



Novinky ve starém dobrém SQL světě

Overview

PRESENTED BY:

Honza Horák

Co nás čeká

- Obecný pohled na data ve světě
- Staronoví členové SQL klubu
- MySQL
- MariaDB
- Galera
- PostgreSQL

Malá odbočka...

“Buzzword bingo, zname take jako Bullshit bingo je hra ve stylu binga, kde si ucastnici pripravi karty s buzzword vyrazy a skrtaji je, jakmile jsou pouzite na porade/prednasce. Cil hry je vyskrtat zvoleny pocet slov v radku nebo sloupci a v takovem pripade zvolat 'Bingo' (nebo 'Bullshit')”

(http://en.wikipedia.org/wiki/Buzzword_bingo)



• Nový pohled na data

- Cloud – pets vs. cattle
 - HA, fail-over, replikace, škálování
- Agilní programování
 - Konec vodopádů
 - Frameworky
 - Time To Market
- Méně složitých datových modelů



NoSQL a proč jsou in

- MongoDB
 - JSON
 - Struktura automaticky
 - Replikace a škálování out-of-the-box
 - Map-Reduce model paralelizace
- Cassandra
- Hadoop
- CouchDB
- NoSQL vychytávky označené (§)



Ukázka MongoDB aplikace

```
db = connection.students.myseminar
```

```
# insert a document
```

```
student_record =
```

```
{'name':student_name,'grade':student_grade}
```

```
db.insert(student_record)
```

```
# find all documents
```

```
results = db.find()
```

```
for record in results:
```

```
    print(record['name'] + ', ',record['grade'])
```

