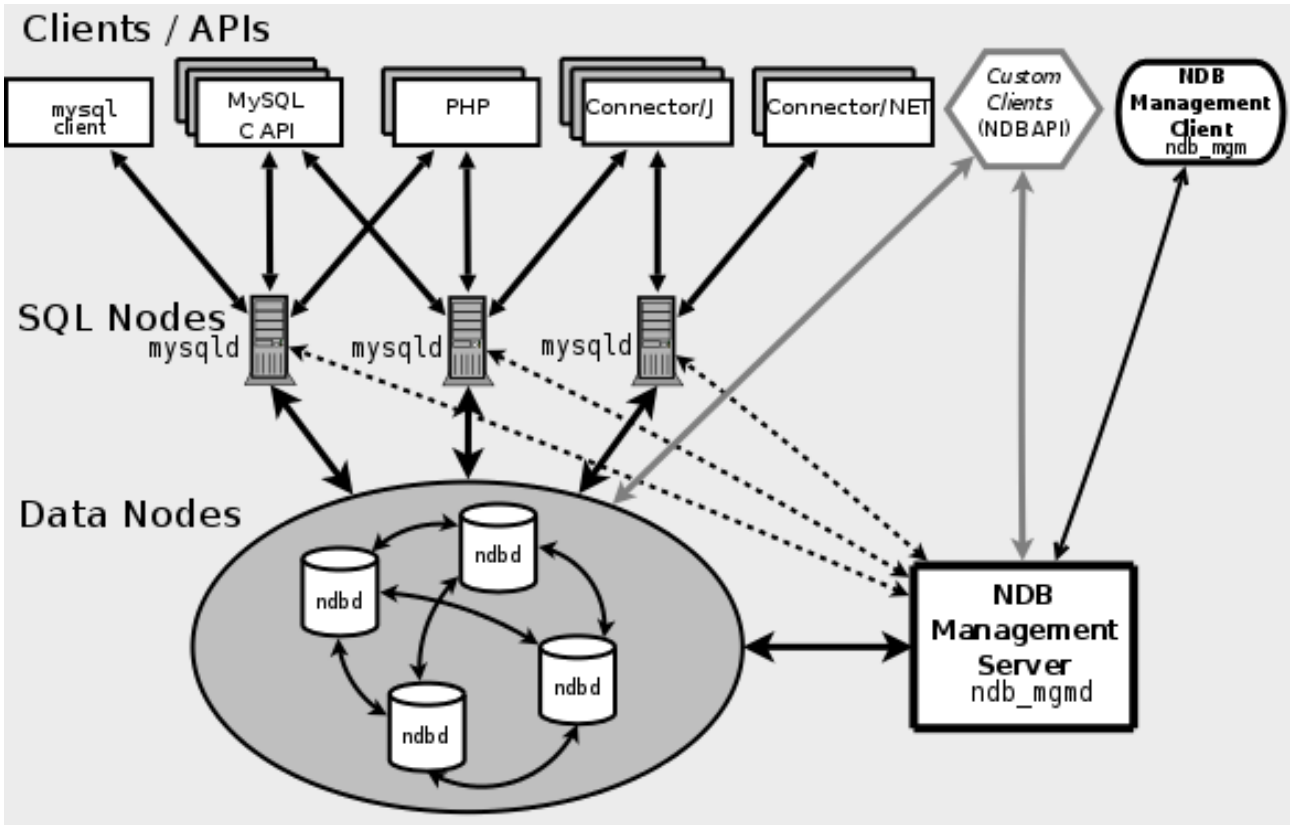
roman85@students.zcu.cz

Obsah

1 MySQL Cluster

Cluster je skupina počítačů spolupracujících na co nejrychlejším odbavování požadavků. Poskytuje mnohem vyšší dostupnost a odolnost vůči chybám oproti jedinému serveru. Každý server v clusteru plní speciální činnost.

