NoSQL

Stephane VAROQUI
Field Services - Senior Consultant

Agenda

- SkySQL
- Caching
- Memcached
- Handler_socket
- MySQL et NoSQL

Introduction SkySQL

16 out of 20 most popular websites use MySQL as main front-end Database

1. Google
2. Facebook
3. YouTube
4. Yahoo!
5. Windows Live
6. Baidu
7. Wikipedia
8. Blogger
9. QQ.com
10. Twitter

1. MSN
2. Yahoo! JP
3. Amazon.com
4. Taobao
5. Google IN
6. Sina
7. Google DE
8. Google HK
9. Wordpress
10. eBay

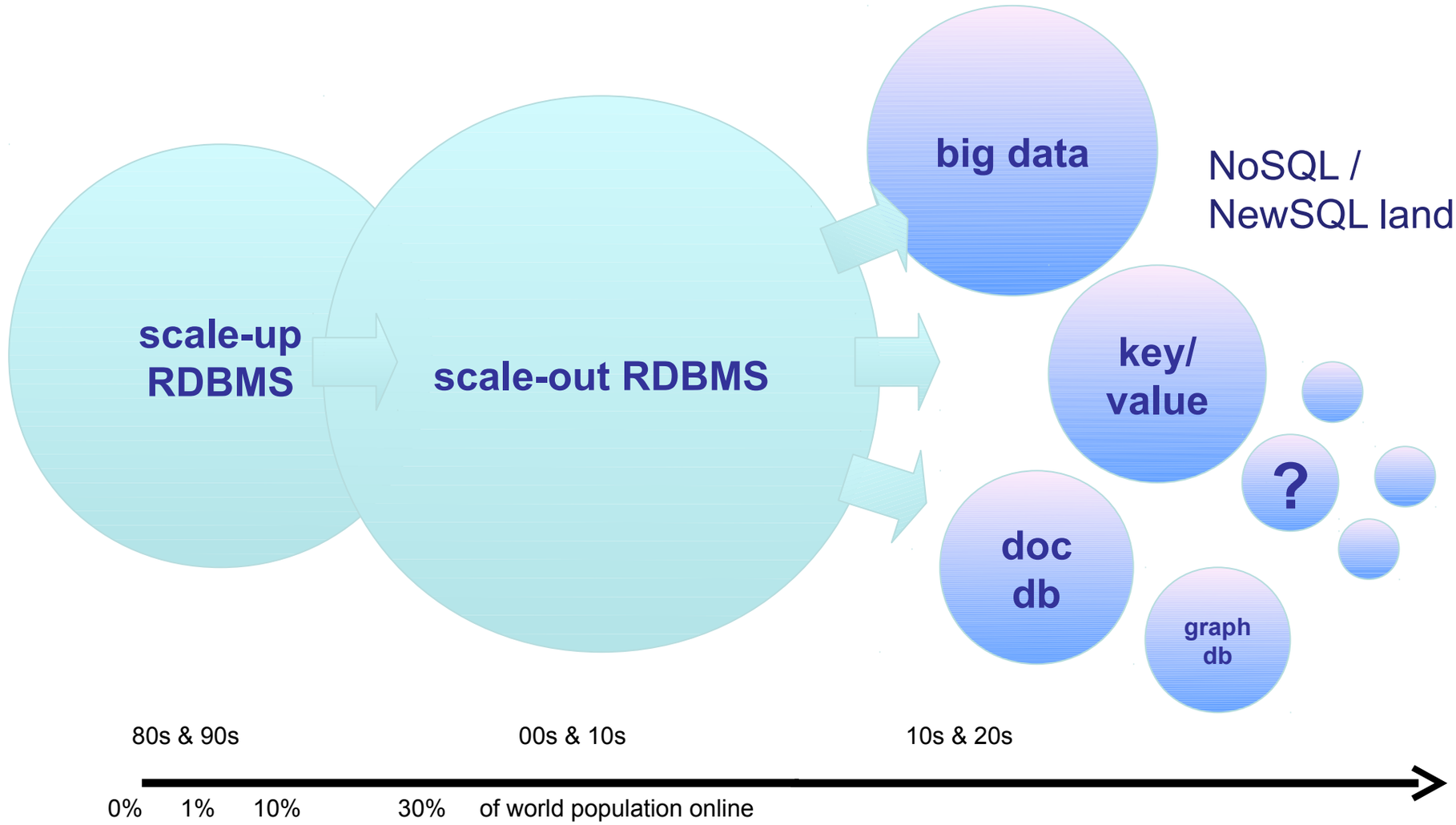
Source: Alexa Top Sites - 14 December 2010

