

mysqlnd

Нова ера

WebTech 2007
Варна, България

9-ти Юни 2007

Андрей Христов <andrey@mysql.com>

Лекторът

Андрей Христов

- работи за MySQL като разработчик
- в момента част от частта занимаваща с клиентските библиотеки
- автор на MySQL Event Scheduler, част от 5.1 (дипломна работа)
- бърка в кода на PHP от години
- разработил почти изцяло mysqlnd
- поддържа ext/mysql и ext/mysql_i заедно с Georg Richter
- ... ще се опитам да отговоря на въпросите си - просто ги задайте! Ако сте срамежливи, ще ви очаквам след лекцията.
- ... отговарям на, някои, писма изпратени до мен на andrey@mysql.com



Какво означава „mysqlnd“?

СУРБД MySQL

- **The world's most popular open source database**
- **'LAMP'** (Linux, Apache, **MySQL**, **PHP** / Perl / Python)
- <http://dev.mysql.com/>

mysqlnd - MySQL native driver for PHP

- **PHP специфичен драйвер за свързване на PHP с MySQL.**
- Силно интегриран в PHP, използва максимално от биосферата.
- **Не е нов модул (разширение), за PHP.**
- <http://dev.mysql.com/downloads/connector/php-mysqlnd/>

По подробно..

Библиотека, а не нов модул

- Не се вижда директно от скриптовете
- Реализиран на C

(Почти) директен еквивалент на `libmysql`, за PHP

- работи със съществуващия софтуер
- почти: някои функции, които от години са обявени за експериментални не са достъпни : `mysqli_rpl_*`, вграден MySQL (embedded MySQL)

Дали да се използва се избира при компилация

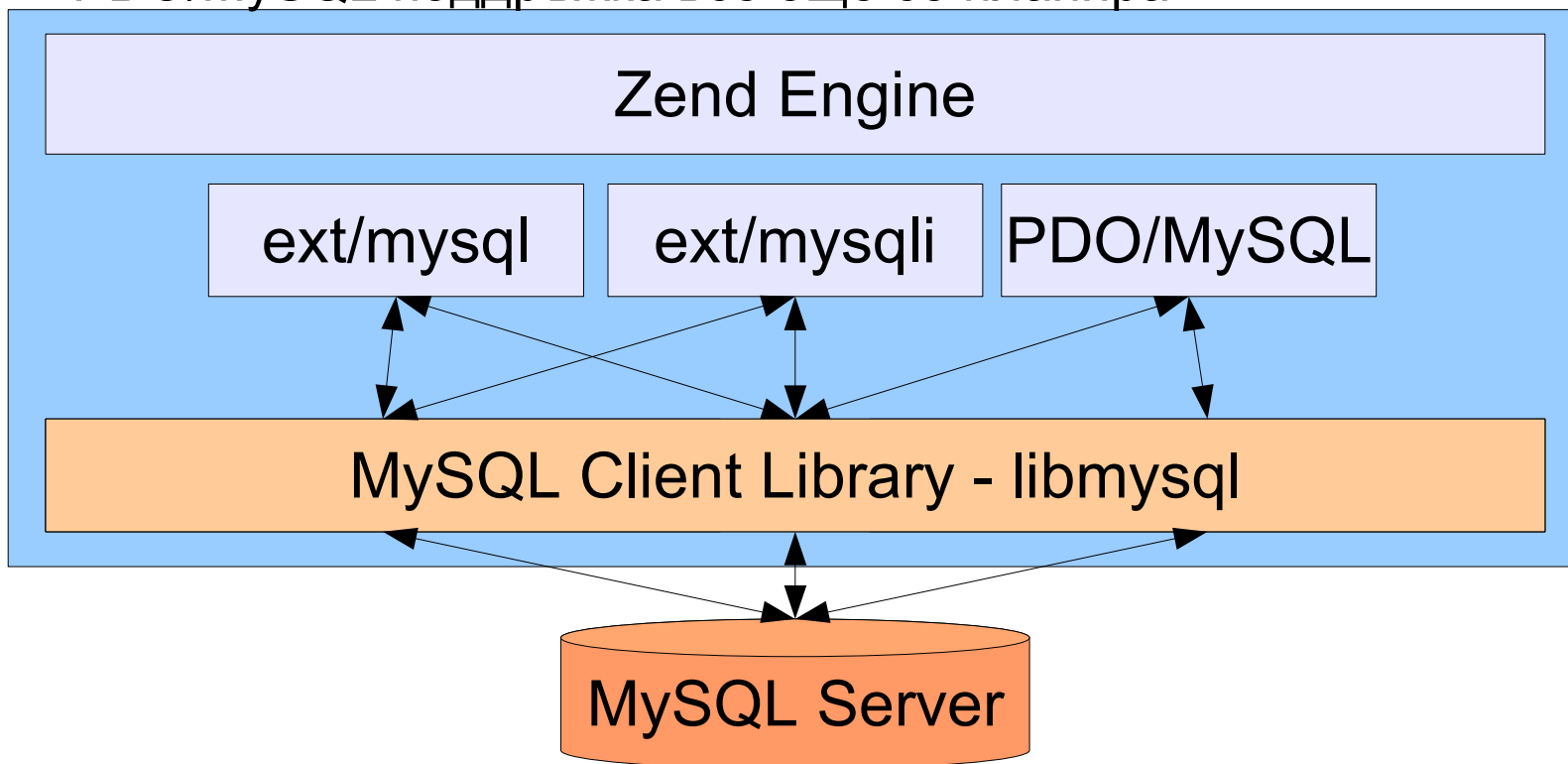
- засега `ext/mysqli` има официална версия, която го поддържа, `ext/mysqli` код в Subversion хранилище.
- PDO/MySQL поддръжка се планира

Независим от версията на PHP. Работи с 5 и 6

- все още не е портиран за 4. Потоците са по-различни.

Обща картина, или как PHP се закача за MySQL

- libmysql остава като възможност
- mysqlnd и libmysql могат да съществуват независимо в един изпълни файл.
- PDO/MySQL поддръжка все още се планира



Защо mysqlnd? Лицензът!

- mysqlnd е лицензиран под PHP License v. 3.01
- т.е. BSD License
- дава възможност ext/mysql и ext/mysql да са модули компилирани по подразбиране.

