

# **Automatic Web Requests Clustering for Latency Analysis**

**Naser Ezzati**  
**Polytechnique Montreal**

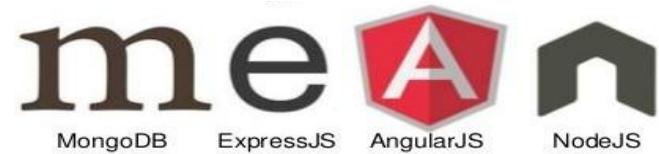
Confoo 2018  
Montreal, Canada

# Biography

- **Naser Ezzati**

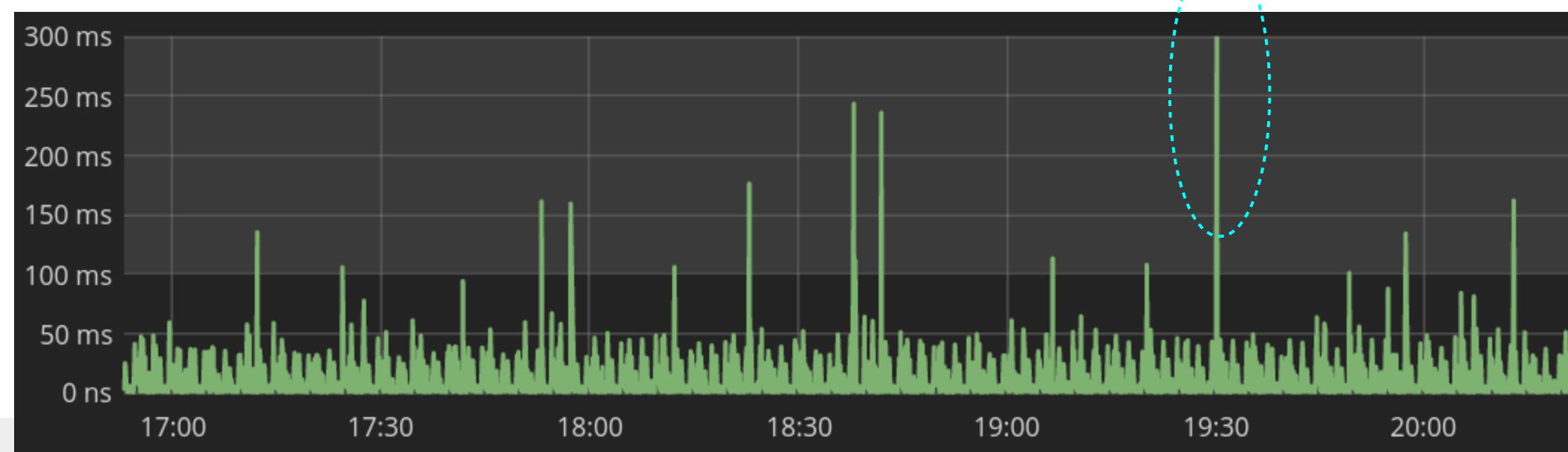
- Researcher at Polytechnique Montreal
- Works on
  - Performance Analysis
  - Automated Analysis
    - Machine Learning
    - Data Clustering

**LAMP:**



# Motivations

- **Challenges: Root Cause Analysis of Web Application Problems**
  - Slow application debugging:



# Debugging!

- **Challenges: root cause analysis of web application problems**
  - Slow application debugging:
    - Always
      - Bad design! (Code, db, configs, etc.)
      - Logging (echo, print, etc.)
    - Sometimes!
      - Lack of resources
      - Contention
  - Several components and layers are involved
    - Web server problem?
    - Code problem?
    - Bad database design? No table indexes?
    - System resource limitation?
      - VM, Container
  - Various debugging tools
  - Unified way
    - Trace-based approach
    - LAMP stack
    - MEAN stack

**LAMP:**



# Tracing

- **Tracing is a form of logging:**
  - It consists in recording events that happen in the system
    - Application or kernel
      - Tracepoints
        - Dynamic vs Statics
      - Includes a timestamp and a payload
    - Records much lower-level events
      - Occur much more frequently
      - Fast
      - Low overhead ( $\sim 100$  ns)
      - Multiple levels
        - Important!

# Tracing Levels

- **Kernel Level**

- Tracepoints in the kernel mainline
- Install a tracer, enable tracepoints and collect info
- Low-level data
  - Os level knowledge
  - Huge!
  - Useful but ... !

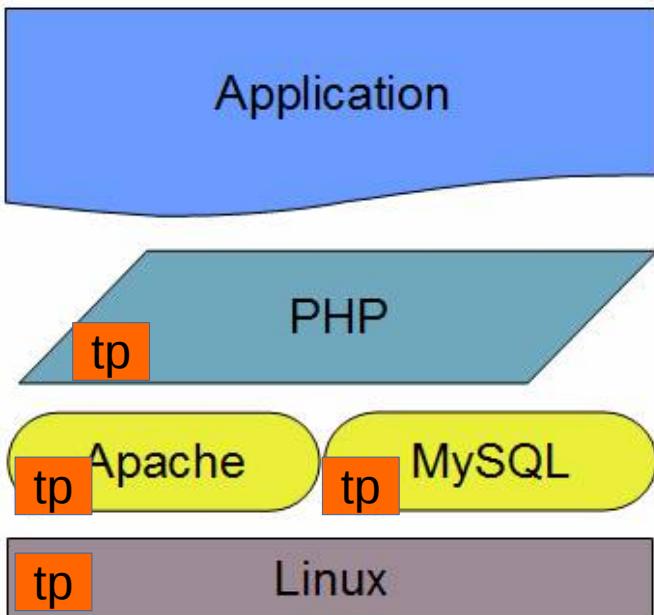
- **Application Level**

- Instrument your application
- High-speed & flexible logging
- Some open source applications are instrumented (lamp, node.js, qemu, etc.)
- High-level data about application
  - Useful but ... !

# Userspace tracing

- You can trace your application

- tracepoints
  - LTTng-UST
  - FTrace



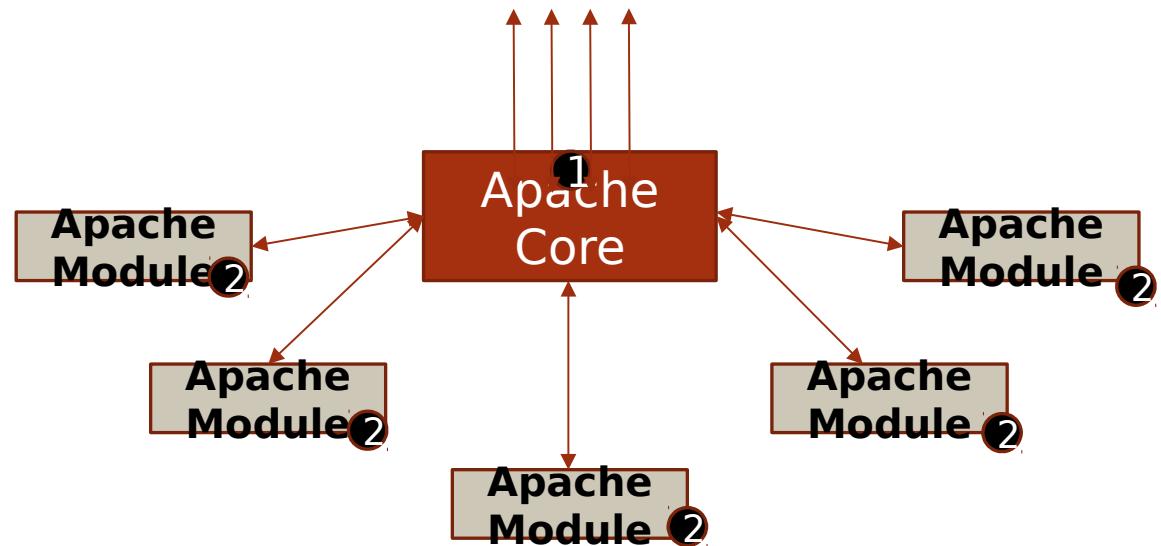
```
void function(void)
{
    int i = 0;
    long vals[3] = { 0x42, 0xCC, 0xC001CAFE };
    float flt = M_PI;

    [...]
    tracepoint(ust_tests_hello,
               tptest,
               i,
               &vals,
               flt);
    [...]
}
```

# 1- Apache

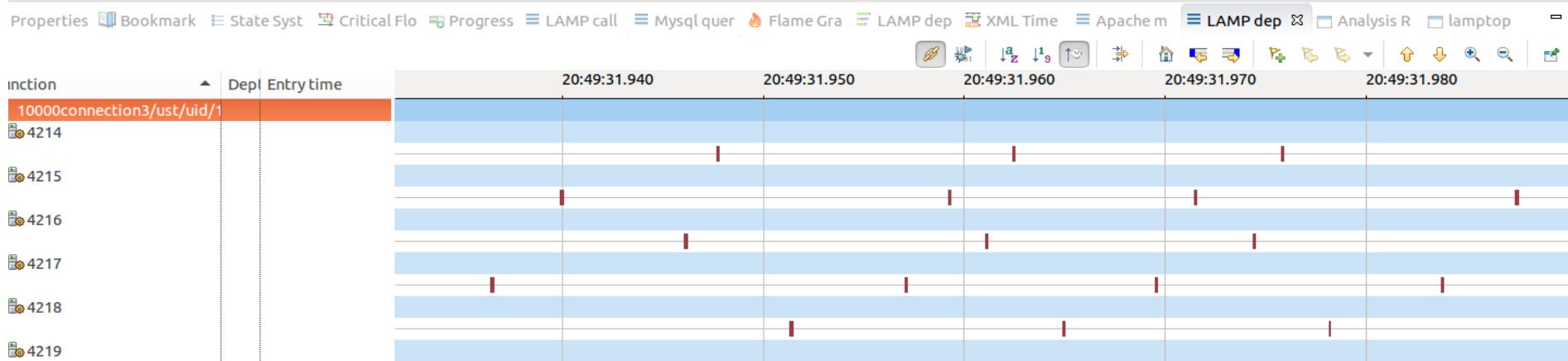
- **Apache LTTng module:**

- Hooks LTTng probes into the Apache web server.
- These probes extract runtime information about the web requests and the apache itself
  - Web requests
  - Apache internals