SkySQL Ab

- Funded by:
  - MySQL AB founders Monty Widenius and David Axmark
  - US Investment group OnCorps.org
- Operating in 13 countries, with 90% of the team ex-MySQL® AB employees
- Backed by:
  - Product Engineering MontyProgram Ab
  - Top Community contributors and End Users

SkySQL partenariat de choix

MySQL® écosystème de compétition



http://www.oracle.com/us/corporate/press/042364

MySQL® NoSQL compétition

Akiban, Cassandra, Citrusleaf, Clustrix, Couchbase, Dynomite, FlockDB, GenieDB, Hadoop, Hive, HyperGraphDB, HyperTable, MarkLogic Server, Memcached, MemSQL, MongoDB, MySQL, MySQL Cluster, MySQL with HandlerSocket, Neo4J, NimbusDB, Objectivity/DB, Ravel, Redis, RethinkDB, Riak, SimpleDB, Terracotta, Tokyo Cabinet, Voldemort, VoltDB, Xeround, Sphinx,HBase

Eventuellement consistant

Atomic, **C**onsistency , **I**solation, **P**artitionTolerance

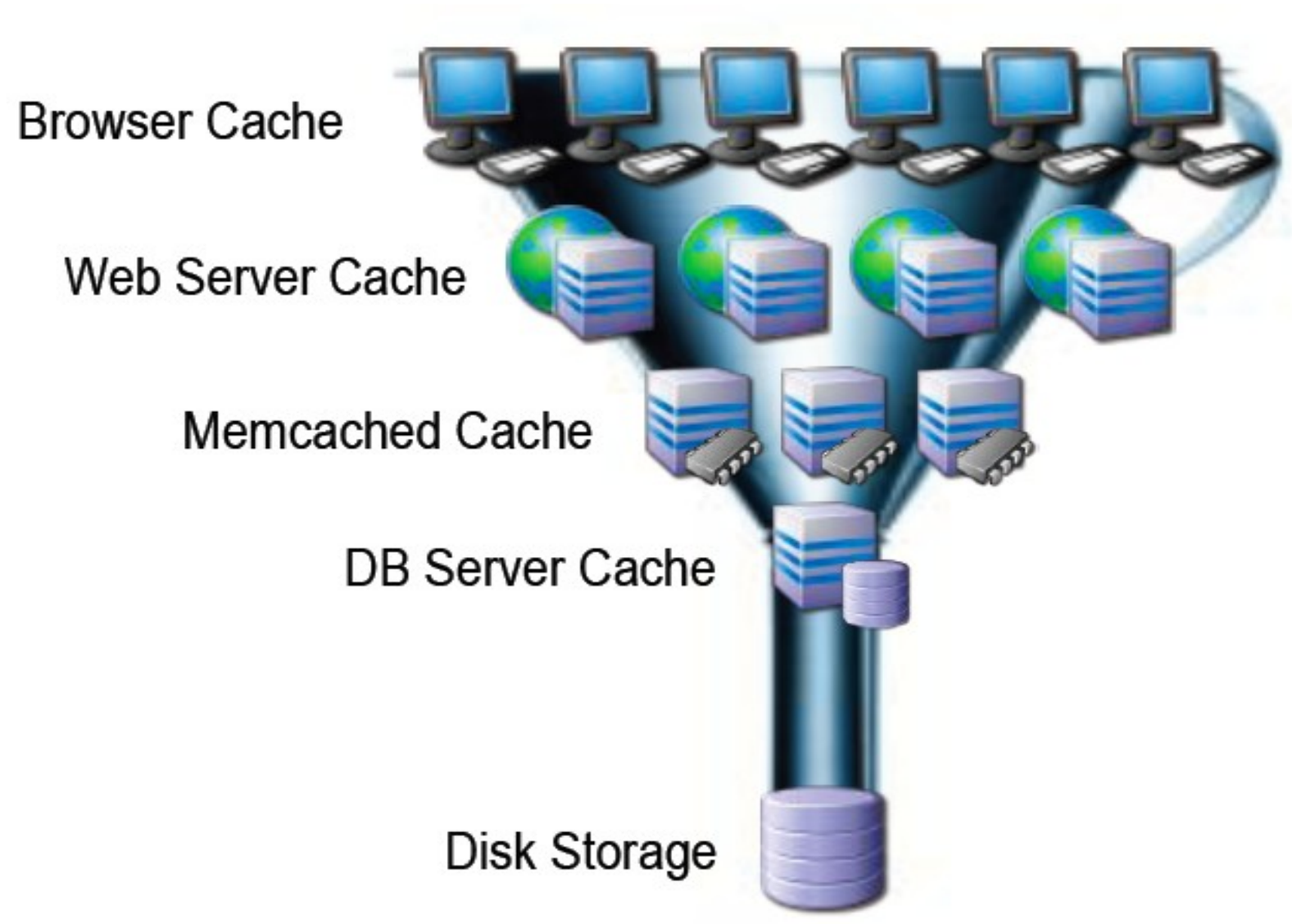
2PC, Lock Manager,Certificate
MySQL Cluster, ScaleDB, VoltDB, Clustrix