Защо mysqlnd? Предимствата (1)

Libmysql идва като част от MySQL Server пакетите

- обновяване на libmysql означава обновяване на MySQL Server пакетите. Е, възможно е и само клиентските.
- libmysql е силно свързана с пускането на MySQL версии.

mysqlnd е библиотека (кодът е в ext/mysqli/mysqlnd)

- обновяване, например след бъг, е просто!
- има собствен календар за нови версии, по-чест
- никога повече проблеми при компилиране и свързване
- не е нужно да има libmysql инсталирна на системата
- по-чист код, libmysql е стара библиотека

Тясно работещ с Zend Engine

- прави магии за да използва оптимално вашите процесори
- по-добре интеграция с PHP (повече на следващия слайд)

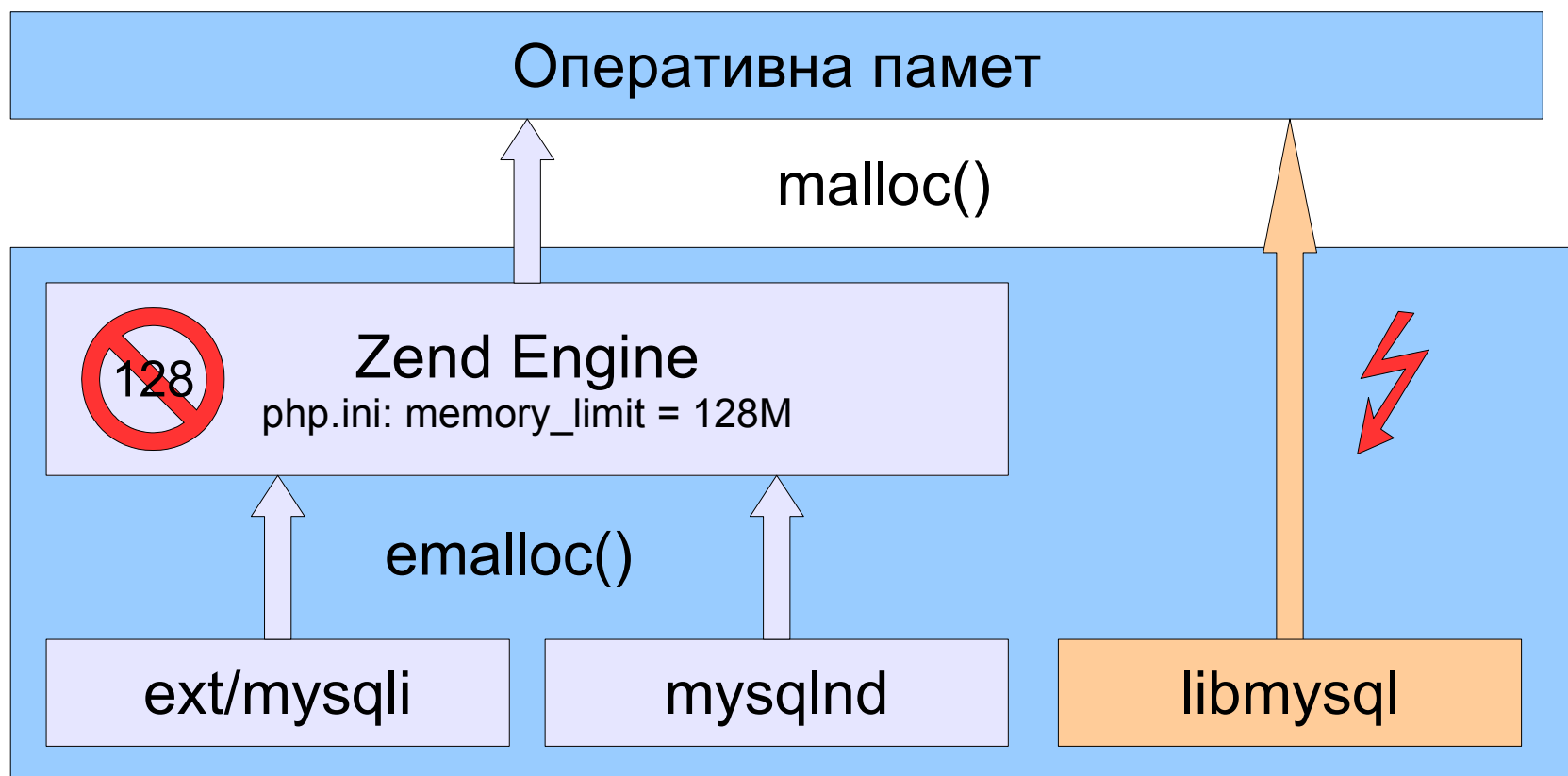
Недостатъците?

По-малко тестван от комбинацията ext/mysql - libmysql

- НО, ние удвоихме броя на тестовете и mysqlnd ги преминава всичките.
- дори намерихме бъгове в libmysql и MySQL по време на разработката
- имаме колега, който пише тестовете и има силно деструктивно мислене. Не му давайте автоматично оръжие!!
- повечето тестове са с конзолната версия на PHP
- не много информация под формата на обратна връзка от потребители