MySQL cluster byl navržen za účelem zpracovávání velkého množství real-time dotazů, s vlastnostmi redundantního vyvažování zátěže.



Ilustrace 1: Topologie cluster řešení

1.1 Uzly (nodes)

V kontextu s MySql clusterem se fyzický počítač nazývá jako hostitel a role, kterou zastává, je uzel. Používají se tři druhy uzlů.

1.1.1 Uzel Sql (Sql nodes)

Jedná se o server zpracovávající data, která dostává od klientů (sql dotazy), nebo data vrácená z datových uzlů. Ničím by se nelišil od ostatních sql serverů, kdyby jako úložiště nepoužíval NDB. NDB slouží k fyzickému uložení dat na jiném druhu uzlů.

1.1.2 Datový uzel (Data nodes)

Jedná se o server, jež zajišťuje ukládání dat v paměti a periodický zápis změn na disk. NDB úložiště primárně používá k ukládání paměť ram. Datové uzly se taktéž starají o redundantní ukládání informací. Maximální počet datových uzlů je 48. *"The maximum number of metadata objects in MySQL Cluster is 20320. This limit is hard-coded."[4].*

1.1.3 Uzel pro správu (Management nodes)

Tento uzel je odpovědný za správu clusteru: spouštění a ukončování ostatních druhů uzlů, sledování konfigurace, protokolování změn, zálohování a obnova dat.

1.2 Principy clusteru - další pojmy

1.2.1 Nic není společné (Shared nothing)

Každý uzel běží na samostatném hostiteli, který je vyhrazen jen pro něj a není sdílen jiným uzlem. Technicky je sice možné umístit datový uzel s například sql uzlem na jednoho hostitele, ale v tomto případě by nemělo smysl budovat clusterové řešení.

1.2.2 Klienti clusteru (Clients)

Standardní aplikace, konektory, API programovacích jazyků atd. Tito klienti nevědí, jestli pracují s obyčejným serverem nebo clusterem.

1.2.3 Úložiště

Jedinou volbou úložiště je NDB. Toto úložiště se stará o rozložení zátěže na více nodů, tím zvyšuje výkon a spolehlivost. Více druhů úložišť je uvedeno v příloze.

1.2.4 Transportér

Jedná se o komunikační protokol používaný pro přenos dat uvnitř clusteru. Není však odpovědný za komunikaci mezi klientem a Sql uzlem.

- Sdílená paměť - Běží-li na jednom hostiteli více druhů uzlů, je možné použít jako transportér paměť hostitele. Toto řešení je velmi rychlé, ale z hlediska bezpečnosti a filozofie clusterů není doporučované.
- TCP/IP - Podporuje lokální i vzdálené umístění hostitelů.
- SCI(Scrable Coherent Interface) - technologie pro navýšení rychlosti komunikace mezi uzly více [6] a [7].

1.2.5 Distribuce dat

Techniky které NDB a MySql cluster používají ke zvýšení odolnosti vůči chybám a dostupnosti:

- Fragmenty - Tabulky jsou rozděleny/seskupeny do částí, jež se nazývají fragmenty. Pro správce je to neviditelný proces.
- Repliky - Naplněné fragmenty jsou distribuovány na více datových uzlů. Těmto kopiím se říká repliky a MySql cluster si jich aktuálně udržuje několik.

1.2.6 Kontrolní body

Distribuovaný systém musí obsahovat mechanizmus, jež bude zajišťovat dohodnutý stav. MySql používá záznam stavu, kterému říká kontrolní bod. Toto opatření zajišťuje konzistenci mezi uzly clusteru. Existují dva druhy kontrolních bodů:

- Lokální kontrolní bod - zaručuje zapsání změn v datech na daném uzlu. Je možné upravovat

frekvenci zápisu.