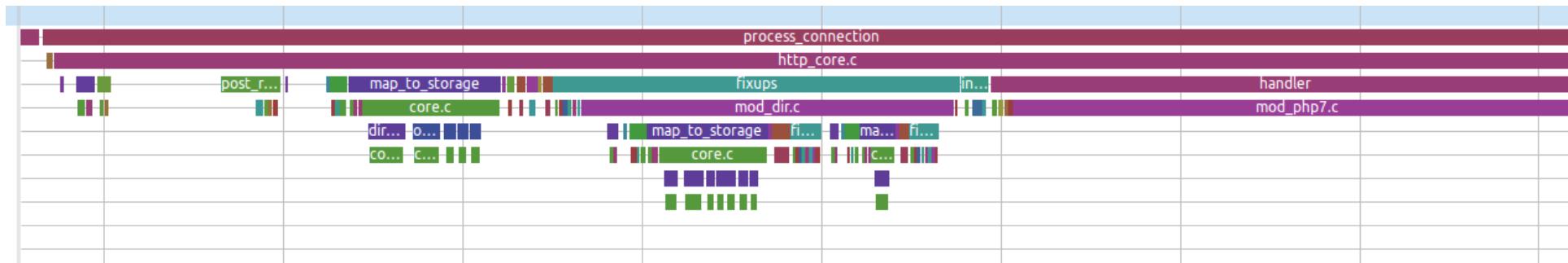
# Web Requests Tracing

Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	<srch>	<srch>
20:49:53.786 561 345	ss_3	3	ust_apache:close_connection	connection_id=16, context._vtid=4236
20:49:53.786 584 375	ss_3	3	ust_apache:suspend_connection	connection_id=16, context._vtid=4236
20:49:53.786 592 518	ss_6	6	ust_apache:accept_connection	connection_id=20, client_ip=132.207.72.9, client_hostname=(null), local_ip=132.207.72.37, local_hostname=(nu
20:49:53.786 596 784	ss_3	3	ust_apache:accept_connection	connection_id=16, client_ip=132.207.72.9, client_hostname=(null), local_ip=132.207.72.37, local_hostname=(nu
20:49:53.786 630 594	ss_3	3	ust_apache:request_entry	connection_id=16, client_ip=132.207.72.9, method=GET, uri=/drupal, protocol=HTTP/1.0, request_info= reques
20:49:53.786 640 316	ss_6	6	ust_apache:request_entry	connection_id=20, client_ip=132.207.72.9, method=GET, uri=/drupal, protocol=HTTP/1.0, request_info= reques
20:49:53.786 724 371	ss_3	3	ust_apache:request_exit	connection_id=16, status=301, context._vtid=4236
20:49:53.786 749 922	ss_6	6	ust_apache:request_exit	connection_id=20, status=301, context._vtid=4244
20:49:53.786 751 414	ss_3	3	ust_apache:close_connection	connection_id=16, context._vtid=4236
20:49:53.786 763 635	ss_3	3	ust_apache:suspend_connection	connection_id=16, context._vtid=4236
20:49:53.786 777 199	ss_3	3	ust_apache:accept_connection	connection_id=16, client_ip=132.207.72.9, client_hostname=(null), local_ip=132.207.72.37, local_hostname=(nu
20:49:53.786 781 806	ss_6	6	ust_apache:close_connection	connection_id=20, context._vtid=4244
20:49:53.786 798 980	ss_6	6	ust_apache:suspend_connection	connection_id=20, context._vtid=4244
20:49:53.786 809 775	ss_3	3	ust_apache:request_entry	connection_id=16, client_ip=132.207.72.9, method=GET, uri=/drupal, protocol=HTTP/1.0, request_info= reques
20:49:53.786 907 271	ss_3	3	ust_apache:request_exit	connection_id=16, status=301, context._vtid=4236
20:49:53.786 933 238	ss_3	3	ust_apache:close_connection	connection_id=16, context._vtid=4236
20:49:53.786 947 884	ss_3	3	ust_apache:suspend_connection	connection_id=16, context._vtid=4236
20:49:53.789 370 776	ss_3	3	ust_apache:accept_connection	connection_id=4, client_ip=132.207.72.9, client_hostname=(null), local_ip=132.207.72.37, local_hostname=(null
20:49:53.789 408 718	ss_3	3	ust_apache:request_entry	connection_id=4, client_ip=132.207.72.9, method=GET, uri=/drupal, protocol=HTTP/1.0, request_info= request:
20:49:53.789 502 508	ss_3	3	ust_apache:request_exit	connection_id=4, status=301, context._vtid=4218
20:49:53.789 528 073	ss_3	3	ust_apache:close_connection	connection_id=4, context._vtid=4218



# Apache Modules Tracing

Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	ust_apache:apache_module	<srch>
11:12:07.494 274 785	channel0_3	3	ust_phrequest_exit	path=/usr/local/apache2/htdocs/drupal/index.php, url=/drupal/index.php, m...
11:12:07.494 278 193	channel0_3	3	ust_apache:apache_module_complete	name=handler, src=mod_php7.c, result=0
11:12:07.494 278 607	channel0_3	3	ust_apache:apache_module_exit	name=handler, result=0
11:12:07.494 285 929	channel0_3	3	ust_apache:apache_module_complete	name=process_connection, src=http_core.c, result=0
11:12:07.494 286 217	channel0_3	3	ust_apache:apache_module_exit	name=process_connection, result=0
11:12:07.494 287 452	channel0_3	3	ust_apache:apache_module_entry	name=protocol_get
11:12:07.494 287 781	channel0_3	3	ust_apache:apache_module_exit	name=protocol_get, result=0
11:12:07.494 291 250	channel0_3	3	ust_apache:apache_module_entry	name=protocol_get
11:12:07.494 291 447	channel0_3	3	ust_apache:apache_module_exit	name=protocol_get, result=0
11:12:07.494 291 727	channel0_3	3	ust_apache:apache_module_entry	name=log_transaction
11:12:07.494 292 316	channel0_3	3	ust_apache:request_exit	id=0, status=200
11:12:07.494 292 703	channel0_3	3	ust_apache:apache_module_invoke	name=log_transaction, src=mod_log_config.c
11:12:07.494 312 568	channel0_3	3	ust_apache:apache_module_complete	name=log_transaction, src=mod_log_config.c, result=0
11:12:07.494 312 929	channel0_3	3	ust_apache:apache_module_exit	name=log_transaction, result=0
11:12:07.494 320 420	channel0_3	3	ust_apache:apache_module_entry	name=suspend_connection
11:12:07.494 320 695	channel0_3	3	ust_apache:apache_module_exit	name=suspend_connection, result=0
11:12:07.496 537 029	channel0_3	3	ust_apache:apache_module_entry	name=resume_connection
11:12:07.496 537 342	channel0_3	3	ust_apache:apache_module_exit	name=resume_connection, result=0
11:12:07.496 537 627	channel0_3	3	ust_apache:apache_module_entry	name=process_connection
11:12:07.496 538 076	channel0_3	3	ust_apache:apache_module_invoke	name=process_connection, src=mod_reqtimeout.c
11:12:07.496 538 598	channel0_3	3	ust_apache:apache_module_complete	name=process_connection, src=mod_reqtimeout.c, result=-1
11:12:07.496 538 849	channel0_3	3	ust_apache:apache_module_invoke	name=process_connection, src=http_core.c
11:12:07.496 539 461	channel0_3	3	ust_apache:apache_module_entry	name=protocol_get
11:12:07.496 539 721	channel0_3	3	ust_apache:apache_module_exit	name=protocol_get, result=0
11:12:07.496 541 165	channel0_3	3	ust_apache:apache_module_entry	name=create_request
11:12:07.496 541 497	channel0_3	3	ust_apache:apache_module_invoke	name=create_request, src=core.c
11:12:07.496 541 942	channel0_3	3	ust_apache:apache_module_complete	name=create_request, src=core.c, result=0



## 2- PHP

- **LTTng probes in PHP**
  - Provide detailed information about the PHP requests
- **Monitor the entire PHP script execution:**
  - **13 tracepoints**
    - Start/close a request
    - Function calls
    - Line executions
    - db connections
    - errors/exceptions
    - New PHP function: `trace_print`
  - **arguments**
    - request info, function name, file name, class name, line number, etc.
- **Trace Compass views:**
  - **CallStack, Flame Graph, Request lists/response time distribution**

# Tracepoints

Event	Description
request_entry	Fires when a request starts.
request_exit	Fires when a request exits.
compile_file_entry	Fires when a file compilation starts.
compile_file_exit	Fires when a file compilation ends.
function_entry	Fires when the PHP engine calls a function/method.
function_exit	Fires when the PHP engine returns from a function/method.
execute_entry	Fires when a line code is to be executed.
execute_exit	Fires after execution of a line code.
php_error_entry	Fires just before logging a PHP error
php_error_exit	Fires just after logging a PHP error
php_exception_thrown_entry	Fires just before logging a thrown PHP exception
php_exception_thrown_exit	Fires just after logging a thrown PHP exception
trace_print	A <b>PHP function</b> that you can call from your script to output a string in the trace

# LTTng PHP exetnsion

## Installation

You can build and install the LTTng extension from source which is straightforward:

```
git clone https://github.com/naser/LTTng-php-tracing-module.git
cd LTTng-php-tracing-module
phpize

/* Before configuration, make sure you have LTTng 2.X installed in your machine. For installation manual

./configure
make
sudo make install
```

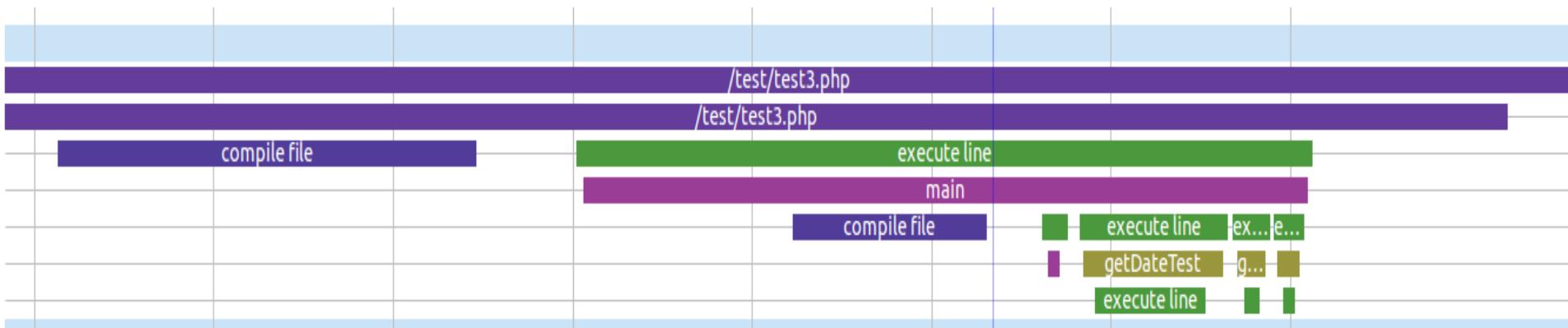
# Example

```
1 Hello
2 <p>Welcome!</p>
3 Today's date is:
4
5 <?php
6 require_once('include.php');
7 getDateTest();
8 getDateTest();
9 getDateTest();
10
11 ?>

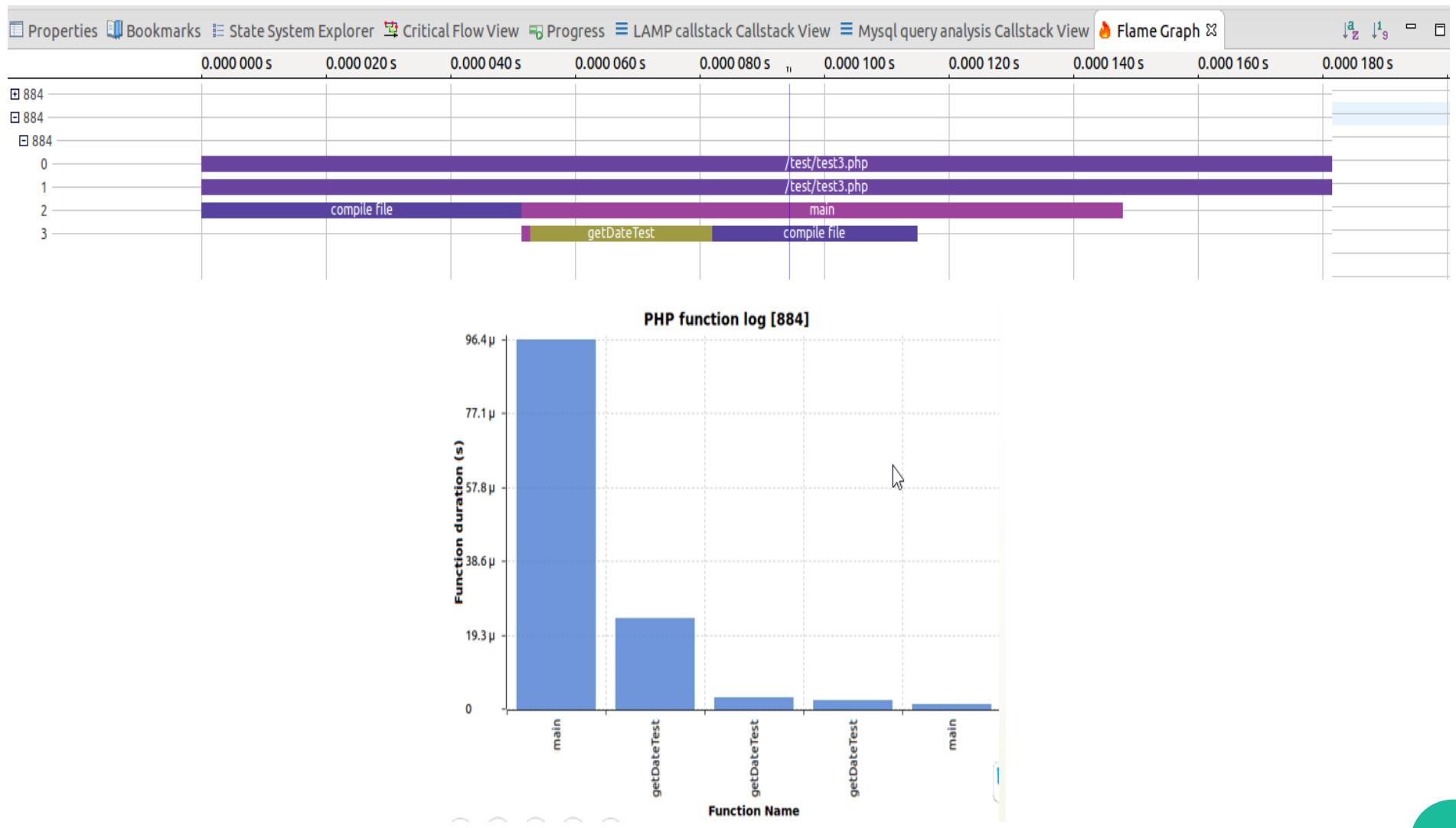
/usr/local/apache2/htdocs/test/include.php [FORMA
```

```
1 <?php
2
3 function getDateTest(){
4     echo date('m/d/Y') . '.1';
5 }
6
7 ?>

/usr/local/apache2/htdocs/test/include.php [FOR
```