Абракадабра: memory limit

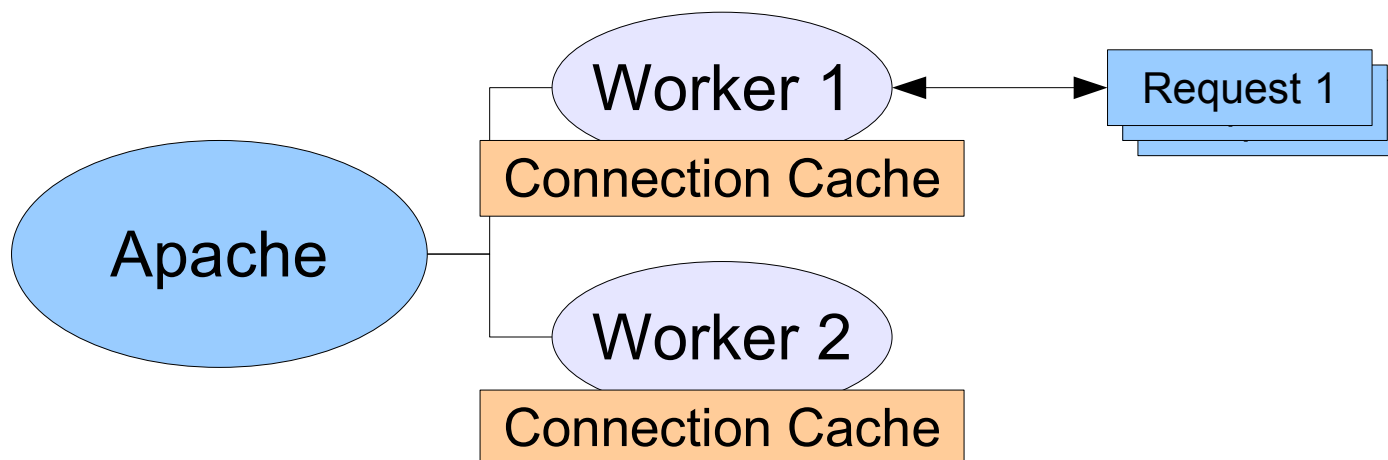
Memory limit (--enable-memory-limit) вече се поддържа!



Абракадабра: постоянни връзки

Постоянни връзки както при ext/mysql и libmysql

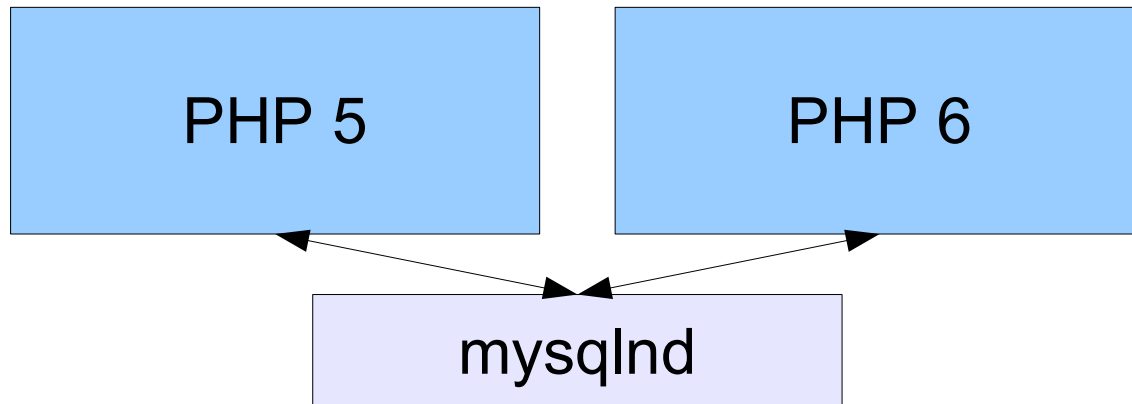
- спестява времето за свързване, колкото и да е малко то: +5% за Dell DVD Store
- работи с Apache prefork и Fast-CGI
- в процес, без поделяне между тях, иначе би било трудно :)
- Допълнителен флаг, за олекотено или нормално почистване на връзката от предишния скрипт.
- `mysql_i_connect('p:localhost', ...)`



Abracadabra: one code base

PHP 5 and PHP 6 use the same code base

- all extensions use one code base of mysqlnd
- one fix, fixes it all (one bug, breaks it all)
- Unicode is compiled out for PHP 5
- nevertheless the code has only few `#ifdef` and is readable
- If php.net wants mysqlnd they can go for one or two branches

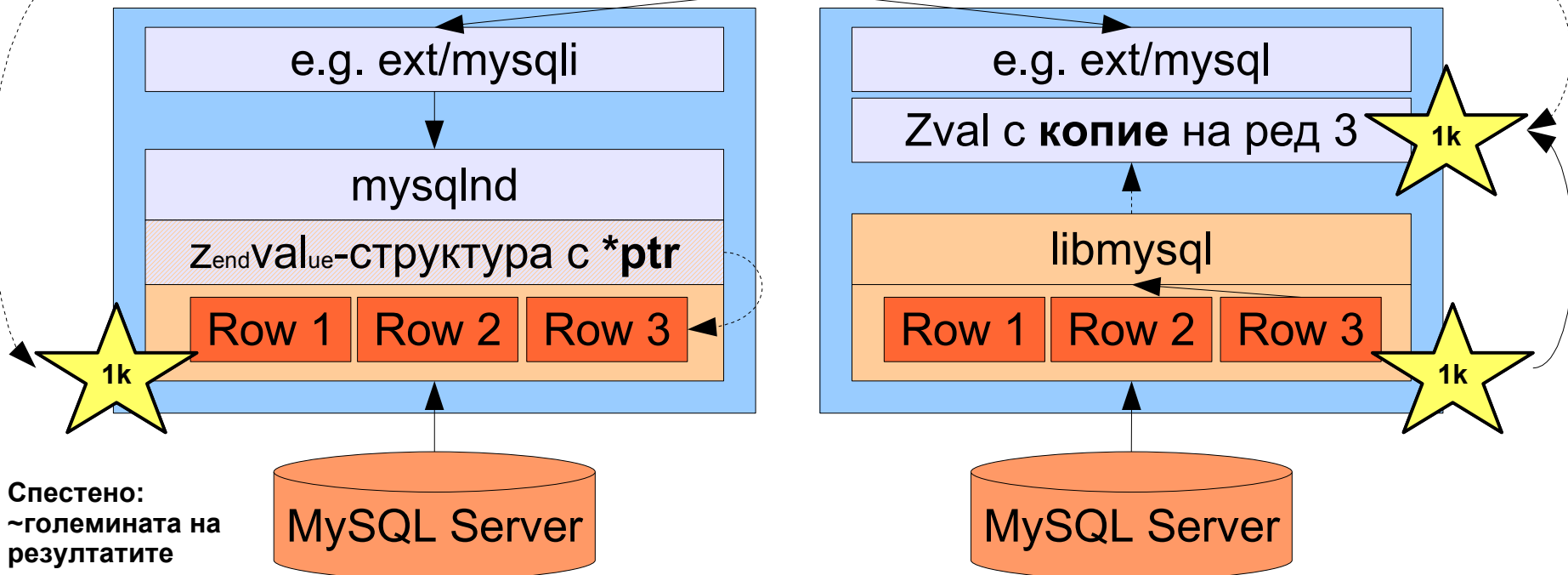


Абракадабра: пестене на памет

„Read-only“ променливи преди PHP да ги има

- mysqlnd се опитва да пази само едно копие на данните в паметта, за разлика от ext/mysql + libmysql, а може би и други модули за достъп на СУБД.

```
<?php $a = mysqli_fetch_assoc($res) ?>
```