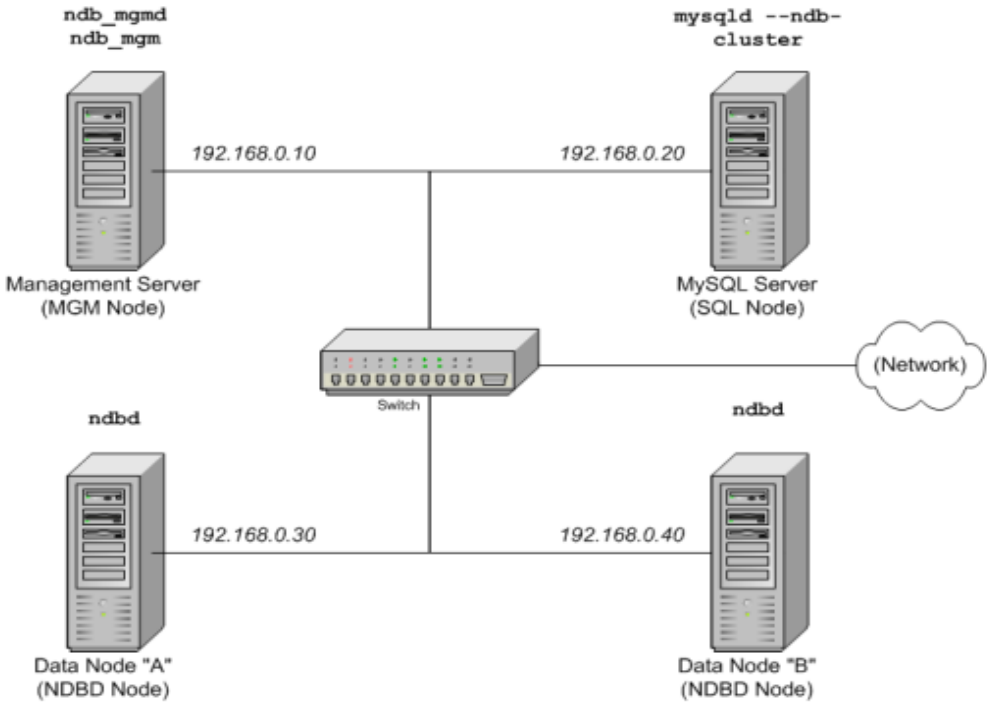
- Globální kontrolní bod - zaručuje konzistenci v celém clusteru.

1.3 Vlastnosti clusteru

- Dostupnost - clustery bez problémů pracují při výpadku jednotlivých nodů (i většího počtu, záleží na topologii) oproti replikaci, kdy jsou při výpadku primárního serveru data na podřízených serverech považována za nekonzistentní. [2]
- Konzistence - jednotlivé datové nody jsou aktualizovány téměř okamžitě. [2]
- Rychlost - data jsou udržována v paměti, tím se zvyšuje rychlost přístupu, který však přináší i omezení na maximální velikost databáze. [2]
- Spolehlivost transakcí - aktualizace dat na všechny nody má za důsledek i spolehlivější transakce. [2]
- Škálování - clusterový přístup umožňuje snadné přidávání dalších nodů, a tím rozmělňovat zátěž.[2]
- Přidávání a odebírání datových uzlů *online* není možné. V takovém případě se musí cluster restartovat.[4]
- Síť - dobrá síťová infrastruktura je předpokladem každého distribuovaného řešení. Velká latence sítě se v tomto případě projeví značně negativně oproti například replikaci. [4]
- Nehodí se na složitější dotazy, kde je nutné pracovat s mnoha tabulkami. *"Range scans - There are query performance issues due to sequential access to the NDB storage engine; it is also relatively more expensive to do many range scans than it is with either MyISAM or InnoDB."* [4]
- Veškeré stroje, které jsou v clusteru použity, musí mít stejnou architekturu. Jedná se o x86 versus PowerPC a big-endian vs. little-endian.
- Není určen pro použití v síti, ve které je propustnost menší než 100Mbps.
- Zabezpečení mezi uzly clusteru není šifrované. Jediný způsob ochrany přenášených dat je ochranou sítě například DMZ.
- Jedná se o novou technologii, což skýtá mnoho problémů při řešení chyb. Špatným použitím se technologie stává kontraproduktivní. Cluster se skládá z mnoha menších částí, z nichž každá by mohla potencionálně ovlivnit celkový výkon. Literatura [1] uvádí že se pro stávající aplikace příliš nehodí: *"A word of warning: NDB Cluster is very "cool" technology and definitely worth some exploration to satisfy your curiosity, but many technical people tend to look for excuses to use it and attempt to apply it to needs for which it's not suitable. In our experience, even after studying it carefully, many people don't really learn what this engine is useful for and how it works until they've installed it and used it for a while. This commonly results in much wasted time, because it is simply not designed as a general-purpose storage engine."*