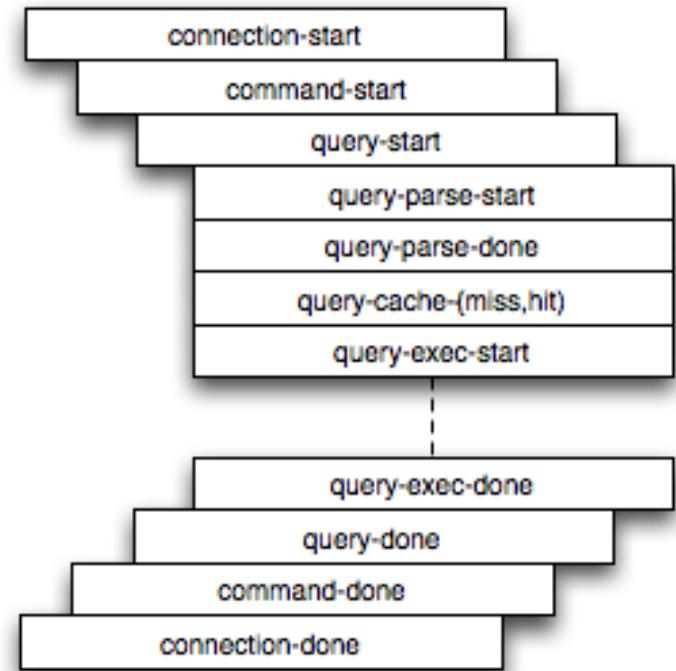
# Example (cntd)



# 3- MySQL/MariaDB

- **LTTng probes in MySQL/MariaDB**

- Provide information about query executions
- 60 tracepoints in 200 different locations
  - We hook to the existing probes.
- Monitor the full query execution process
  - DB connections
  - Query execution
    - Query type (select, update, insert, etc.)
    - Query parsing
    - Row-level operations in storage engines
    - Table R/W locks
    - File sorts
  - Cache miss, hit
  - Network I/O
  - More information in the arguments
    - Connection ID, DB name, user, host, etc



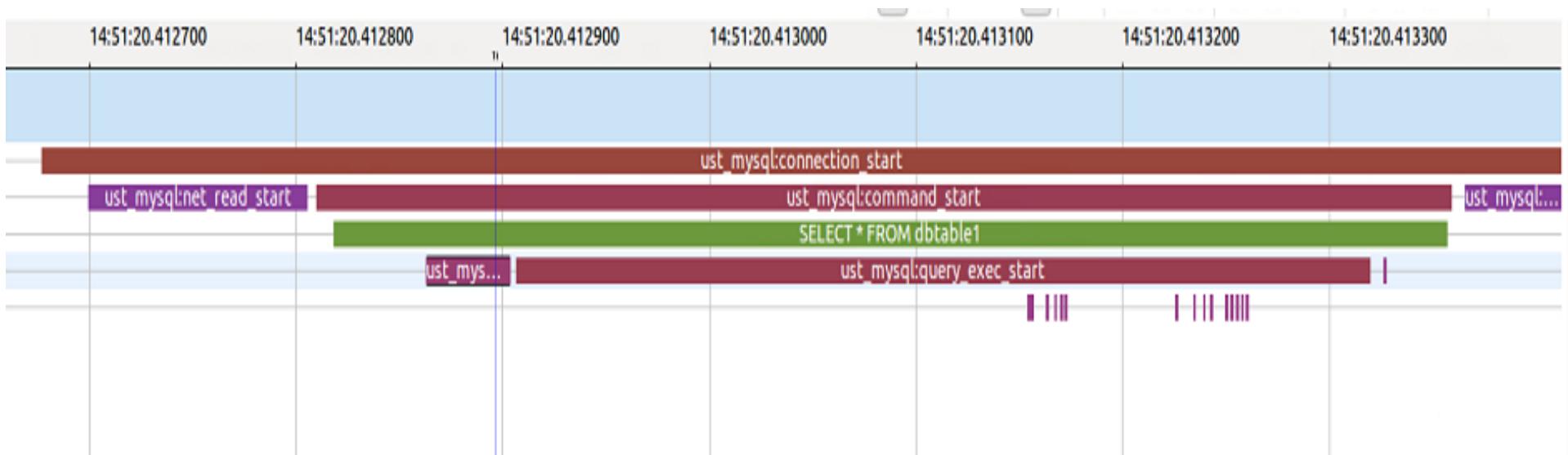
# Installtion

```
./configure --enable-Ittng  
cmake .  
make  
sudo make install
```

```
/home/naserez $> lttng list -u | grep "mysql:"  
ust_mysql:net_write_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:net_write_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:net_read_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:net_read_miss (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:net_read_hit (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:net_read_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:multi_delete_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:multi_delete_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:delete_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:delete_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:multi_update_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:multi_update_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:update_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:update_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:insert_select_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:insert_select_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:insert_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:insert_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:select_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:select_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:filesort_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:filesort_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:handler_unlock_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:handler_unlock_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:handler_wlock_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:handler_wlock_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:handler_rdlock_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:handler_rdlock_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:index_read_row_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:index_read_row_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:read_row_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:read_row_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:delete_row_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:delete_row_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:update_row_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:update_row_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:insert_row_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:insert_row_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:query_exec_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:query_exec_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:query_cache_miss (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:query_cache_hit (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:query_parse_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:query_parse_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:query_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:query_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:command_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:command_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:connection_done (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)  
ust_mysql:connection_start (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
```

# Example

Select \* from dbtable1;



## Example 2

```
select * from pfwp_statistics_visitor where id = 273;  
(no_cache) T: 407,792 ns
```

```
set global query_cache_size=2 * 1024 * 1024;
```

```
select * from pfwp_statistics_visitor where id = 273;  
(cache_miss) T: 408,858 ns
```

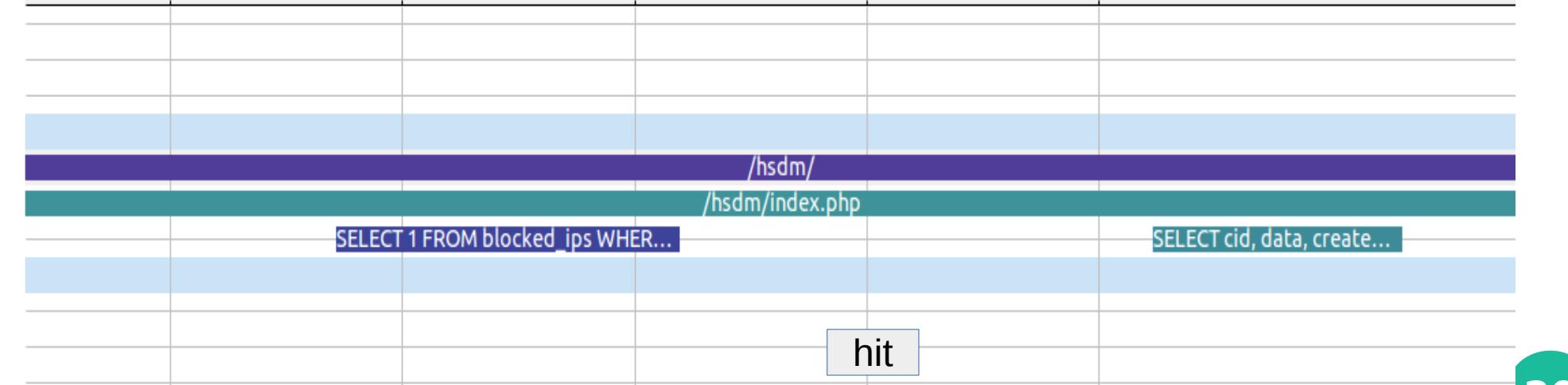
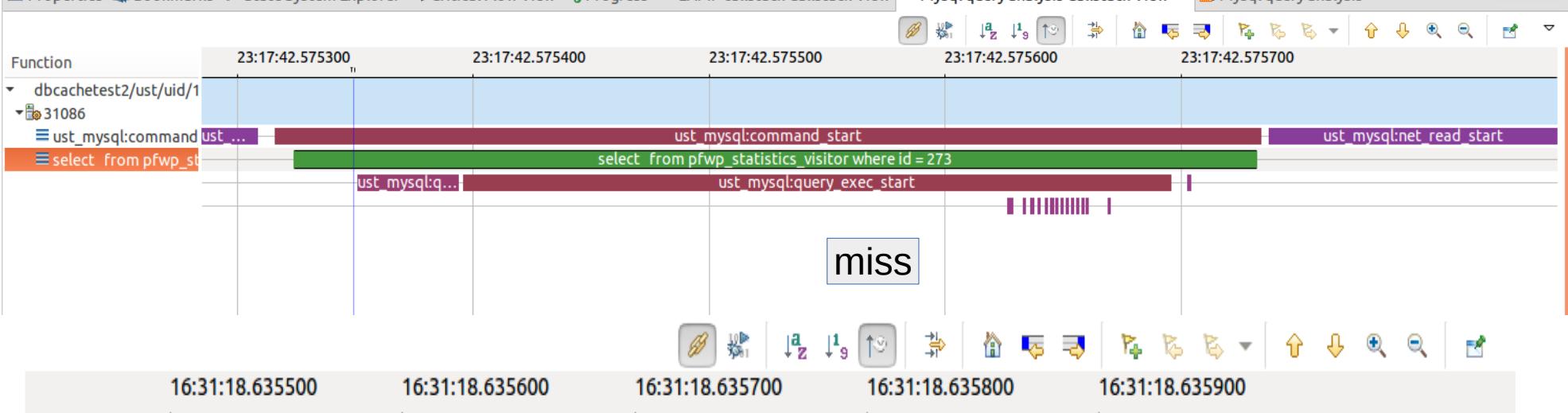
```
select * from pfwp_statistics_visitor where id = 273;  
(cache_hit) T: 42,708 ns
```



## dbcachetest2/ust/uid/1000/64-bit

Timestamp	Channel	CPU	Event type	Contents
23:17:39.369 848 397	<srch>	<srch>	<srch> <srch>	<srch>
23:17:39.369 856 327	ss_2	2	ust_mysql:command_done	result=0, thread_id=17787, context._vtid=31086
23:17:42.575 308 310	ss_2	2	ust_mysql:net_read_start	context._vtid=31086
23:17:42.575 315 827	ss_0	0	ust_mysql:net_read_done	result=0, len=53, context._vtid=31086
23:17:42.575 324 029	ss_0	0	ust_mysql:command_start	thread_id=17787, command=3, priv_user=root, host_or_ip=localhost, context._vtid=31086
23:17:42.575 349 057	ss_0	0	ust_mysql:query_start	query=select * from pfwp_statistics_visitor where id = 273, thread_id=17787, db=my_wiki, priv_user=root
23:17:42.575 351 159	ss_0	0	ust_mysql:query_cache_miss	query=select * from pfwp_statistics_visitor where id = 273, context._vtid=31086
23:17:42.575 393 433	ss_0	0	ust_mysql:query_parse_start	query=select * from pfwp_statistics_visitor where id = 273, context._vtid=31086
23:17:42.575 395 746	ss_0	0	ust_mysql:query_parse_done	result=0, context._vtid=31086
23:17:42.575 433 718	ss_0	0	ust_mysql:query_exec_start	query=select * from pfwp_statistics_visitor where id = 273, thread_id=17787, db=my_wiki, priv_user=root
			ust_mysql:handler_rdlock_start	db=my_wiki, table_name=pfwp_statistics_visitor, context._vtid=31086

Properties Bookmarks State System Explorer Critical Flow View Progress LAMP callstack Callstack View Mysql query analysis Callstack View Mysql query analysis