Реализацията на променливи само за четене

Резултатът се състои от:

- двуизмерен zval буфер
- даните от сървър
- други

zval собственост на резултатат:

- refcount = 1

Извличане на данни:

- бързо!: refcount++
- за разлика от libmysql имената на колоните се хешират веднъж, защото те всъщност не се променят!
- `unset($row): refcount--`

`mysqli_free_result():`

- Разделяне (`zval_copy_ctor()`) на всички zval променлив, за които все още има референция (и):
`$lover = $row['име']`

```
zval.type      = string
zval.value     = *ptr_row1_col1
zval.refcount  = 1
```

Row 1 - 1k 3 Иван 5 желяе 8 Мария ...

Други

```
zval.type      = string
zval.value     = *ptr_row1_col1
zval.refcount  = 2 (3, при fetch_array)
```

Row 1 - 1k 3 Иван 5 желяе 8 Мария ...

Други

```
zval.type      = string
zval.value     = 'Иван'
zval.refcount  = 1
```

Управление на паметта от mysqlnd

Библиотеките за заделяне на памет са синхронизирани

- синхронизацията забавя, особено когато се използва критична секция.
- случва се дори и когато имате еднонишково приложение, каквото е PHP, в повечето случаи

mysqlnd заделя памет на големи блокове

- може би заделяме повече
- но минимизираме фрагментацията на хийпа (heap)

mysqlnd се опитва да преизползва zval памет

- пести цикли на процесора
- по-малко фрагментация на хийпа ;)
- НО: ще го забележите ли?

Магия: zval кеш

- При инициализация на модула, MINIT за тези, познаващи PHP, zval кешът се инициализира. Заделя се блок от памет само за zval структури, големината се определя от ini настройка.
- mysqlnd усилено работи със zval структури. Когато се нуждае от една такава, mysqlnd се обръща към кеша, и се опитва да намери свободна в него. Ако няма празни позиции, се създава zval променлива по обикновения начин.
- Когато резултат бива освободен (mysql_free_result), zval идващи от кеша се освобождават, освен ако няма референции към тях. Ако има активни референции, тогава позицията от кеша не се освобождава до края на скрипта. Това може да бъде решено с почистване на боклука, на случаен принцип, при следващо освобождаване на памет.
- zval не от кеша се разпознават лесно, те не са в региона от памет заделен от кеша.

zval кеш (продължение)

Глобалният, живеещ в рамките на един процес, кеш се състои от:

- предварително заделен блок памет
- стек с указатели към свободни позиции от блока памет
- брояч на референциите
- двоичен семафор (mutex), за случаите когато PHP е компилирано с нишки.
- други променливи свързани с работата и поддържане на статистики на работата на библиотеката
- създава се по време на инициализация на модула, MINIT

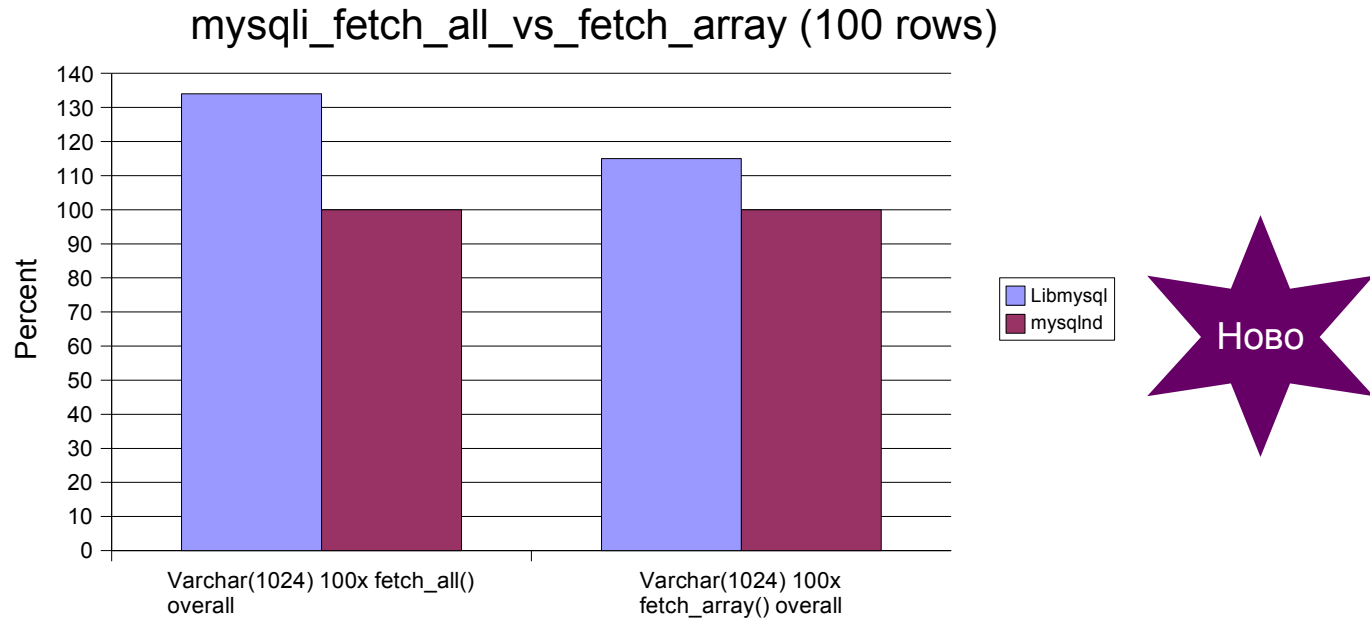
Локалният, за всяка нишка, кеш се състои от:

- стек за почистване на боклука, в момента не се използва
- брояч на референциите
- указател към глобалния кеш
- създава се в началото на всяко изпълнение на скрипт, RINIT

Проблеми с кеша

- В университетите се обяснява, че оптимизацията не може да е максимална. Винаги се жертва нещо, памет, когато се кешира или процесорно време, когато паметта не е много.
- Дълги години, в документация, в статии и др. ви е било обяснявано, че трябва да се извика `mysql_free_result()` възможно най-бързо след като се свърши работа с данните. Това е много добра стратегия при работа с **libmysql**, но но и при `mysqlnd`. `mysqlnd` се опитва по-рядко да копира памет, и това означава възможно най-късно извикване на `mysql_free_result()`.
- Ако `mysql_free_result()` бъде извикан преждевременно производителността на `mysqlnd` спада до тази на `libmysql`.