1.4 Vlastní cluster

Zde bude vysvětleno, jak nakonfigurovat vlastní cluster. Minimální cluster lze postavit se čtyřmi uzly. Každý by se měl nacházet na vlastním hostiteli, mít pevnou IP adresu a operační systém Linux. Balíky pro instalaci mysql-admin, mysql-common, mysql-server-5.1 více o instalaci v [4].



Ilustrace 2: jednoduchý cluster

konfigurace každého datového s sql uzlu probíhá v /etc/my.cnf

```
# Options for mysqld process:
[mysqld]
ndbcluster                      # run NDB storage engine
ndb-connectstring=192.168.0.10  # location of management server

# Options for ndbd process:
[mysql_cluster]
ndb-connectstring=192.168.0.10  # location of management server
```

konfigurace management uzlu v var/lib/mysql-cluster/config.ini

```
# Options affecting ndbd processes on all data nodes:
[ndbd default]
NoOfReplicas=2    # Number of replicas
DataMemory=80M    # How much memory to allocate for data storage
IndexMemory=18M   # How much memory to allocate for index storage
                  # For DataMemory and IndexMemory, we have used the
                  # default values. Since the "world" database takes up
                  # only about 500KB, this should be more than enough for
                  # this example Cluster setup.

# TCP/IP options:
[tcp default]
portnumber=2202   # This the default; however, you can use any
                  # port that is free for all the hosts in the cluster
                  # Note: It is recommended beginning with MySQL 5.0 that
                  # you do not specify the portnumber at all and simply allow
                  # the default value to be used instead

# Management process options:
```

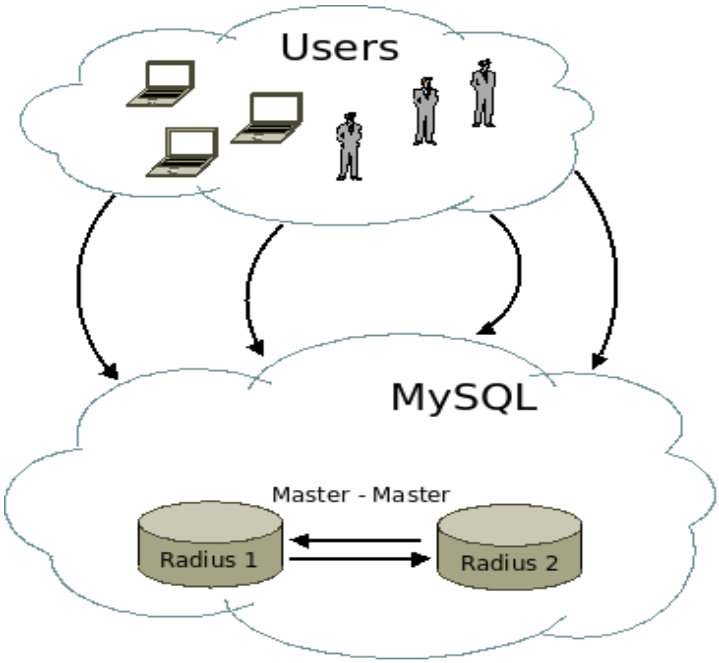
```
[ndb_mgmd]
hostname=192.168.0.10           # Hostname or IP address of MGM node
datadir=/var/lib/mysql-cluster  # Directory for MGM node log files

# Options for data node "A":
[ndbd]
                                # (one [ndbd] section per data node)
hostname=192.168.0.30           # Hostname or IP address
datadir=/usr/local/mysql/data   # Directory for this data node's data files

# Options for data node "B":
[ndbd]
hostname=192.168.0.40           # Hostname or IP address
datadir=/usr/local/mysql/data   # Directory for this data node's data files

# SQL node options:
[mysqld]
hostname=192.168.0.20           # Hostname or IP address
                                # (additional mysqld connections can be
                                # specified for this node for various
                                # purposes such as running ndb_restore)
```