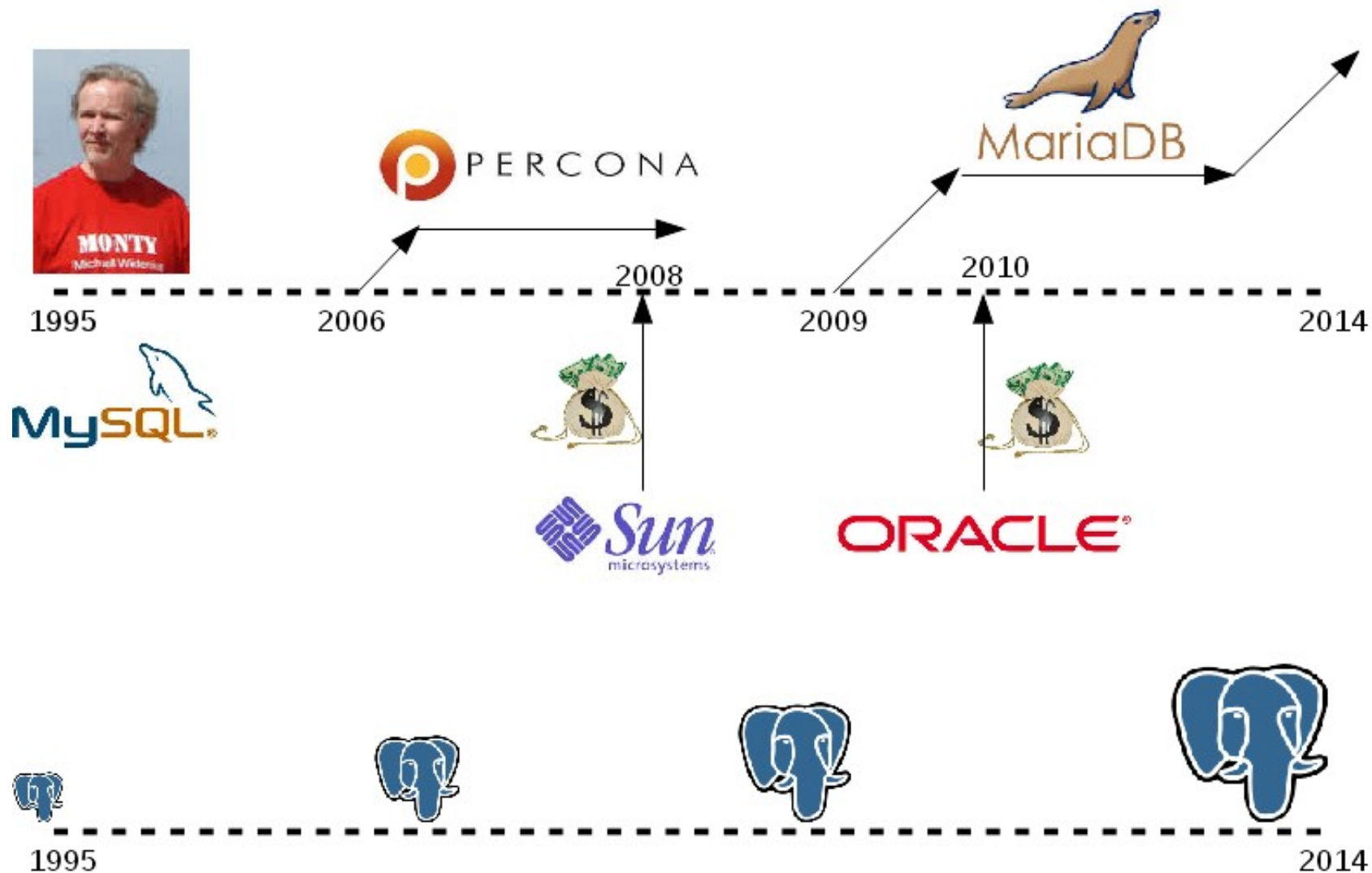

Polyglot persistence

“Polyglot Persistence, like polyglot programming, is all about choosing the right persistence option for the task at hand”

<https://www.altamiracorp.com/blog/employee-posts/polyglot-persistence>



Staronoví členové SQL klubu



MySQL

- Stále podporována 5.5.40
 - GA 5.5 prosinec 2010, EOL prosinec 2015
- Nyní 5.6.21, bugfix verze cca po 2 měsících
 - GA 5.6 únor 2013
- CVE jednou za 3 měsíce; pouze seznam
- Merge kódu při vzdání na lauchpad, jinak vyvíjen uzavřeně
- Komunita mizí, něco pouze v Enterprise verzi
- Důsledek: OS produkt, ne projekt (2015?)



MariaDB

- GA 5.5 duben 2012 (nyní 5.5.40)
- GA 10.0 březen 2014 (nyní 10.0.13)
 - “10” znamená změnu
 - client-server protokol bude fungovat dál
 - Jména souborů in cesty zachovány
- EOL po 5 letech
- Lepší komunita, chce být pravý open-source
- Důraz na delší testování, více testů



Galera

- Codership – patch + plugin k běžnému MySQL
- Synchronní multi-master replikace
 - Zápis může být proveden na libovolném serveru a bude synchronně aplikován v celém clusteru
- MariaDB patch aplikuje a vydává variantu MariaDB-Galera
- Od 10.1 součást hlavní větve MariaDB

PostgreSQL

- Nejčistší open-source projekt
- Stabilní růst
- Minor verze cca po roce
- 9.4 beta3 říjen 2014
- 9.3 září 2013
- Bugfix verze jak je potřeba
- Méně rozšířen zejména z historických důvodů



Sqlite

- 3.8.0 – GA srpen 2013
- Embedded
- Mnoho feature vylepseno
- Zadna tak zajimava jako byl WAL v 3.7.0
 - Nebo je prece v 3.8.0 neco, co bychom meli zminit?

MySQL 5.5 features

- InnoDB defaultní engine
- Semi-Synchronous Replication
 - Master čeká dokud alespoň jeden slave nepotvrdí přijetí
- Replication Heartbeat
- Performance Schema
 - Přístup k low-level MySQL performance
- Nativní Asynchronous I/O for Linux
- Zlepšen výkon i škálovatelnost jednoho serveru



MySQL Performance schema

```
mysql> SELECT EVENT_ID, EVENT_NAME, TIMER_WAIT  
-> FROM events_waits_history WHERE THREAD_ID = 13  
-> ORDER BY EVENT_ID;
```

EVENT_ID	EVENT_NAME	TIMER_WAIT
86	wait/synch/mutex/mysys/THR_LOCK::mutex	686322
87	wait/synch/mutex/mysys/THR_LOCK_malloc	320535
88	wait/synch/mutex/mysys/THR_LOCK_malloc	339390
89	wait/synch/mutex/mysys/THR_LOCK_malloc	377100
90	wait/synch/mutex/sql/LOCK_plugin	614673
91	wait/synch/mutex/sql/LOCK_open	659925
92	wait/synch/mutex/sql/THD::LOCK_thd_data	494001
93	wait/synch/mutex/mysys/THR_LOCK_malloc	222489
94	wait/synch/mutex/mysys/THR_LOCK_malloc	214947
95	wait/synch/mutex/mysys/LOCK_alarm	312993