Hobo: mysqli_fetch_all()



```
<?php
...
$res = mysqli_query('SELECT ...');
$users = mysqli_fetch_all($res);
...
?>
```

```
<?php
...
$res = mysqli_query('SELECT ...');
while ($users[] =
    mysqli_fetch_row($res)) ;
...
?>
```

Ново: статистики

mysqli

Mysql Support	enabled
Client API library version	mysqlnd 5.0.1-beta - 070402 - \$Revision: 341 \$
Client statistics	
bytes_sent	44495
bytes_received	69654
packets_sent	583
packets_received	1932
protocol_overhead_in	7728
protocol_overhead_out	2332
result_set_queries	145
non_result_set_queries	135
no_index_used	0
bad_index_used	0
buffered_sets	145
unbuffered_sets	0
ps_buffered_sets	0
ps_unbuffered_sets	0
flushed_normal_sets	0
flushed_ps_sets	0
rows_fetched_from_server	503
rows_fetched_from_client	503
rows_skipped	0
copy_on_write_saved	2002
copy_on_write_performed	194
command_buffer_too_small	0
connect_success	101
connect_failure	0
connection_reused	0
explicit_close	44
implicit_close	57
disconnect_close	0

Повече информация :

- повече от 40 показателя
- улеснява оценката на тесни места в система
- лесно наблюдение на приложения

Лесно се вижда дали има скриптове, които:

- поискват повече редове, отколкото реално използват
- отварят повече връзки, отколкото е нужно..

Достъпни чрез:

- `phpinfo()`
- `mysqli_get_client_stats()`
- `mysqli_get_connection_stats()`
- `mysqli_get_cache_stats()`

Ново: статистики (2)

Мрежов трафик

- Байтове изпратени и получени
- Пакети, на MySQL протокола, изпратени и получени
- Излишък на протокола, в байтове

SQL свързани

- Заявки връщащи данни (SELECT)
- Други заявки
- Брой заявки неизползвали индекс
- Брой заявки използвали неподходящ индекс

Заявки връщащи резултат

- Брой на буферирани и небуферирани заявки
- Брой на заявки, при които не всички данни са били консумирани от скрипта. **Лошо ако е по-голям от 0**
- Брой редове изпратени от сървъра
- Брой редове консумирани от скрипта. **Лошо ако е по-голям от 0.**
- Неконсумирани редове. Разликата на горните
- Сору on write спестени и изпълнени

Заявки връщащи резултат (2)

- Брой експлицитни free result (e.g. `mysqli_free_result()`) операции
- Брой имплицитни free_result операции, например на края на изпълненото на скрипта.

Относно връзките

- Брой на (не)осъщесвени връзки
- Брой на преизползваните връзки, при постоянни връзки
- Брой експлицитно затворени връзки, например чрез `mysqli_close()`
- Брой имплицитно затворени връзки

Подготвени заявки

- Брой на буферирани и небуферирани подготвени заявки
- Брой на недоконсумирани заявки.
- Брой на експлицитно освободени ПЗ, чрез `mysqli_stmt_close()`.
- Брой на имплицитно затворени ПЗ.

... и още, и още....

Достъп до статистиките

Каквото видяхте на предишния слайд е достъпно чрез:

- За процеса:
`mysql_get_client_stats()`
- За връзката:
`mysql_get_connections_stats(link mysql)`

`zval` кеша може да бъде следен, чрез:

- `mysql_get_cache_stats()`
- PUT добавяния и пропуски
- GET добавяния и пропуски
- Големина в елементи
- Брой на свободните позиции
- Брой на референциите към кеша.

Абракадабра: Производительност

Измерете производителността на вашето приложение!

Значително подобрене в `mysqlnd` не означава, че вашето приложение ще полети

- няма смисъл да се оптимизира нещо, което консумира 2%, до 1%. С изразходваните средства може да си купите нов сървър :)

Отличните резултати при микро измерванията може да се стопят бързо.