Consistency, **A**vailability, **P**artition Tolerance

Assync Replication,Vector clock, Timestamping, ITC
CA Hbase, AP Cassandra

Cache

Cache


Browser Cache
Web Server Cache
Memcached Cache
DB Server Cache
Disk Storage

LAMP = caching

Dans la BDD

- Consistant, sécurisé
- Cache résultats (Protocole,Parsing, Exécution)
- Cache mémoire (disque IO)

Dans les front (ex APC)

- Cache résultats, objets en locale
- Réseau, BDD exécution, client exécution
- Dupliqué, taille limité

Dans le NoSQL (ex Memcache)

- Scalable (BDD exécution, client exécution)
- Memory vs Disques

Dans le cloud (ex CDN , proxy)

Contrôle du cache

Invalidation du cache

- Direct invalidation on update
- Timeout d'invalidation

Logique dans l'application ?
Architecture assez performante a froid ?
QUID des résultats non déterministes ?

Distribution du cache

- Double indirection par clef de domaine et versionning

Monitoring

MEMCACHE

MEMCACHE - Définition

Table de hash en cluster :

- set/get/replace
- append/prepend
- Autoincrement

Pas une database, pas de backup,d’itération,de persitence, de redundance, de sécurité

MEMCACHE - Internals

Serveur

- Never bloks
- libevents epoll/kqueue
- slab alocator
- LRU
- Thread per cpu

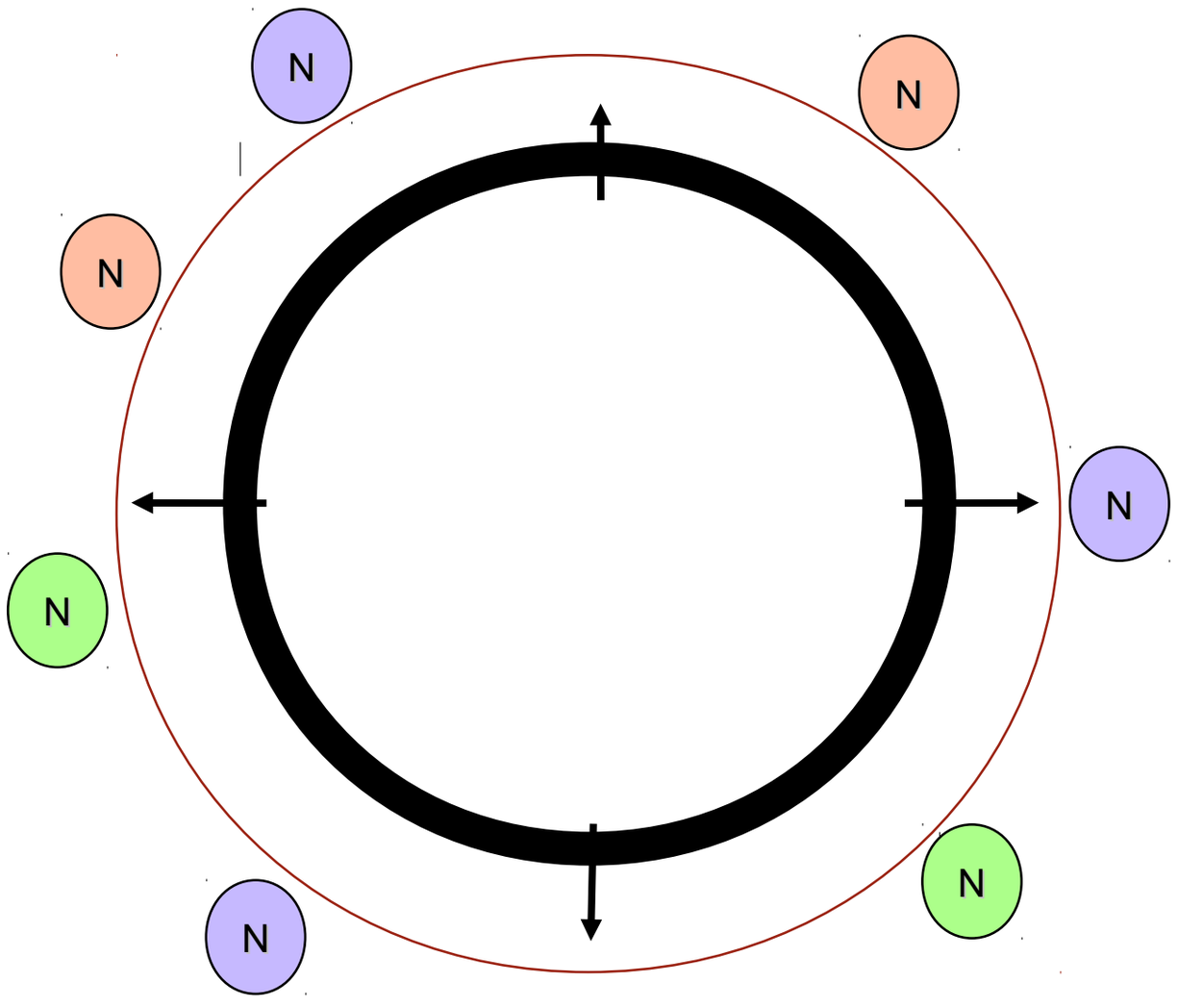
Protocol

- Telnet’able (default port 11211)
- All text base, ex :stats
- Store binary (nothing is escaped)
- Limit 1M object 500B key

MEMCACHE - Internals

Consistante hashing

- Distribution des clefs non linéaire



MEMCACHE dans l'architecture

Invalidation du cache

- Direct invalidation on update
- Timeout d'invalidation

Logique dans l'application ?
Architecture assez performante a froid ?
QUID des résultats non déterministes ?