(<http://dev.mysql.com/doc/refman/5.5/en/performance-schema-quick-start.html>)



MySQL 5.6 features

- Výkon++
 - Index Condition Pushdown, Multi-Range Read, File Sort Optimization
- Partitioning explicitně definován
- Multi-threaded slaves
- Crash-Save slaves
- GTID (3E11FA47-71CA-11E1-9E33-C80AA9429562:1-5)
- Memcached (§)

(<http://dev.mysql.com/tech-resources/articles/whats-new-in-mysql-5.6.html>)



MySQL 5.6 memcached

```
mysql> CREATE TABLE `test`.`users` (  
  `user_id` varchar(32) NOT NULL DEFAULT '',  
  `first` varchar(100) DEFAULT NULL,  
  `last` varchar(100) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB;
```

```
mysql> INSERT INTO `containers` (`name`, `db_schema`, `db_table`, `key_columns`,  
  `value_columns`, `unique_idx_name_on_key`)  
VALUES ('default', 'test', 'users', 'user_id', 'first|last', 'PRIMARY');
```

```
<?php  
$memcache = new Memcache;  
$memcache->connect('localhost', 11211);  
$memcache->set('bar', 'John|Smith');  
?>
```

```
mysql> select * from `test`.`users`;  
+-----+-----+-----+  
| id   | first | last   |  
+-----+-----+-----+  
| bar  | John  | Smith  |  
+-----+-----+-----+
```

(<http://chipersoft.com/p/MySQL-via-Memcache/>)



MariaDB 5.5 features

- InnoDB defaultní engine (vlastně XtraDB)
- Aria – crash safe MyISAM
- Connect, TokuDB, SphinxSE
- Non-blocking API u klienta
- Vybrané vychytávky z 5.6
- Extended Keys (index-only scans)
- Další vychytávky:
 - LIMIT ROWS EXAMINED, nové status proměnné



MariaDB 5.5 Connect

```
$ cat boys.txt
```

John	Boston	25/01/1986	02/06/2010
Henry	Boston	07/06/1987	01/04/2008
George	San Jose	10/08/1981	02/06/2010
Sam	Chicago	22/11/1979	10/10/2007
James	Dallas	13/05/1992	14/12/2009
Bill	Boston	11/09/1986	10/02/2008

```
mysql>
```

```
create table boys (  
  name char(12) not null,  
  city char(12) not null,  
  birth date not null date_format='DD/MM/YYYY',  
  hired date not null date_format='DD/MM/YYYY' flag=36)  
engine=CONNECT table_type=FIX file_name='boys.txt' lrecl=48;
```

Zdroj:

<https://mariadb.com/kb/en/mariadb/documentation/storage-engines/connect/connect-table-types/connect-table-types-data-files/>

MariaDB 10.0 features

- GTID, ale trochu jinak než v MySQL (např. 0-1-10)
- Co má navíc oproti MySQL: see link bellow
- Cassandra (§), Spider engine (§), flexible parallel slave repl.
- Engine Independent Statistics
- Dynamické sloupce (§)
- Vychytávky dostupné jen v MySQL Enterprise
 - Roles, Audit plugin, PAM plugin
- <https://mariadb.com/blog/mariadb-10-vs-mysql-56-feature-comparison-update>



MariaDB dynamic columns

```
create table assets (  
  item_name varchar(32) primary key, -- A common attribute for all items  
  dynamic_cols blob -- Dynamic columns will be stored here  
);  
  
INSERT INTO assets VALUES  
  ('MariaDB T-shirt', COLUMN_CREATE('color', 'blue', 'size', 'XL'));  
INSERT INTO assets VALUES  
  ('Thinkpad Laptop', COLUMN_CREATE('color', 'black', 'price', 500));  
  
SELECT item_name, COLUMN_GET(dynamic_cols, 'color' as char) AS color FROM  
assets;
```

item_name	color
MariaDB T-shirt	blue
Thinkpad Laptop	black

(<https://mariadb.com/kb/en/mariadb/documentation/nosql/dynamic-columns/>)



PostgreSQL 9.0

- **Hot Standby + Streaming replikace**
- Passwordcheck
- Sloupcové + WHEN trigger
- Vylepšený VACUUM – rychlost
- PL/pgSQL by default
- Hstore (§) vylepšení