spuštění clusteru

```
#On the management host
shell> ndb_mgmd -f /var/lib/mysql-cluster/config.ini

#On each of the data node hosts
shell> ndbd

#ověření spuštění
shell> ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm> SHOW
Connected to Management Server at: localhost:1186
Cluster Configuration
---------------------
[ndbd(NDB)]     2 node(s)
id=2    @192.168.0.30  (Version: 5.0.72, Nodegroup: 0, Master)
id=3    @192.168.0.40  (Version: 5.0.72, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1    @192.168.0.10  (Version: 5.0.72)

[mysqld(SQL)]   1 node(s)
id=4    @192.168.0.20  (Version: 5.0.72)
```

Nyní je cluster spuštěn a připraven pro nahrání dat. Jedná se o minimální konfiguraci, kterou by bylo třeba pro produkční provoz dále konfigurovat a optimalizovat.

2 Stávající infrastruktura DB



Ilustrace 3: infrastruktura

Stávající infrastruktura obsahuje dva oddělené databázové stroje, jež využívají replikace dat typu Master-Master.

Z nedostačujících výkonnostních důvodů či podivného chování databáze vyvstal požadavek na zajištění větší robustnosti tohoto řešení. Zkoumanou technologií se měl stát MySql cluster, RAW přístup na Fibre Chanel pole či clusterový file systém.

3 Návrh řešení

Po prostudování clusterového MySql řešení vyplývá, že se pro tuto databázi nehodí hned z několika důvodů. Pro minimální fungování clusteru je potřeba nejméně čtyř oddělených serverů, na rozdíl od dostupných dvou. Formát úložiště musí být NDB. To by znamenalo stávající InnoBD do tohoto formátu transformovat. Pravděpodobně se jedná o případ, kdy se stává použití clusteru kontraproduktivní.

3.1 RAW přístup

MySql umožňuje zapisovat data i na nenaformátovaný oddíl. Toto řešení má výhodu ve vynechání mechanizmů operačního systému, čímž se zrychlí přístup na disk. Další výhodou je velmi nízká fragmentace zapisovaných dat.

Nevhodnost spočívá v možnosti nechtěného přepsání oddílů jiným uživatelem či procesem (neexistence filesystému znemožňuje použití přístupových politik).

ukázka nastavení:

```
[mysqld]
innodb_data_home_dir=
innodb_data_file_path=/dev/sdg1:3Gnewraw;
```