Fragmentation du cache

- Double indirection par clef de domaine et gestion de version

Monitoring

Memcache usage

```
./memcached -d -m 2048 -l 192.168.0.10 -p
  11211 -u mysql -vv

Telnet 127.0.0.1 11211
Set <key> <flags> <exptime> <bytes>


set key1 0 0 5
hello
STORED
get key1
VALUE key1 0 5
hello
END
```

```
stats slabs
STAT 1:chunk_size 96
STAT 1:chunks_per_page 10922
STAT 1:total_pages 1
STAT 1:total_chunks 10922
STAT 1:used_chunks 1
STAT 1:free_chunks 0
STAT 1:free_chunks_end 10921
STAT 1:mem_requested 74
STAT 1:get_hits 1
STAT 1:cmd_set 3
STAT 1:delete_hits 0
STAT 1:incr_hits 0
STAT 1:decr_hits 0
STAT 1:cas_hits 0
STAT 1:cas_badval 0
STAT active_slabs 1
STAT total_malloced 1048512
```

```
stats items
STAT items:1:number 1
STAT items:1:age 9566
STAT items:1:evicted 0
STAT items:1:evicted_nonzero 0
STAT items:1:evicted_time 0
STAT items:1:outofmemory 0
STAT items:1:tailrepairs 0
STAT items:1:reclaimed 0
```

```
stats
STAT pid 59541
STAT uptime 9617
STAT time 1303251992
STAT version 1.4.5
STAT pointer_size 64
STAT rusage_user 0.137825
STAT rusage_system 0.253672
STAT curr_connections 10
STAT total_connections 11
STAT connection_structures 11
STAT cmd_get 1
STAT cmd_set 3
STAT cmd_flush 0
STAT get_hits 1
STAT get_misses 0
STAT delete_misses 0
STAT delete_hits 0
STAT incr_misses 0
STAT incr_hits 0
STAT decr_misses 0
STAT decr_hits 0
STAT cas_misses 0
STAT cas_hits 0
STAT cas_badval 0
STAT auth_cmds 0
STAT auth_errors 0
STAT bytes_read 154
STAT bytes_written 188
STAT limit_maxbytes 67108864
STAT accepting_conns 1
STAT listen_disabled_num 0
STAT threads 4
STAT conn_yields 0
STAT bytes 74
STAT curr_items 1
STAT total_items 1
STAT evictions 0
STAT reclaimed 0
END
```