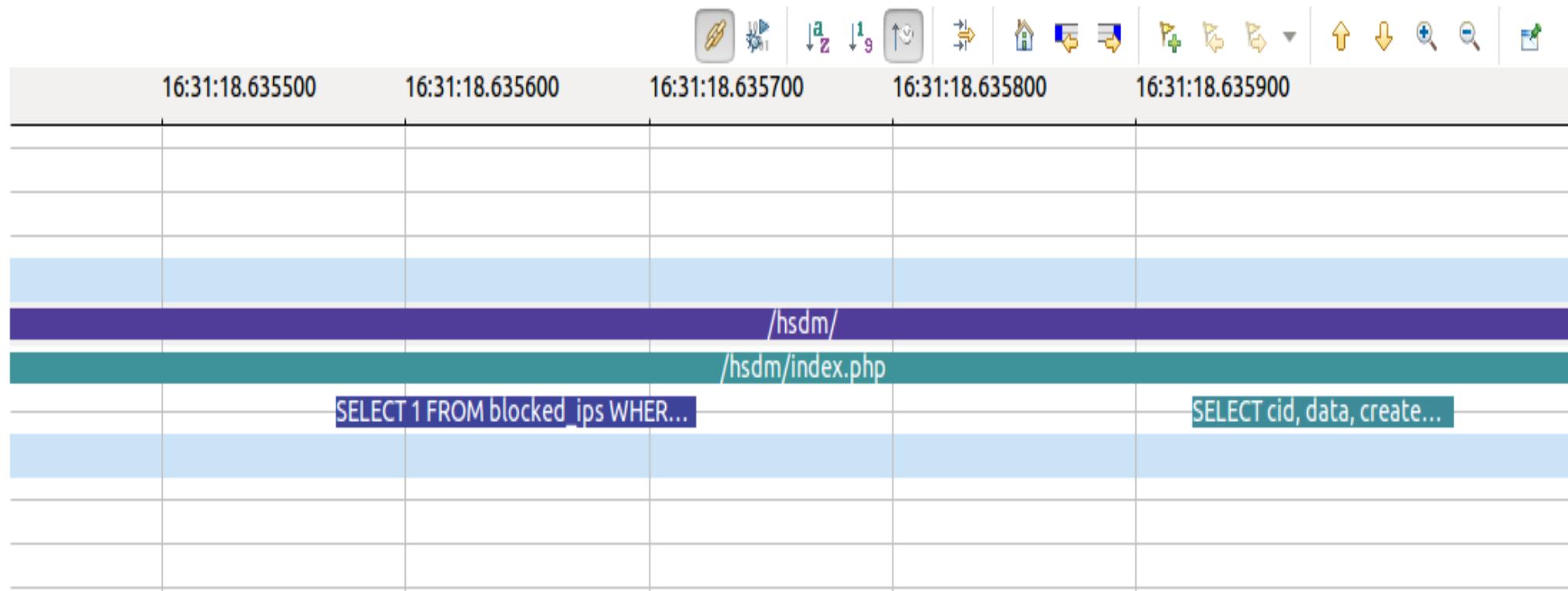


1

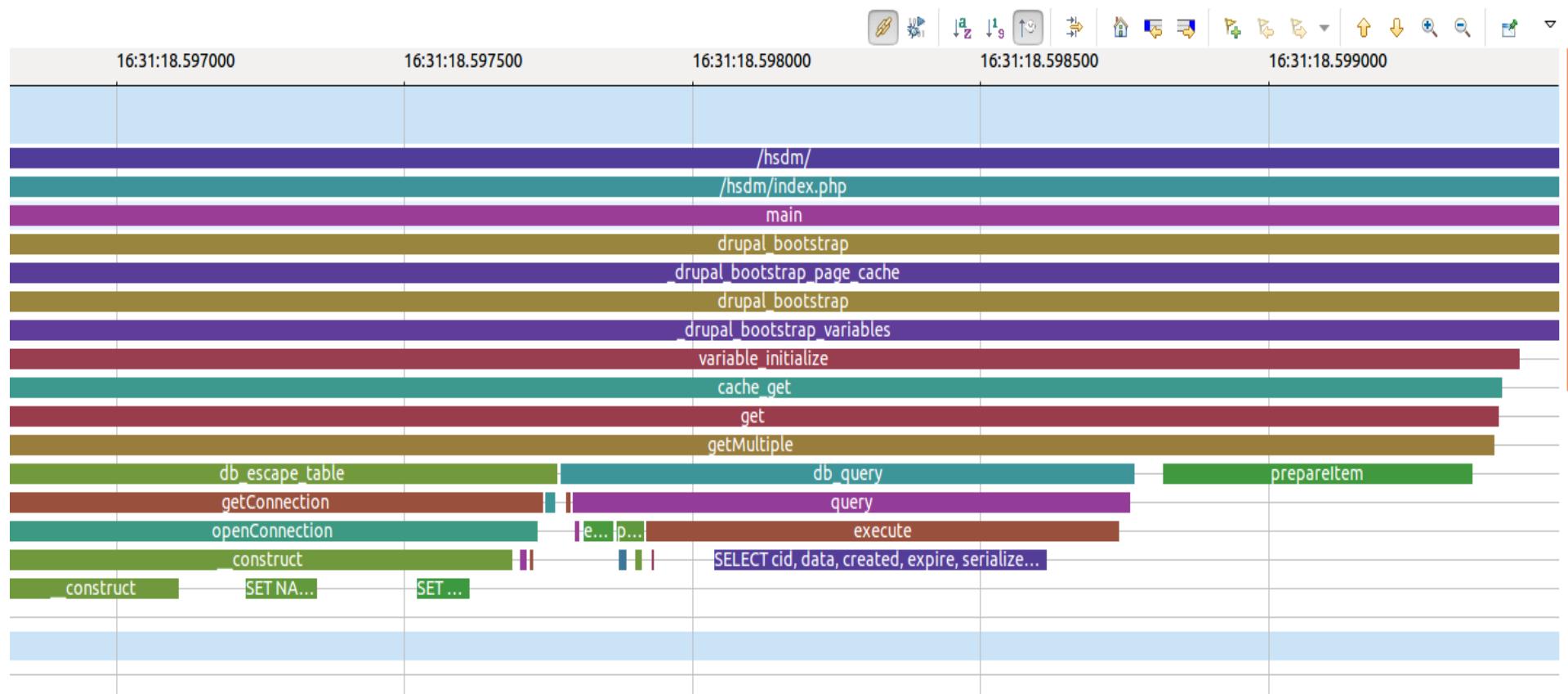
2

20

# 4- LAMP stack analysis

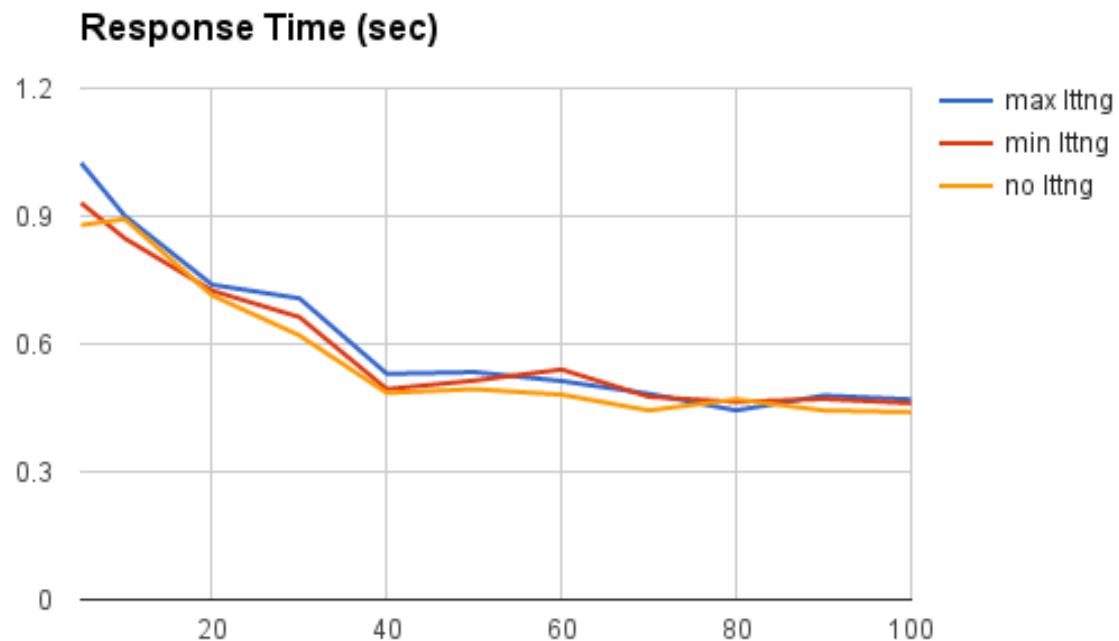


# Flame chart



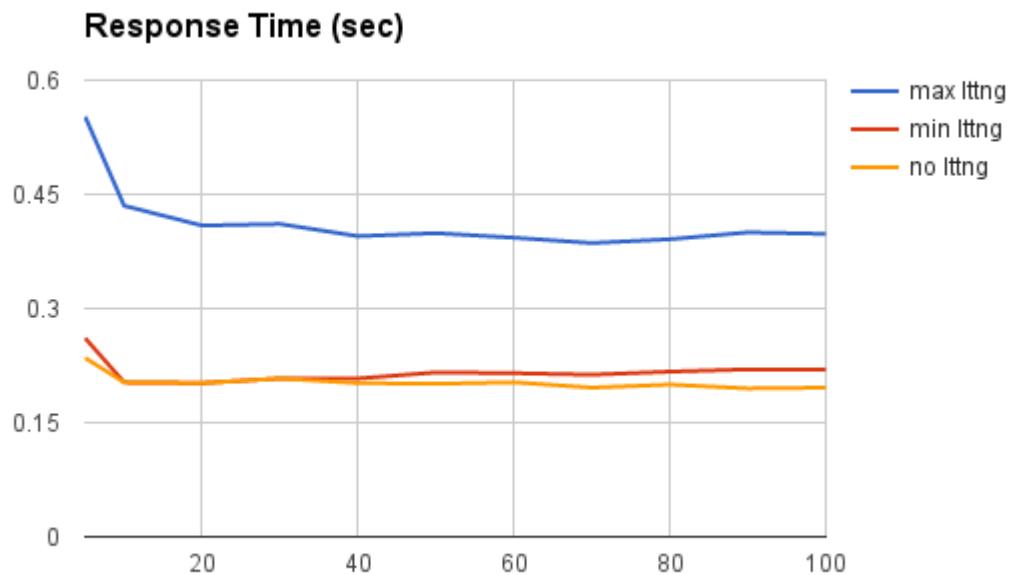
# Performance

- **ab -c [5-100] -n 5000 <http://132.207.72.37/drupal>**
- **28000 lines of code**



# Performance (the worst case)

- **ab -c [5-100] -n 1000 http://32.207.72.37/test/bench.php**
- **65,000,000 lines of code**

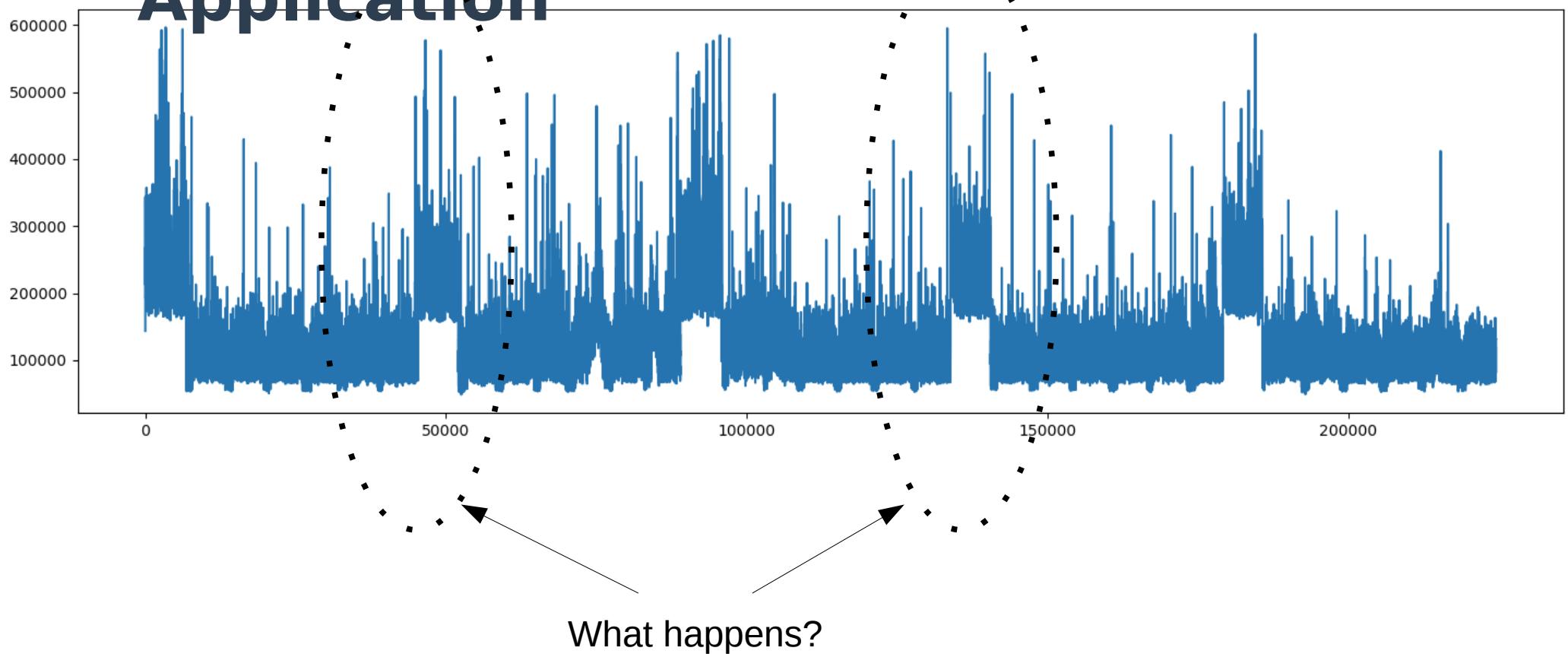


# Use-case 1:

## PHP OP-CACHE

### Performance Analysis

# Response Time of a Web Application



# PHP Request Anatomy

- **PHP Is a scripting language**

- compiles any file you ask it to run, obtain OPCodes from compilation, run them, and trash them away immediately.