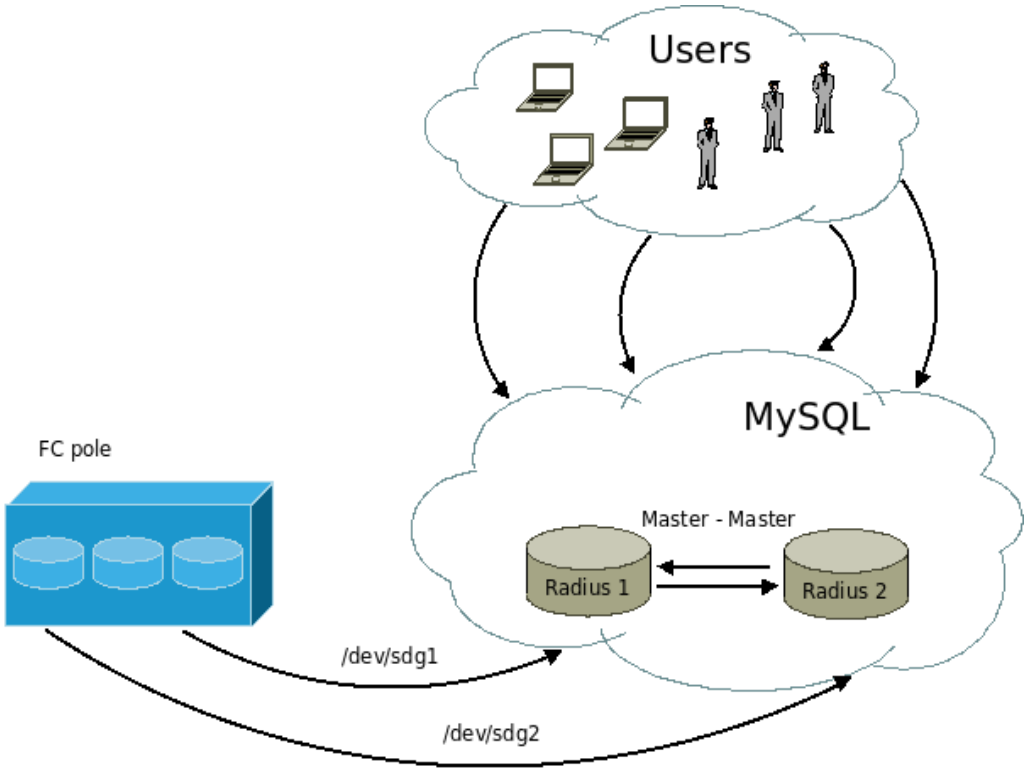
Použití spočívá v nakonfigurování cesty k diskovému oddílu a inicializování výrazem *newraw*.

3.2 Cluster filesystém

Filesystémů tohoto druhu je velké množství. Výčtem: GFS, ActiveScale, PVFS2... V tomto případě nepovažuji výběr filesystému za klíčový a spolehl bych se na volbu administrátora.

3.3 Doporučené řešení

Po zjištění již zmíněných nedostatků MySql clusteru bych se přiklonil k použití replikace. Spíše bych hledal nedostatky ve stávající konfiguraci, než budoval databázový subsystém od začátku. Otázku, jaký bude filesystém na FC poli, bych pokládal za méně podstatnou.



Ilustrace 4: navrhované řešení

4 Závěr

Tímto průzkumem jsem poukázal na nevhodnost použití clusterového řešené pro zmíněnou databázi. Celá technologie se jeví velice slibně, ovšem při hledání konkrétních databázových řešení jsem nenalezl nikoho, kdo by clusterové řešení používal. Naproti tomu velké množství uživatelů využívá replikaci v nejrůznějších modifikací (viz wikipedia[8]).

5 Zdroje

[1] Balling D.J., Zawodny J.: High Performance MySQL, O'Reilly 2008

[2] Schneider R.D.: MySql Oficiální průvodce tvorbou, správou a laděním databází, Grada 2006

[3] http://dev.mysql.com/doc/refman/5.1/en/index.html

[4] http://dev.mysql.com/doc/refman/5.0/en/mysql-cluster.html

[5] http://www.howtoforge.com/mysql_master_master_replication

[6] http://en.wikipedia.org/wiki/Scalable_Coherent_Interconnect

[7] http://ntrg.cs.tcd.ie/undergrad/4ba2.05/group12/index.html

[8] http://upload.wikimedia.org/wikipedia/commons/f/ff/Wikimedia-servers-2008-11-10.svg