Handler Socket

Handler Socket

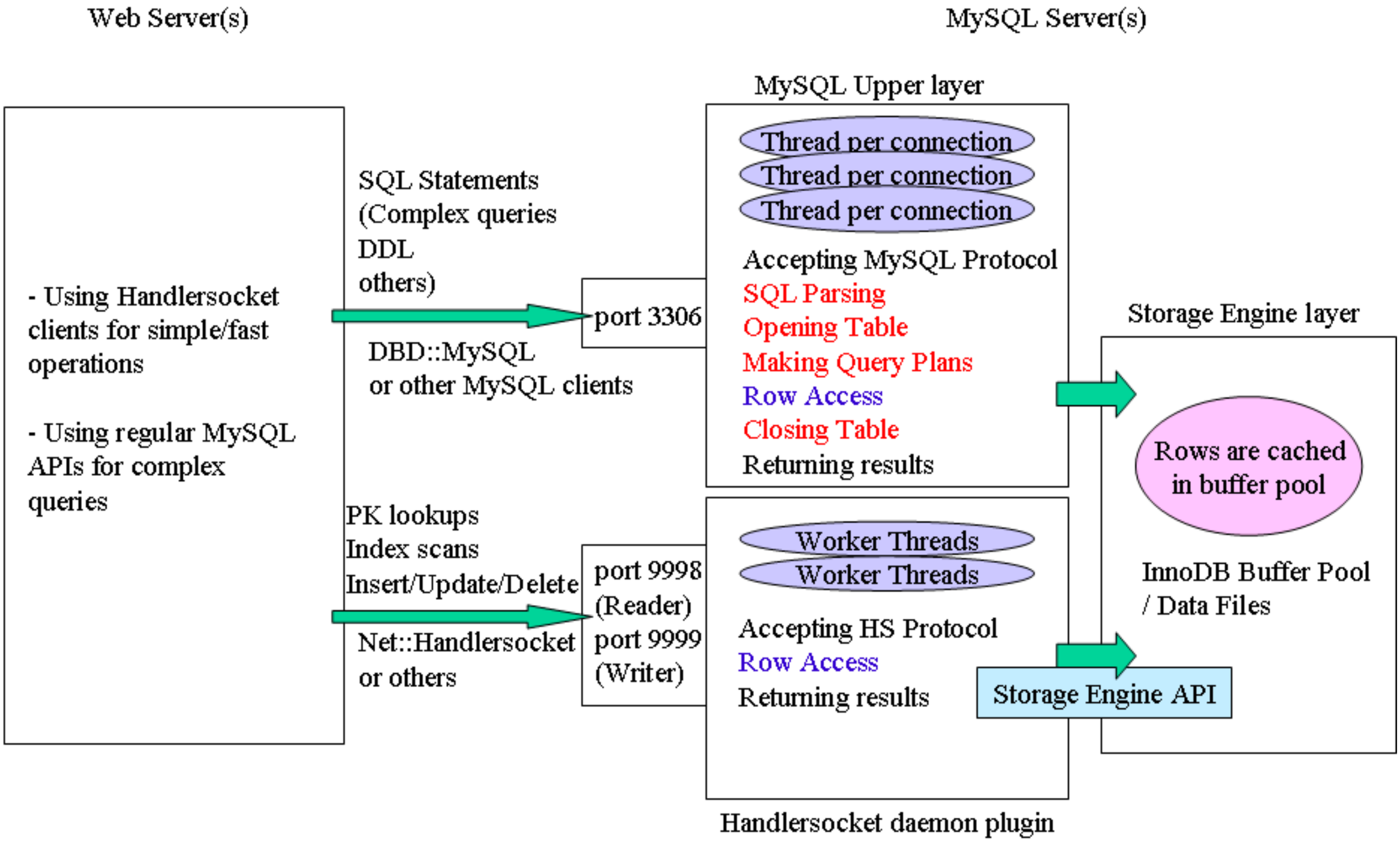
MySQL plugin :
https://github.com/ahiguti/HandlerSocket-Plugin-for-MySQL

- POC avec memcache Kazuho Oku at Cybozu
- Code par Akira Higuchi chez DeNA
- Déployé par Yoshinori Matsunobu chez DeNA

Handler Socket

- HandlerSocket manipule les données sans parser le SQL, moins de consommation CPU .
- HandlerSocket lit plusieurs requêtes et les exécute en bulk,réduit le CPU et l'usage des disques.
- HandlerSocket protocole client/server est plus compacte que mysql/libmysql pair, moins de consommation CPU sur le réseau.
- HanlerSocket est sujet au queeing (ne pas negliger la mémoire)

Handler Socket



Handler Socket API

PHP
http://openpear.org/package/Net_HandlerSocket
http://github.com/tz-lom/HSPHP
http://code.google.com/p/php-handlersocket/
Java
http://code.google.com/p/hs4j/
http://code.google.com/p/handlersocketforjava/
Python
https://code.launchpad.net/~songofacandy/+junk/pyhandlersocke
Ruby
https://github.com/winebarrel/ruby-handlersocket
https://github.com/miyucy/handlersocket
JavaScript
https://github.com/koichik/node-handlersocket

Handler Socket Setup

- loose_handlersocket_threads
  8 * the number of processor cores

- loose_handlersocket_threads_wr
  number of processor cores

- Auto increment et atomique update pas dans la première version

Handler Socket protocol

- Protocol is a simple request/response protocol
- Support pipelines pour recherche multi ligne
- open, find , find_modify, insert
- HandlerSocket '=', '>', '>=', '<', '<='
- HandlerSocket supporte limit offset