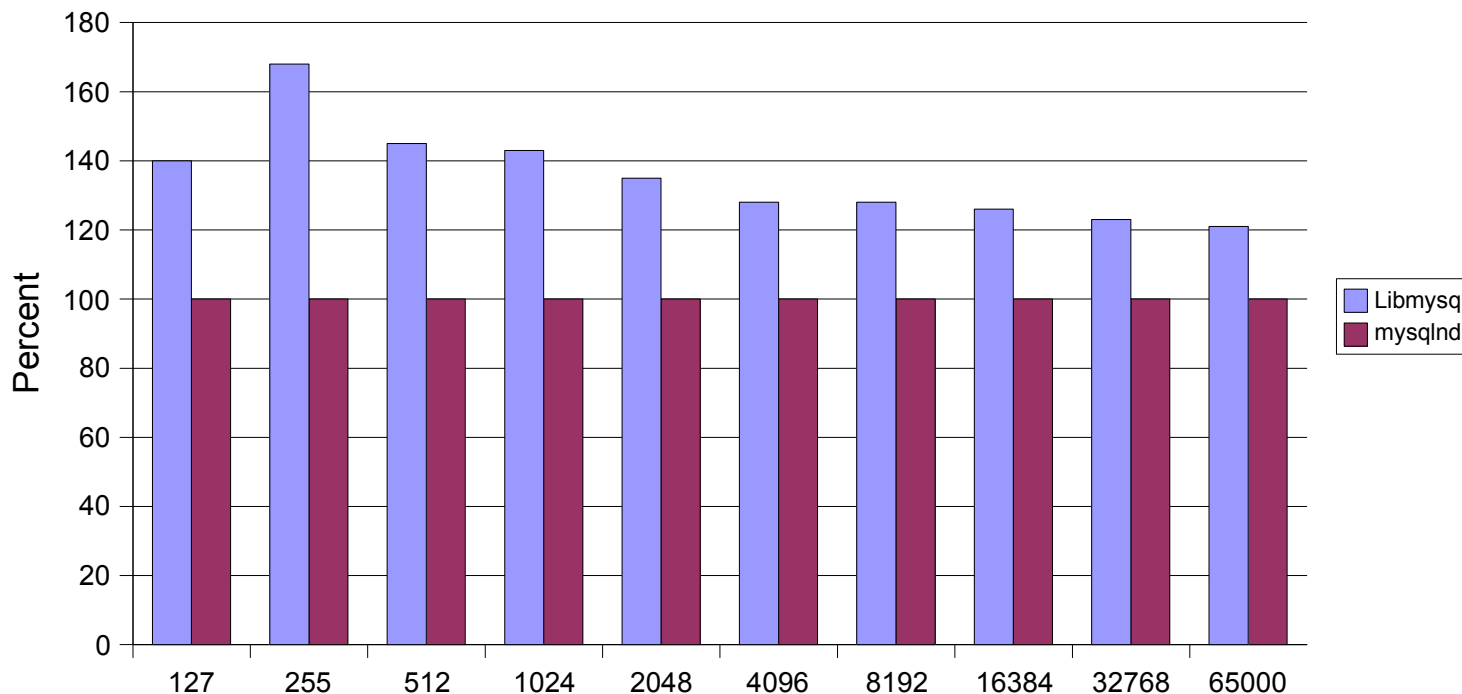
- Ако общото време на изпълнение е 10 сек. и времето в `libmysql` е само 0,1 сек.
- Дори `mysqlnd` да е двойно по-бърза => 0,05 сек.
- Подобренението е от 10s до 9.95s, само 0,5%!

Понякога по-бърза?

По-ниските стойности са по-добри, в случая са и базови.

Извличане на 100 реда, съдържащи само една колона `varchar` column of a variable size.

`mysql_select_varchar_buffered.php`

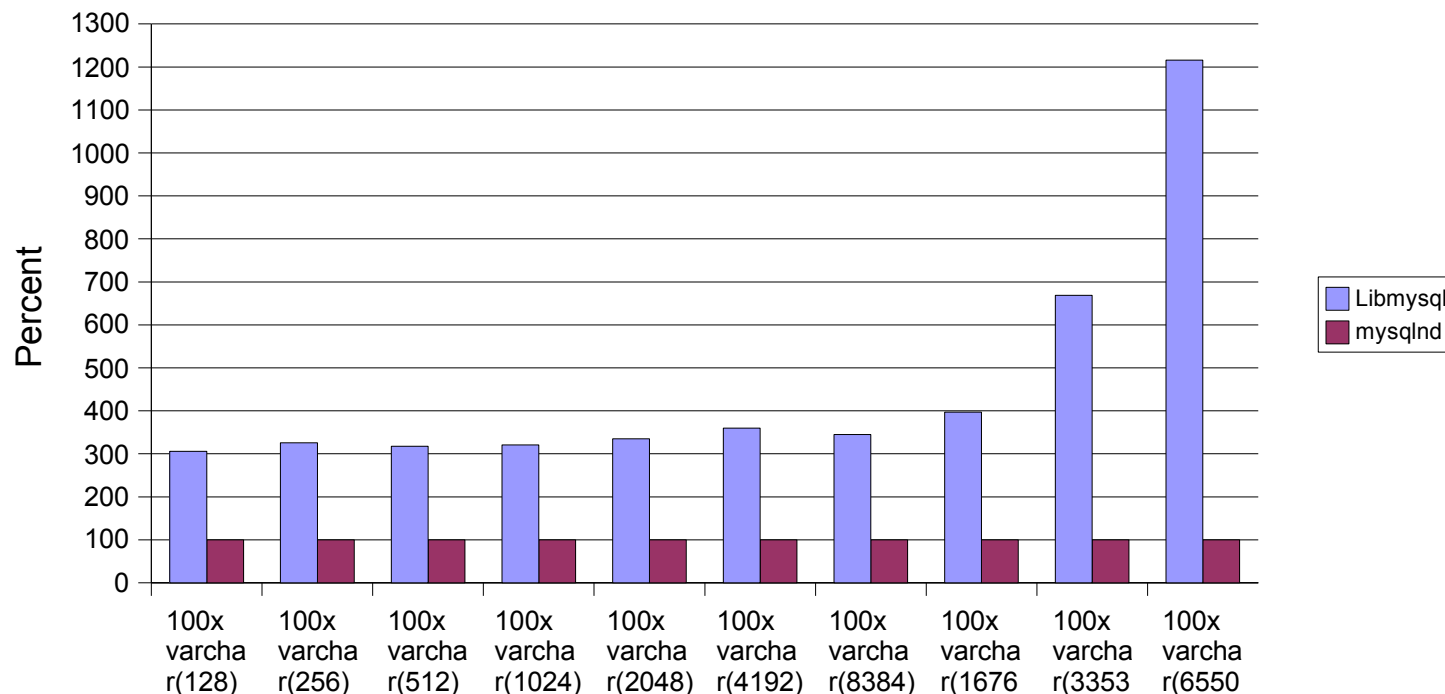


Ние харесваме тази графика :)

По-ниските стойности са по-добри!

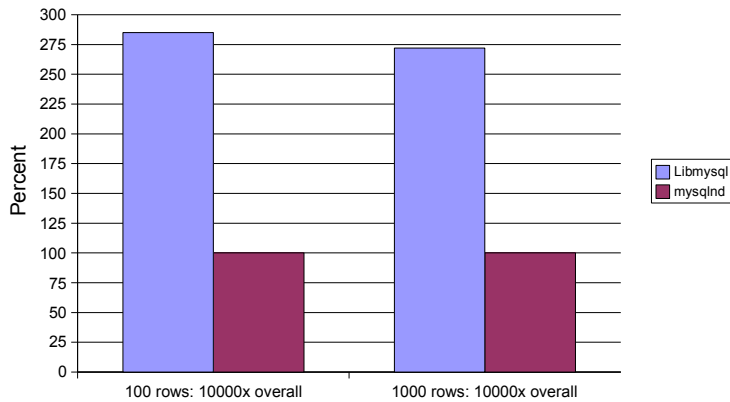
Закачане, заявка, извличане на 100 реда с една колона varchar(m), 1000 пъти `mysql_data_seek(mt_rand())`, затваряне на връзката.