- Parse, compile, execute, forget

Parse, compile, execute, forget

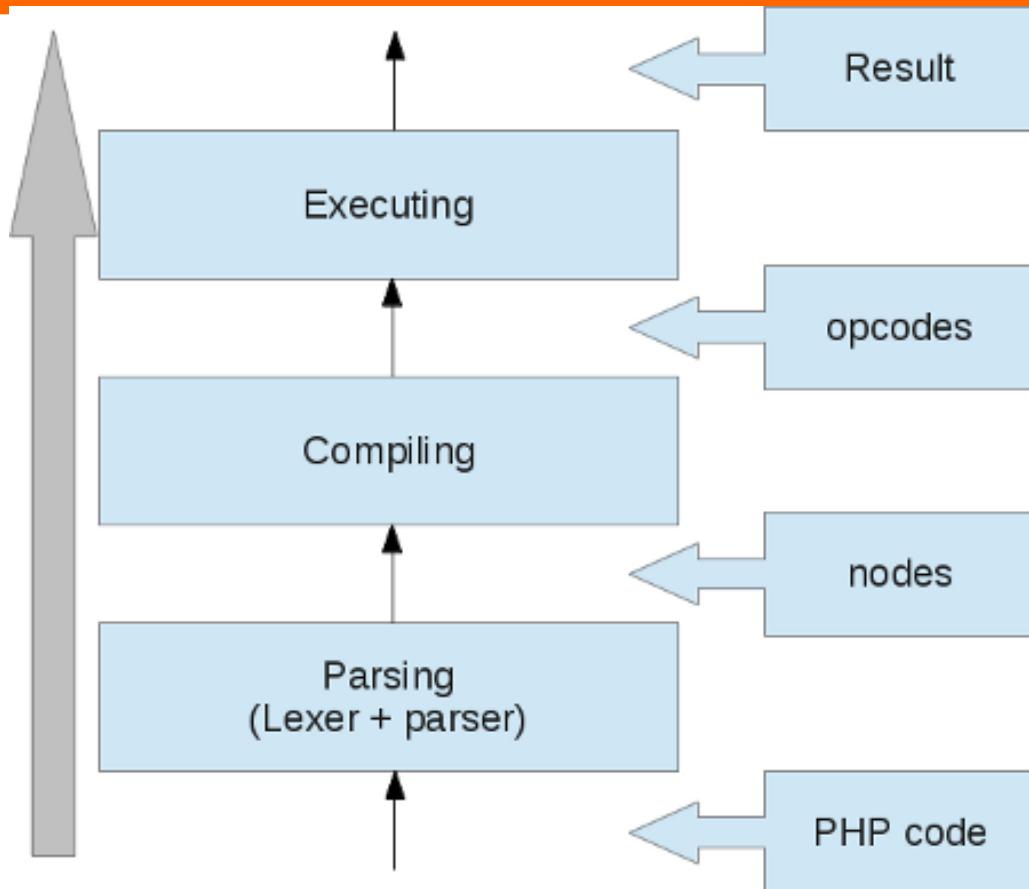
Parse, compile, execute, forget

...

**PHP "forgets" everything it's done in request**

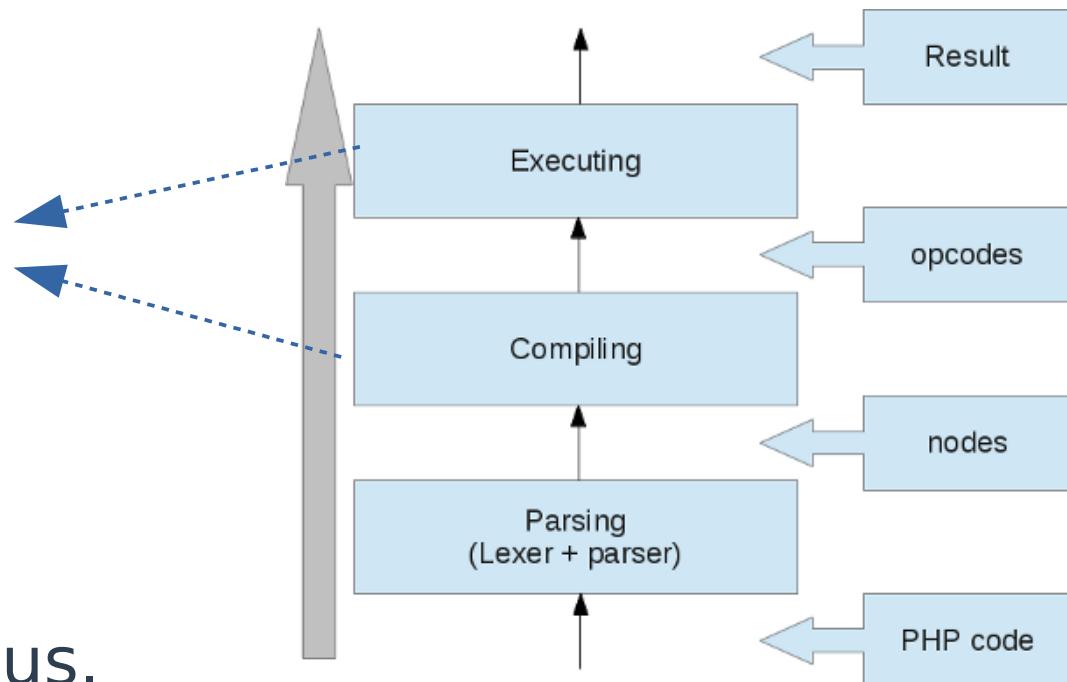
**N-1, when it comes to run request N.**

- Even if it calls the same scripts several times.

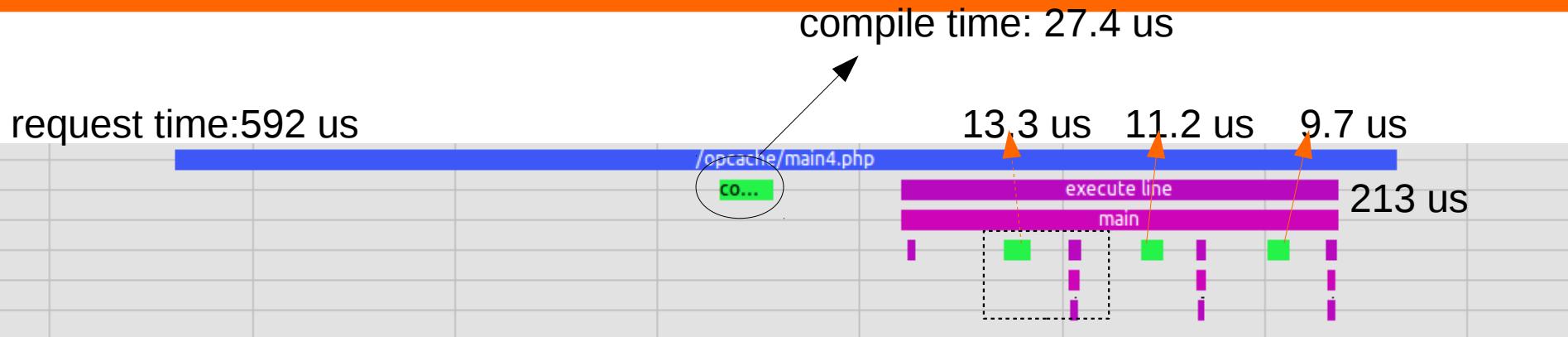


# PHP Request Anatomy (2)

- Which one is the longest?
  - It depends!
  - Let's see what trace data gives us.

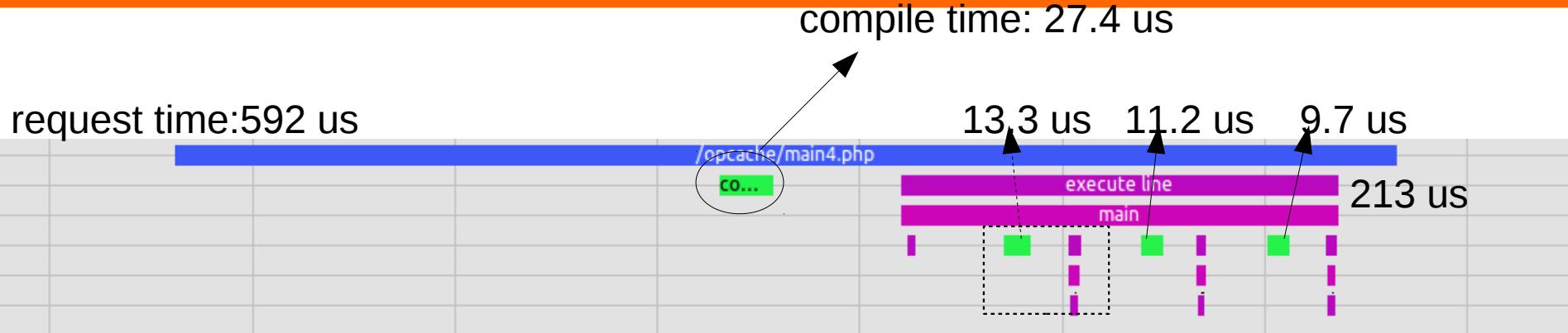


# Compile Time Analysis



```
1 <?php /*main4.php*/
2
3 $x = rand(0,1000);
4
5 echo $x.PHP_EOL;
6 $xy = 123;
7 include 'folder1/'.$xy.'.php';
8 include 'folder2/'.$xy.'.php';
9 include 'folder3/'.$xy.'.php';
10
11 echo $x.PHP_EOL;
12 ?>
```

# Compile Time: UST Events



```
1 <?php /*main4.php*/
2
3 $x = rand(0,1000);
4
5 echo $x.PHP_EOL;
6 $xy = 123;
7 include 'folder1/'.$xy.'.php';
8 include 'folder2/'.$xy.'.php';
9 include 'folder3/'.$xy.'.php';
10
11 echo $x.PHP_EOL;
12 ?>
```

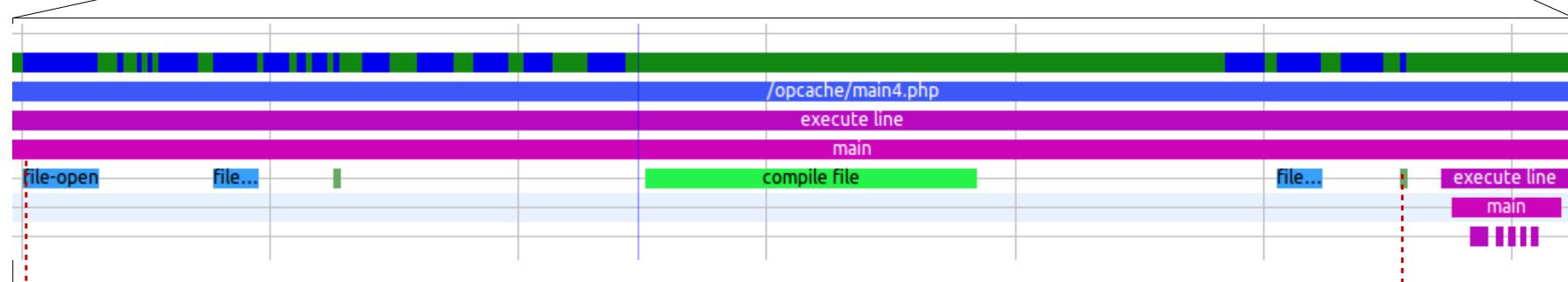
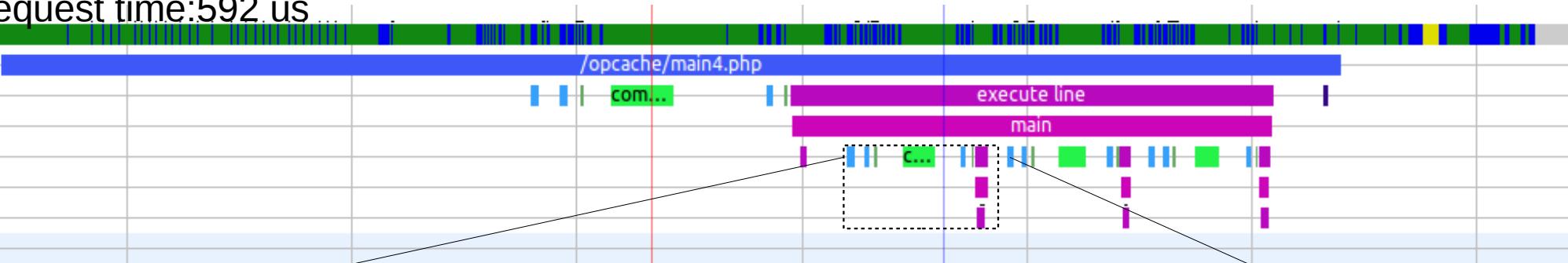
Compilation time: 60us (~ 10 % of the request time)

But, let's go deeper!