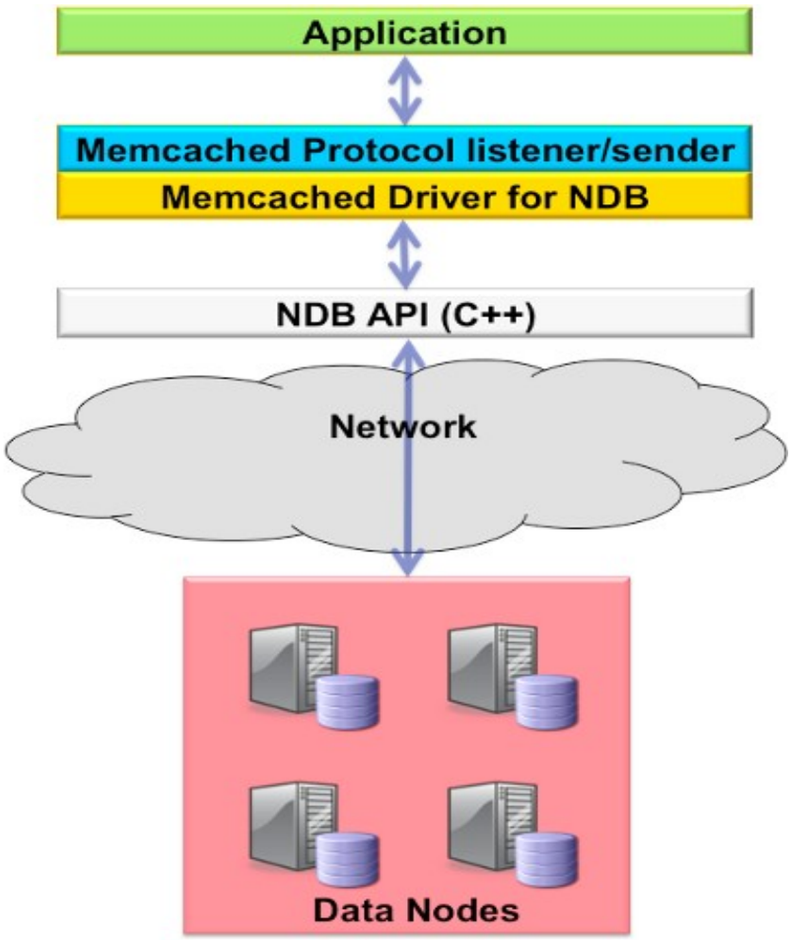
```
INSTALL PERL CLIENTS
./autogen.sh
  $ ./configure --disable-handlersocket-server
  $ make
  $ sudo make install
  $ cd perl-Net-HandlerSocket
  $ perl Makefile.PL
  $ make
  $ sudo make install
```