`mysql_data_seek_sequential.php` (1000 rows, overall)

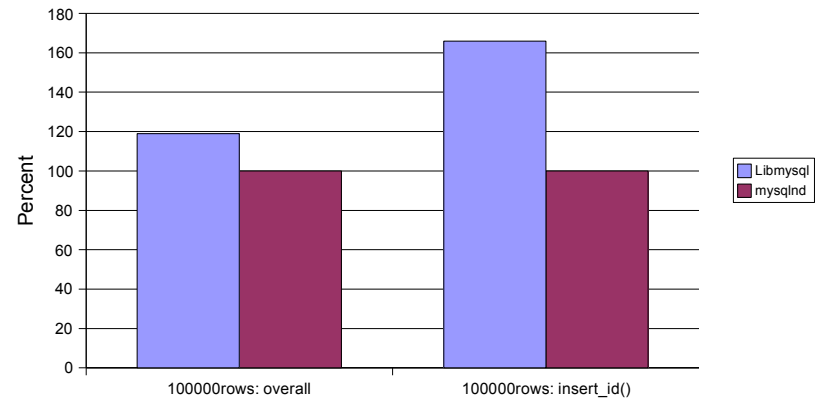


Още...

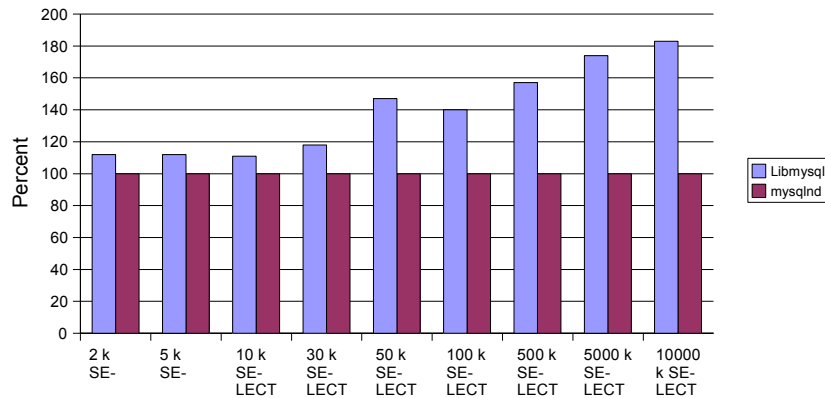
mysqli_affected_rows.php



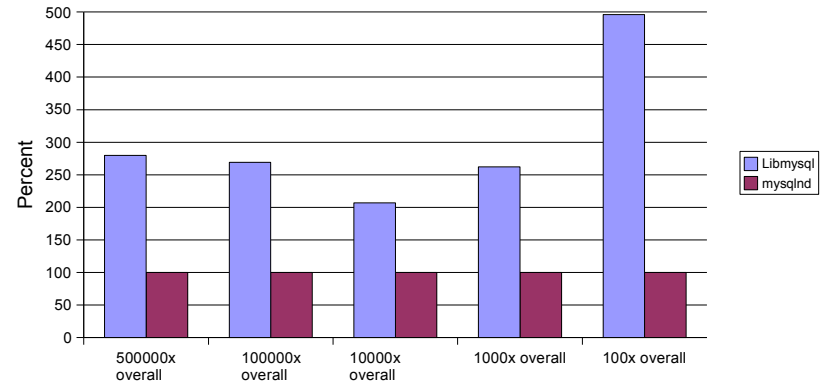
mysqli_insert_id.php



mysqli_fetch_long_buffered.php (10 rows, overall)



mysqli_real_query.php



Но....