# Kernel + UST Events

Now looks much more!

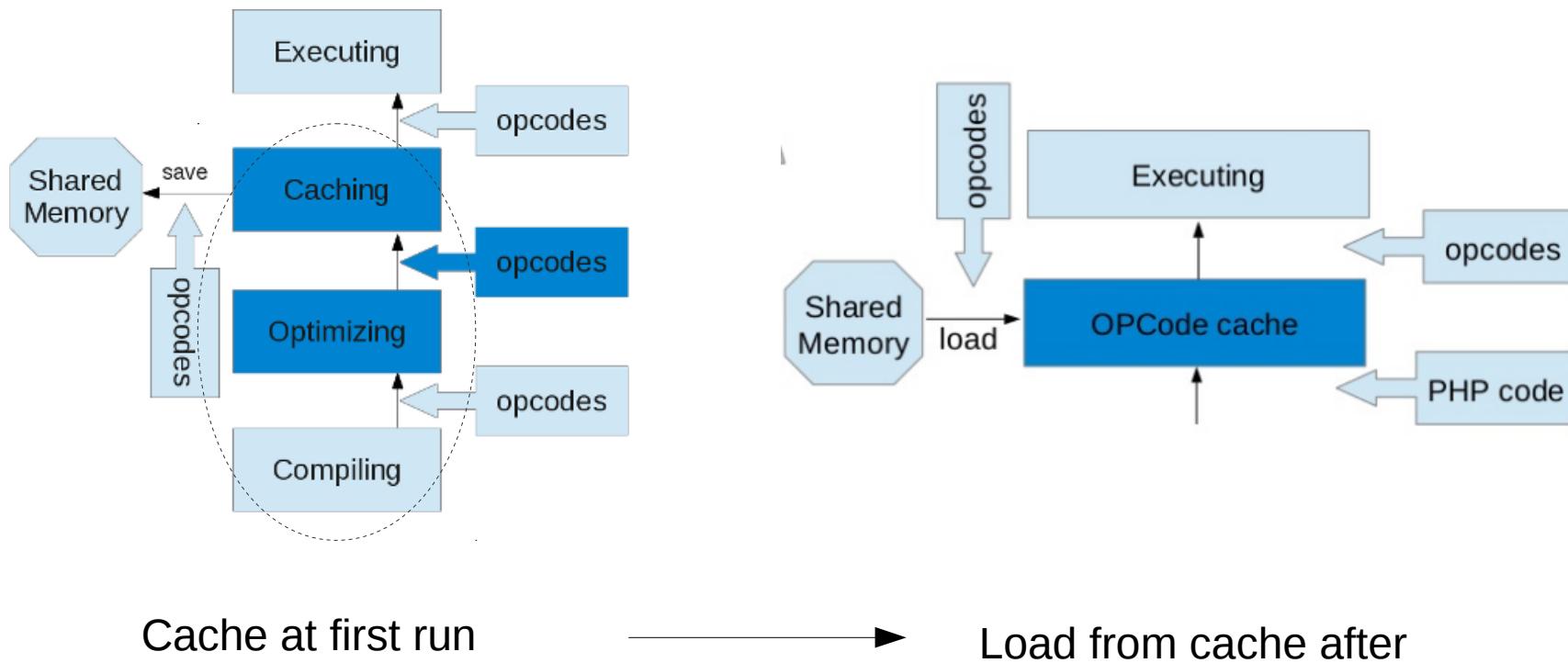
request time:592.us



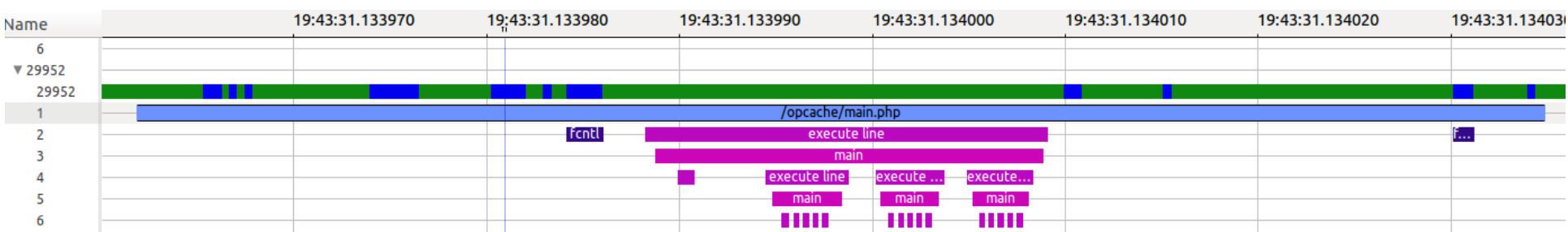
Parse, Compile, Optimize (55us)

Execute(5.2us)

# Solution: Opcode Cache (Opcache)



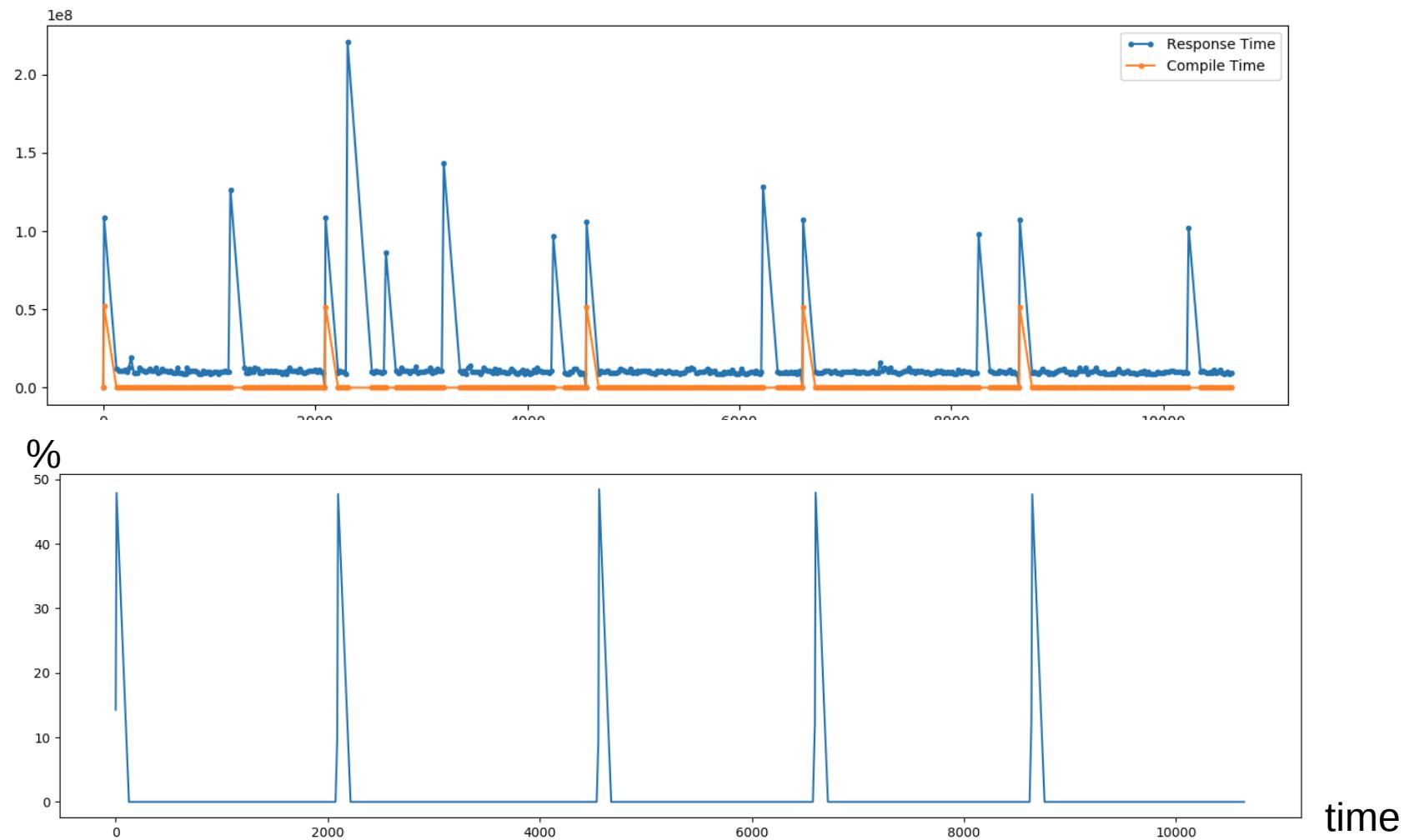
# Solution: Opcode Cache (Opcache)



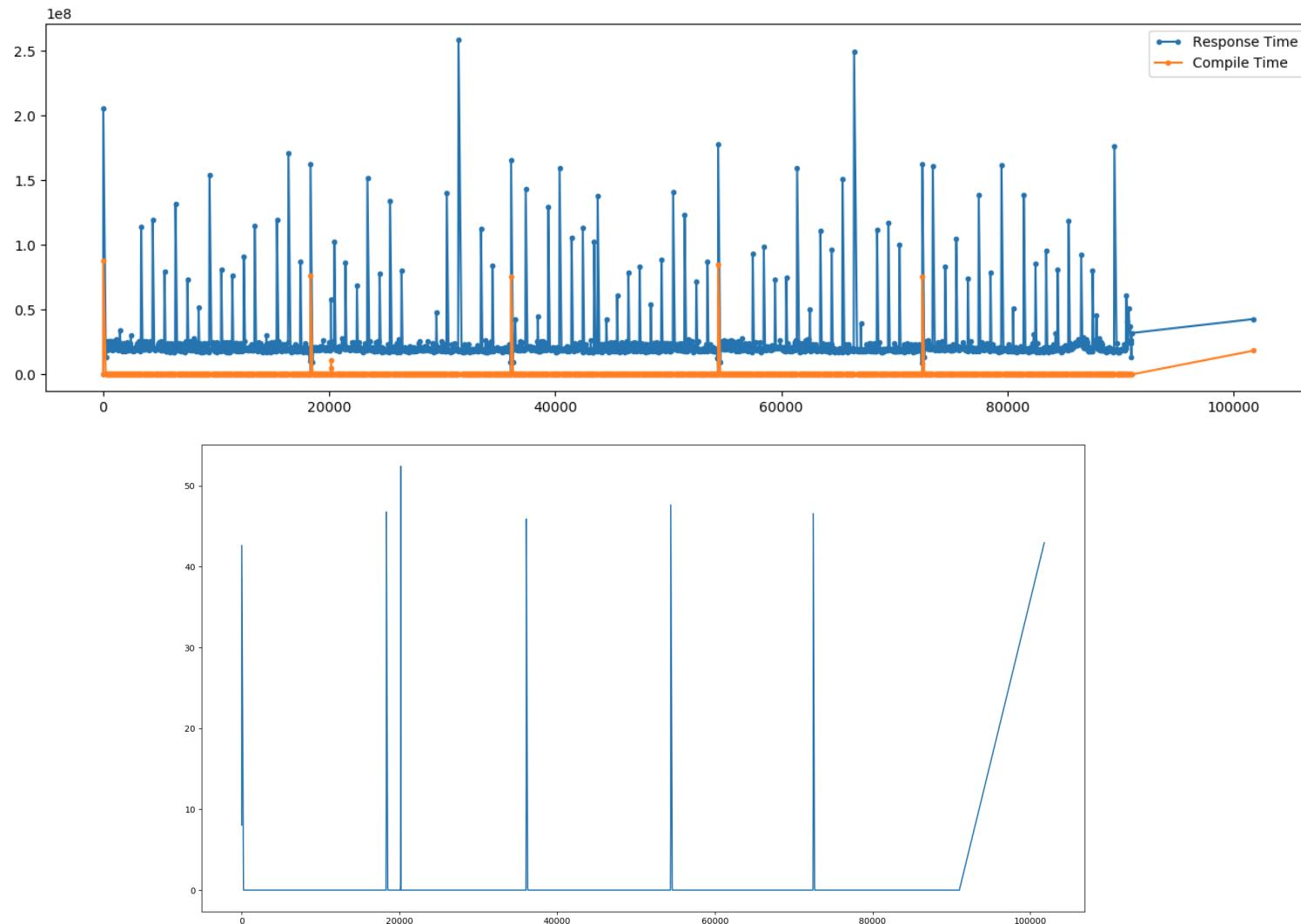
Loads from cache

Response time:  
592 us --> 73 us (with caching)