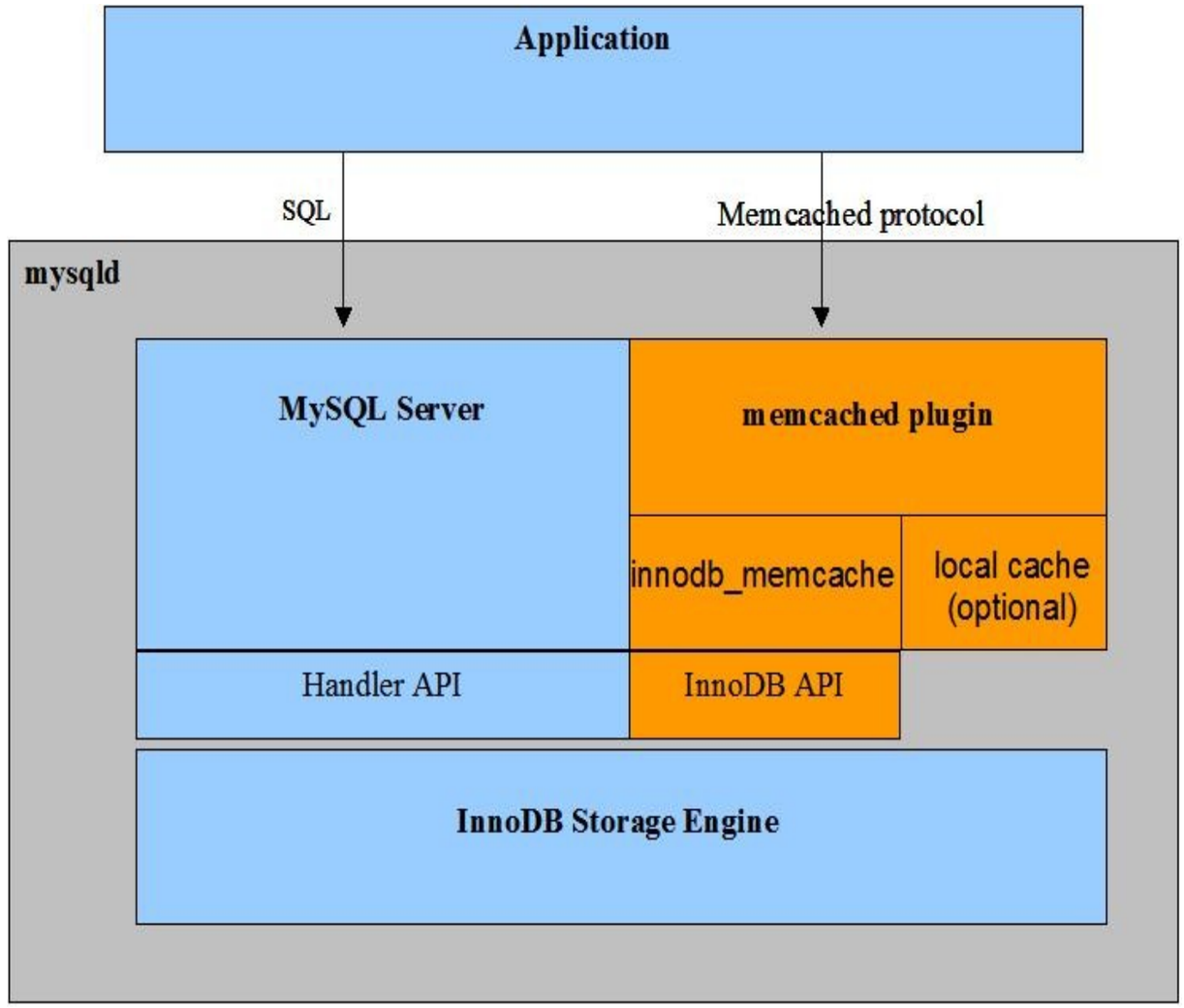
```
#!/usr/bin/perl
use strict;
use warnings;
use Net::HandlerSocket;

my $args = { host => '127.0.0.1', port => 9998 };
my $hs = new Net::HandlerSocket($args);

my $res = $hs->open_index(0,'test',
'login','PRIMARY','user_name,user_email,created
');

die $hs->get_error() if $res != 0;

for (my $search = 0; $search < 10000000; ++$search) {
$res = $hs->execute_single(0, '=', [ '1000' ],
1, 0);
die $hs->get_error() if $res->[0] != 0;
shift(@$res);

for (my $row = 0; $row < 1; ++$row) {
   my $user_name= $res->[$row + 0];
   my $user_email= $res->[$row + 1];
   my $created= $res->[$row + 2];
}
}
```

MySQL est NoSQL

MySQL 5.6.2

- Memcache embarqué
- Backends InnoDB,Memory,NdB
- http://labs.mysql.com





MariaDB “NoSQL”

MariaDB 5.2

- Virtual column = fonctional index

MariaDB 5.3

- HANDLER commands;
  HANDLER READ avec les prepared statements.
- HandlerSocket (accès directe a InnoDB)
- Support des données non structurées via les Dynamic columns
  (chaque ligne peu avoir un ensemble de colonne dynamique)
- http://varokism.blogspot.com/2011/01/20-to-50-improvement-in-mariadb-53.html

MariaDB 5.6

- Hbase Storage engine for dynamique column

MariaDB “Dynamic Column”

```
COLUMN_CREATE(column_nr, value,[column_nr,....])
COLUMN_ADD(blob,column_nr, value, [column_nr,...])
COLUMN_DELETE(blob, column_nr, column_nr...);
COLUMN_EXISTS(blob, column_nr);
COLUMN_LIST(blob, column_nr);
COLUMN_GET(blob, column_nr, type);
```