PostgreSQL 9.1

- Nelogované tabulky
- Writeable Common Table Expressions
- SE-Postgres, foreign tables (nyní jen SELECTy)
- K-nejbližší sousedi:

```
SELECT text_data, similarity(text_data, 'hello')  
FROM test_trgm WHERE text_data % 'hello'  
ORDER BY similarity(text_data, 'hello') LIMIT 2;
```

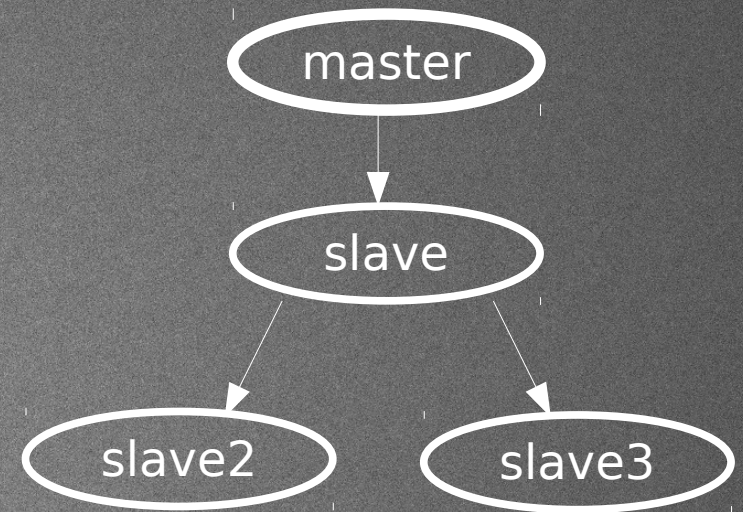
```
SELECT text_data, text_data <-> 'hello'  
FROM test_trgm  
ORDER BY text_data <-> 'hello' LIMIT 2;
```

([https://wiki.postgresql.org/wiki/What%27s_new_in_PostgreSQL_9.1#K-Nearest-Neighbor_Indexing](https://wiki.postgresql.org/wiki/What%27s_new_in_PostgreSQL_9.1#K-Nearest_Neighbor_Indexing))



PostgreSQL 9.2

- Index-only scans
- Kaskádová replikace
- JSON datový typ
- Datový typ RANGE
- SP-GiST – pro neuniforme distribuovaná data
- Z hstore mizí => operátor



PostgreSQL – dat. typ Range

```
=# SELECT '(1000,2000]'::numrange * '[1000,1200]'::numrange;  
?column?
```

```
-----  
(1000,1200]  
(1 row)
```

```
=# SELECT * FROM test_range WHERE period && '[2012-01-03 00:00:00,2012-  
01-03 12:00:00]';
```

```
period  
-----  
["2012-01-01 00:00:00+01", "2012-03-01 00:00:00+01"]  
["2008-01-01 00:00:00+01", "2015-01-01 00:00:00+01"]  
(2 rows)
```

(https://wiki.postgresql.org/wiki/What%27s_new_in_PostgreSQL_9.2#Range_Types)



PostgreSQL 9.3

- JSON je použitelný (§)
- Event Triggers na DDL
- Materializované pohledy
- Updatovatelné pohledy
- Zapisovatelné Foreign tables
- Streaming-Only Remastering a spol.
- Background workers



PostgreSQL – dat. typ JSON

```
CREATE TABLE books ( id integer, data json );
```

```
INSERT INTO books VALUES (1,  
    '{ "name": "Book the First", "author": { "first_name": "Bob",  
    "last_name": "White" } }');
```

```
INSERT INTO books VALUES (2,  
    '{ "name": "Book the Second", "author": { "first_name":  
    "Charles", "last_name": "Xavier" } }');
```

```
SELECT id, data->>'name' AS name FROM books;
```

id	name
1	Book the First
2	Book the Second

Zdroj:

<http://clarkdave.net/2013/06/what-can-you-do-with-postgresql-a-json/>



PostgreSQL 9.4

- JSONB ($\neq$ BSON) (§§§)
- Obnovení materializovaných pohledů
CONCURRENTLY
- WITH CHECK
- Rychlost agregate funkcí



Otázky?

Honza Horák <hhorak@redhat.com>

IRC: hhorak @ freenode

<http://themindiseverything.eu/>

Twitter: @HonzaHorak



Slidy již teď k dispozici na:

<http://hhorak.fedorapeople.org/slides/2014-11-02-openalt-db-news.pdf>