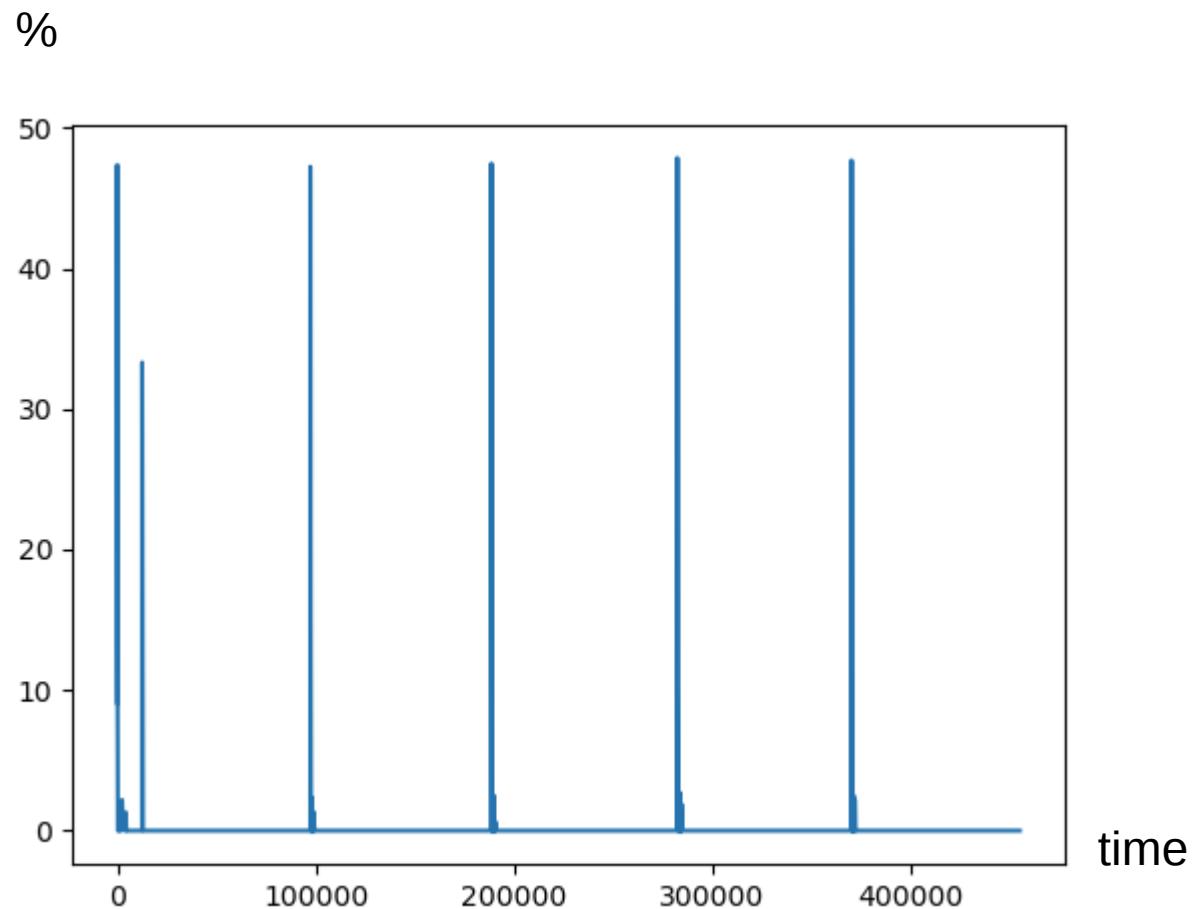
# Compile Time: Drupal



# Compile Time: WordPress

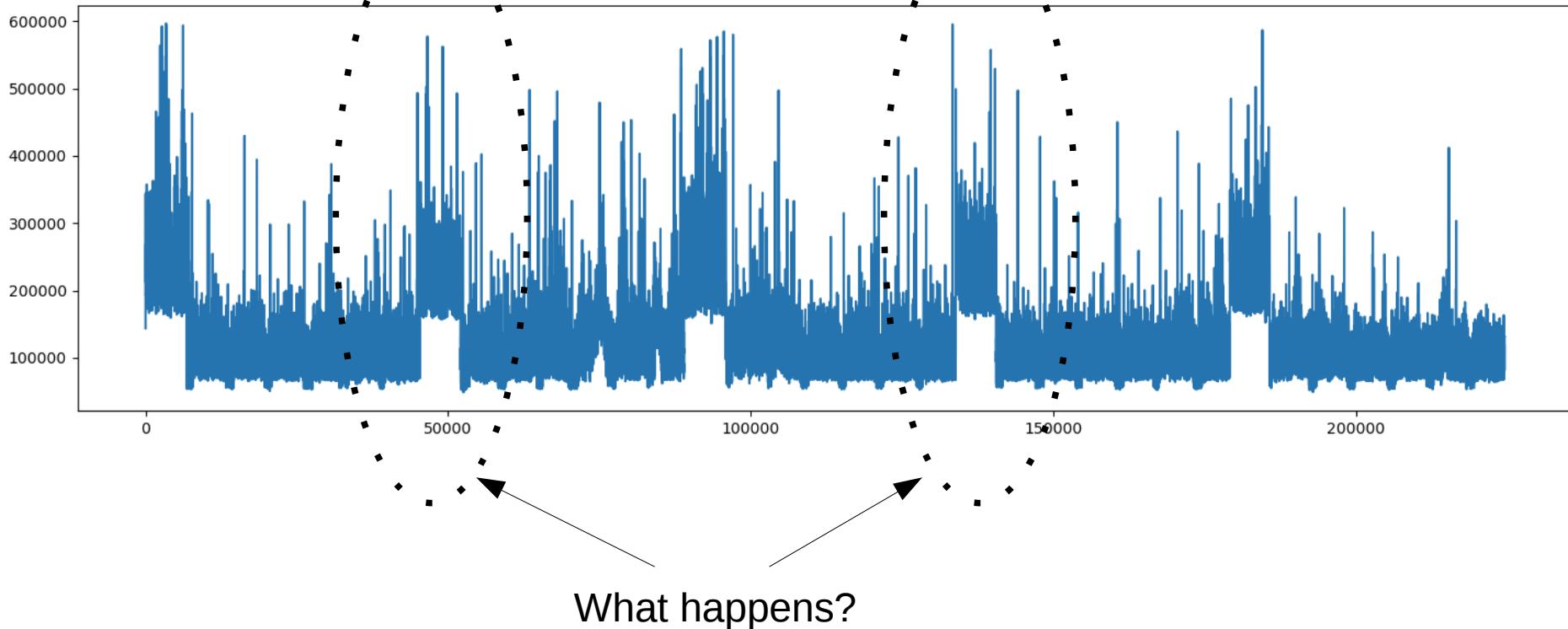


# Compile Time: MediaWiki

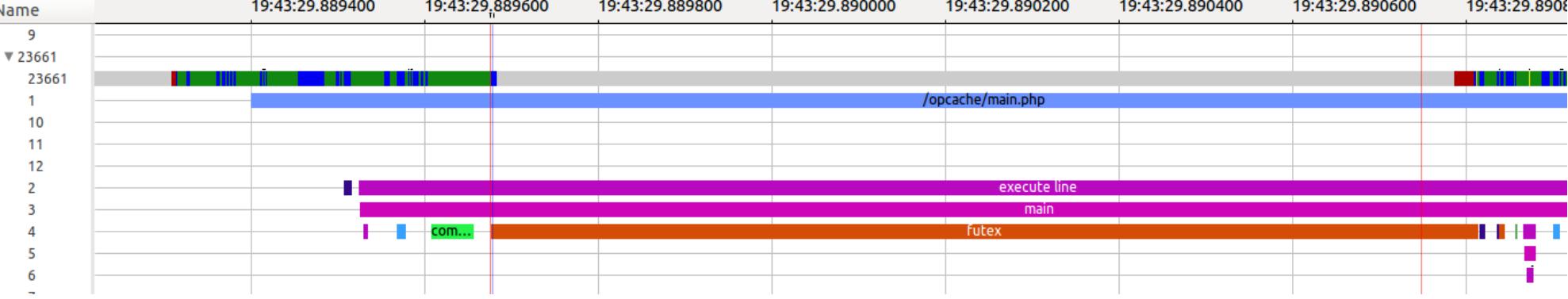


# Let's back to our example

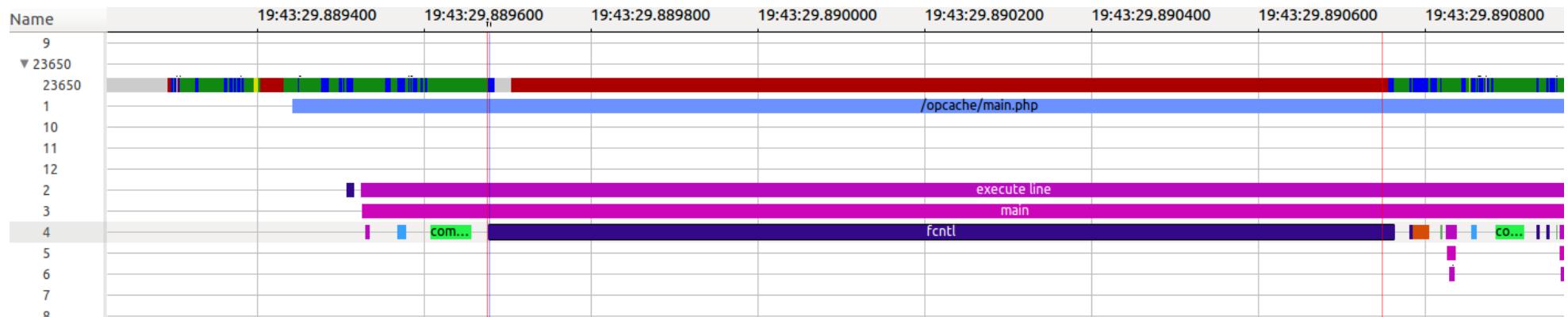
## Response Time



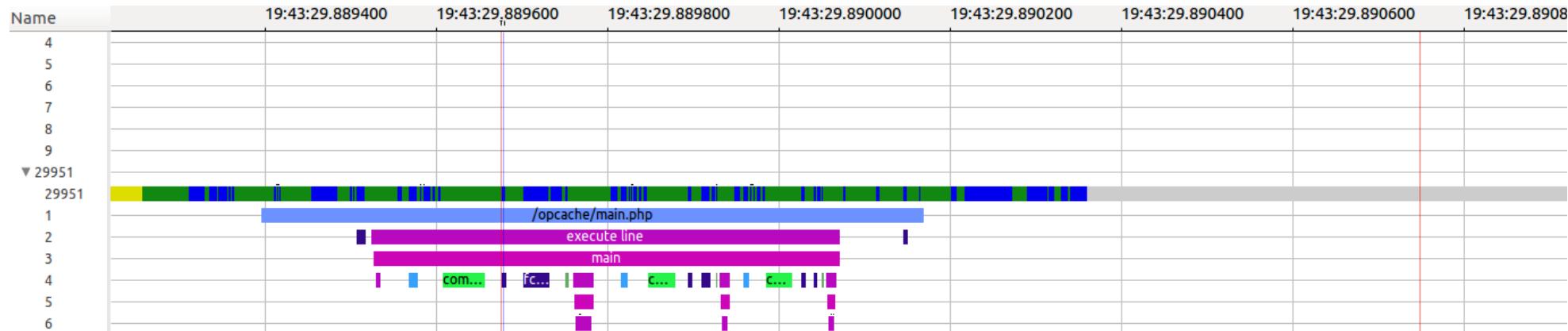
3



2



1



19:43:29.889000

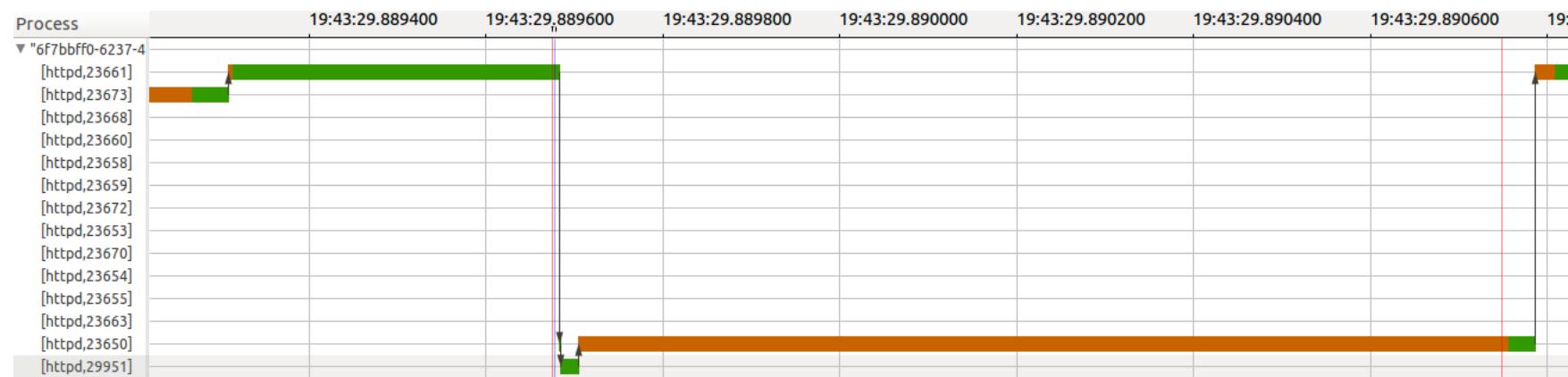
19:43:29.889500

19:43:29.890000

19:43:29.890500

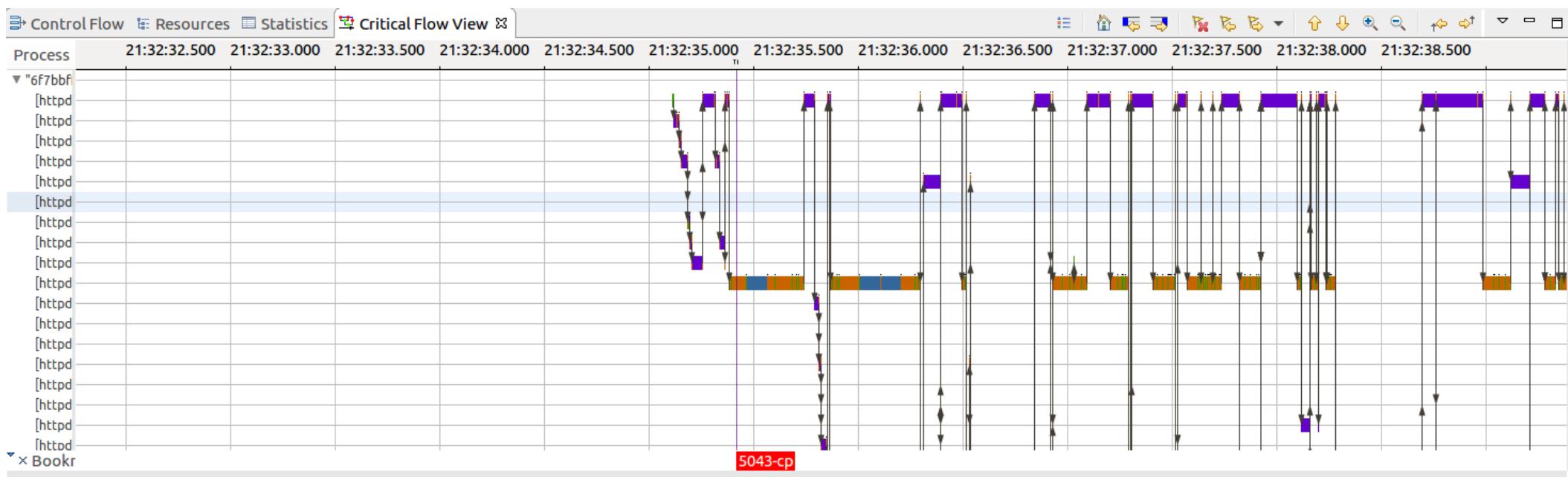
19:43:29.

# Critical Flow



Every PHP process that is willing to write into shared memory will lock every other process willing to write into shared memory as well.

# Critical Flow



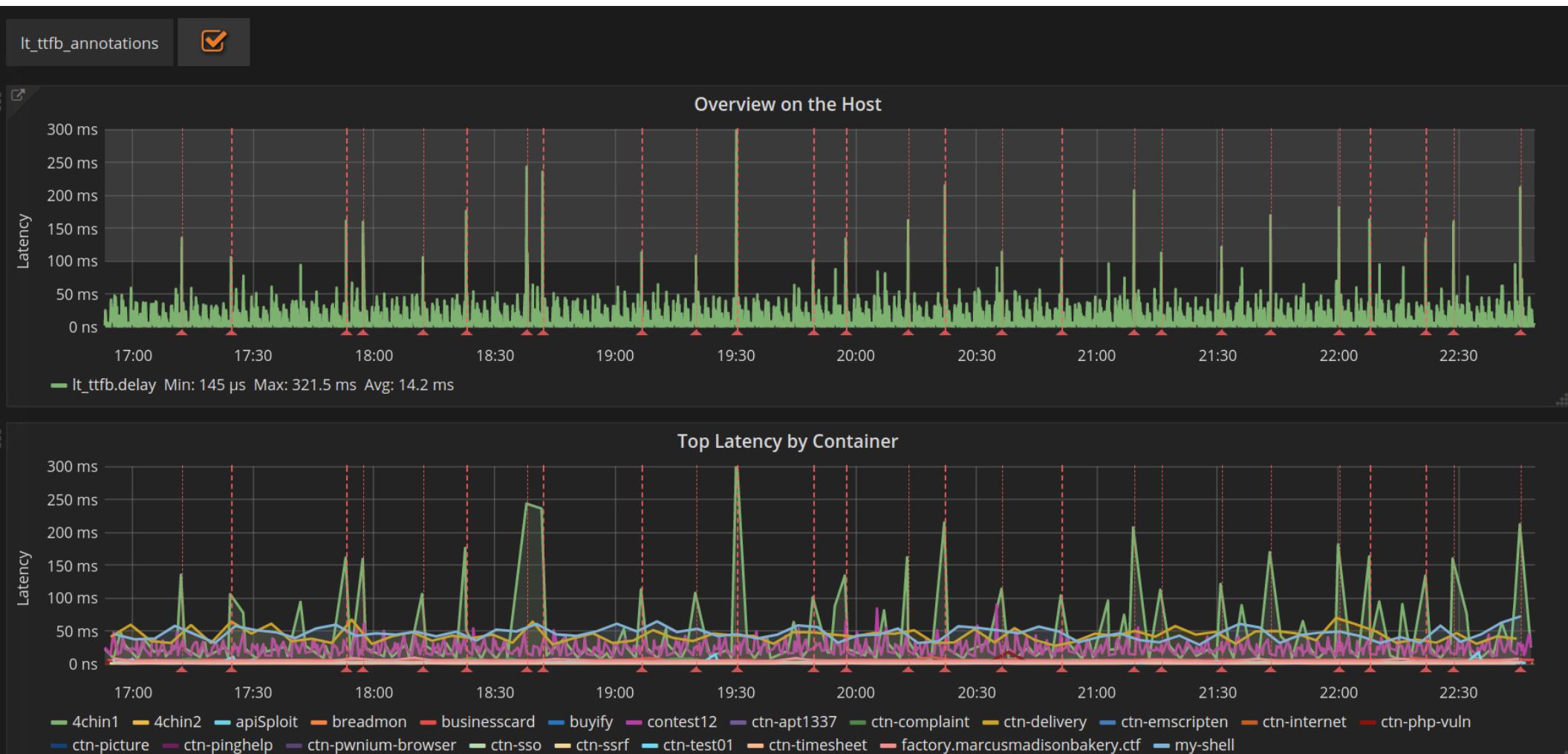
Use-case 2

Latency Analysis

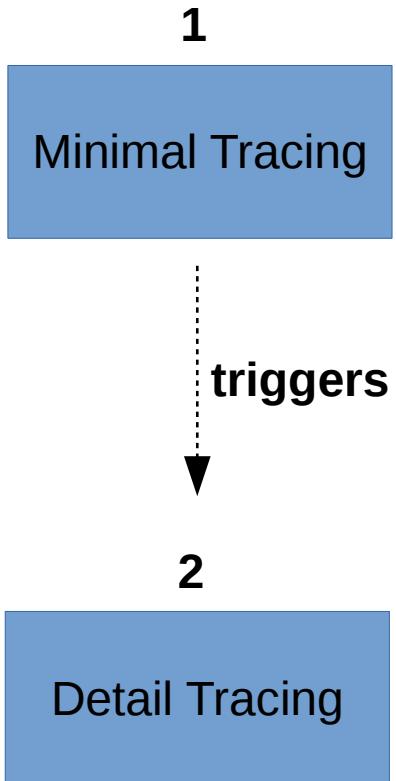
in a

Production Server!

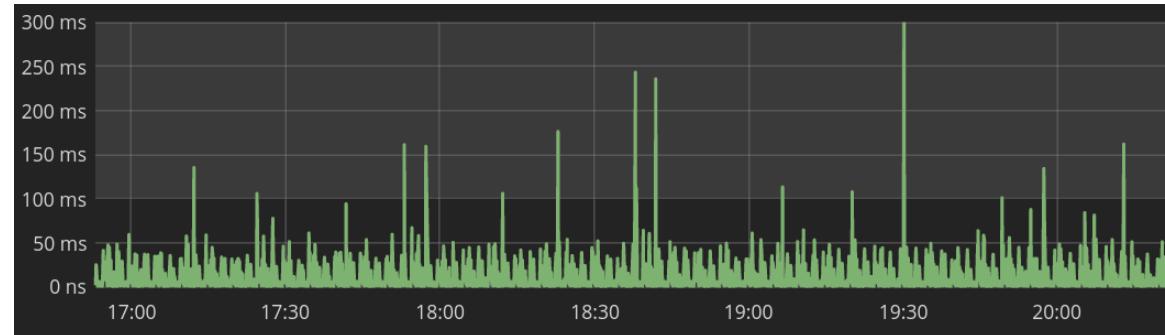
# Tracing is Costly: we can not trace forever!



# Dual Tracing!



## Overview



## Detail

```
[1473798463.279272811] [+0.0000380843] contextID: latency-tracker-1fb: { cpu_id = 0 }, pid = 33975, hostname = "achim1", { comm = "gspace2", tid = 33975, delay = 159135757, fan = 0000:0000:0000:0000:0000:0000:0000:0001", sport = 89, daddr4 = "", daddr6 = "0000:0000:0000:0000:0000:0000:0000:0001", dport = 46636 }
```