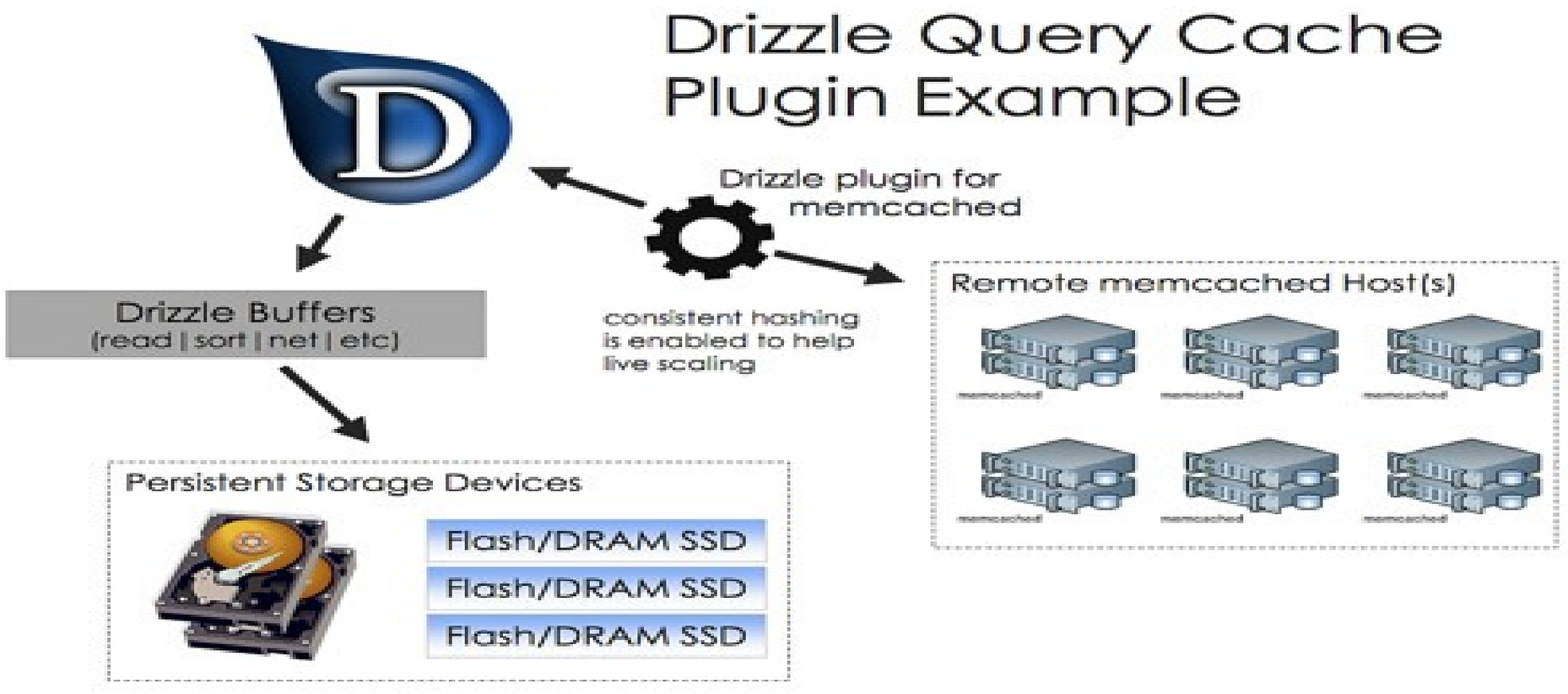
MariaDB sample

```
bzr branch lp:maria/5.3
cd mariadb-5.3/
BUILD/compile-pentium-all –prefix=/usr/local/mariadb-5.3
sudo make install
```

```
int main(int argc, char *argv[])
{
 cout << "start up" << endl;
 MYSQL * connection;
 connection = mysql_init(0);
 connection = mysql_real_connect(connection, 0, "stephane", "", "test", 0,
                                 "/tmp//mysql.sock", 0);
 if (0 == connection)
 {
   cerr << "connect failed:" << mysql_error(connection) << endl;
   return -1;
 }
 static __const char * sql_statement ="HANDLER login READ `PRIMARY`=(?)";
 MYSQL_STMT * stmt_handle = mysql_stmt_init(connection);
 if (0 == stmt_handle)
 {
   cerr << "stmt init failed:" << mysql_error(connection) << endl;
   return -1;
 }
 int mysql_return_code = mysql_stmt_prepare(stmt_handle,
                                            sql_statement,
                                            strlen(sql_statement)
                                           );
if (0 != mysql_return_code)
 {
   cerr << "stmt prepare failed:" << mysql_stmt_error(stmt_handle) <<
     endl;
   mysql_stmt_close(stmt_handle);
   return -1;
 }
```

Drizzle GA

- Plugable Query Cache (memcache)
- Blob Streaming PBMS (HTTP protocole)
- Pas d'authentification possible



MERCI

“NoSQL” No locking 3 replicas

Vector clocks conflit résolution 2 lectures, 2 écritures CAP compatible sans Lock Manager

- Dynamo, Riak,Voldemort, MongoDB
- Dernière écriture gagnante en cas de conflit
- Taille des vecteurs par compactage et ITC pour ajout suppression de noeuds