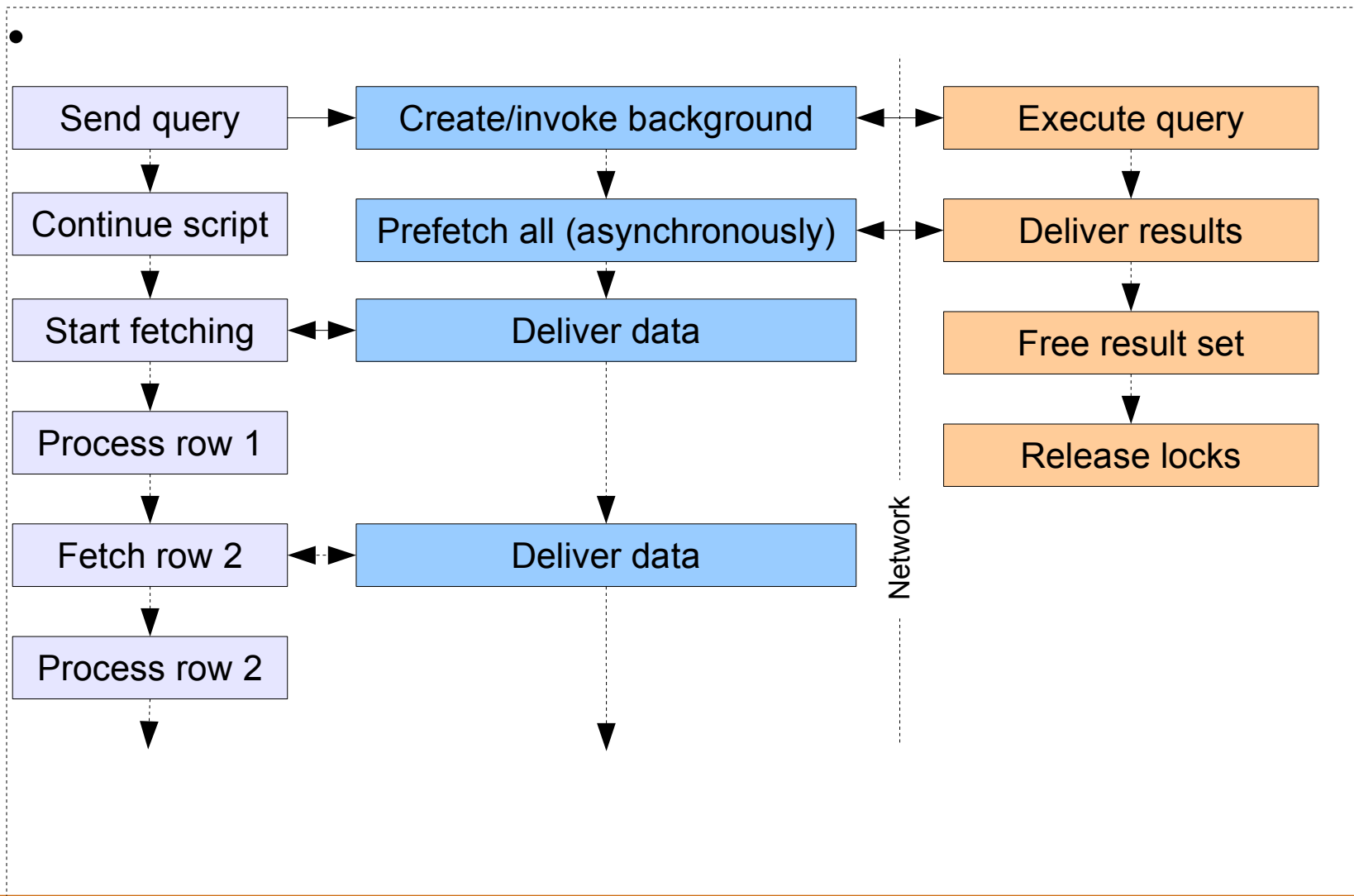
- ... това са микроизмервания
- ... запомнете, че отражението върху реални скриптове ще е различно!
- ... от 0.0001s на 0.00003s най-често няма значение
- ... mysqlnd е по-бърз, графики
проверете предварително
двойна производителност
превключване
- ... документ от 30 страници с повече графики ще се появи в скоро време.

Обикновено виждаме:
-5 до +15%.
Но няма гаранция!

Бъдещето

- Има още какво да се довършва
 - SSL поддръжка
 - Поддръжка на компресирания протокол
 - повече документация!
- Преместване на разработката в хранилището на php.net

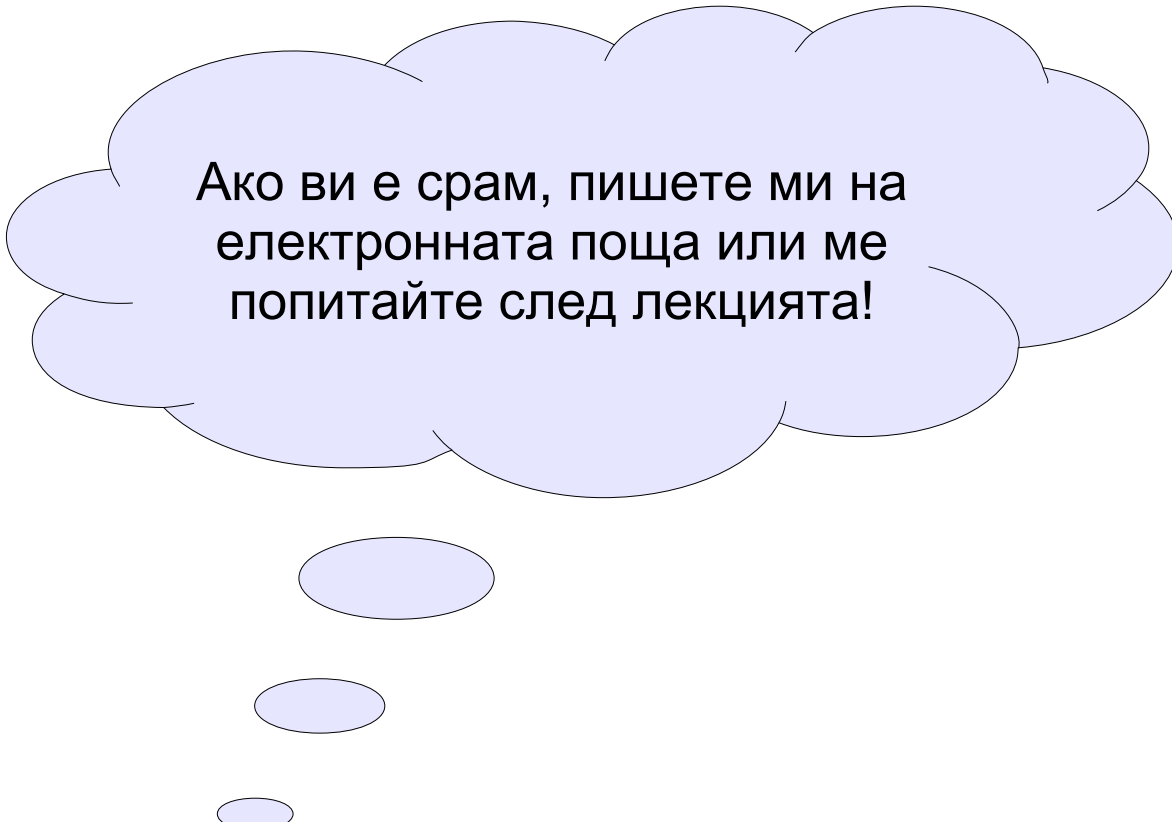
С какво експериментираме



Кой е виновен за всичко това?

- Георг Рихтер (Georg Richter)
 - Мениджмънт и разработка
 - georg@mysql.com
- Андрей Христов
 - Архитектура и разработка
 - Ако желаете още по-задълбочена информация, попитайте!
 - andrey@mysql.com
- Улф Вендел (Ulf Wendel)
 - QA, Documentation
 - Извършил и върши мръсната работа по опитите за счупване на mysqlnd.
 - uwendel@mysql.com

Въпроси ?



Ако ви е срам, пишете ми на
електронната поща или ме
попитайте след лекцията!