Timeline: [2016-10-05 21:57:17.20660996, 2016-10-05 21:57:17.315099881]

Process	Migrations	Priorities
watch-snapshot (984739)	0	[20]
watch-snapshot (984738)	0	[20]
watch-snapshot (984737)	0	[20]
chromium-browser (984664)	0	[20]
chromium-browser (984663)	0	[20]
chromium-browser (984669)	0	[20]
chromium-browser (984718)	0	[20]
chromium-browser (984719)	0	[20]
chromium-browser (984720)	0	[20]
mysql (569718)	0	[20]
mysql (569719)	0	[20]
apache2 (53975)	0	[20]
cat (983956)	0	[20]
cat (983957)	0	[20]
mysql (564790)	0	[20]
Chrome Child1DQT (984721)	0	[20]
Chrome Child1DQT (984720)	0	[20]
Chrome Child1DQT (984723)	0	[20]
watchprocess (531270)	0	[20]
watchprocess (531271)	0	[20]
mysql (32188)	0	[20]
ltpng-relaxed (547220)	0	[20]
ltpng-relaxed (547221)	0	[20]
python3 (563225)	0	[20]
swapper/r (0)	0	[1]

Per-CPU Usage

CPU	Usage (%)
0	10.44 %
1	10.44 %
2	10.44 %
3	10.44 %

Total CPU Usage: 53.47%

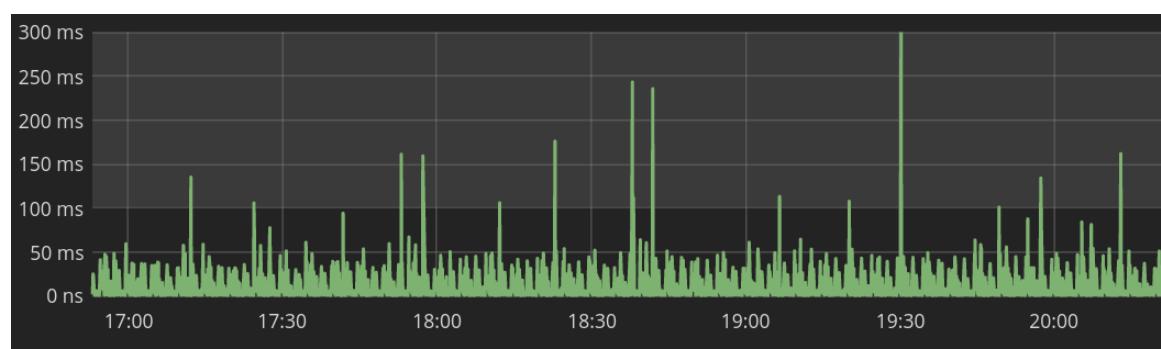
```
Timeline: [2016-10-05 21:57:17.20660996, 2016-10-05 21:57:17.315099881]
```

Name	Duration (usec)	Size	Proc.	PID	Filename
open	55.158	N/A	Chrome IOThread	984698	/dev/hda/.org.chromium.Chromium.MogSRJ (fd=97)
socket	15.489	N/A	redis-server	578959	/proc/582/rstat (fd=5)
open	23.889	N/A	redis-server	578959	33975/socket (fd=1)
socket	23.545	N/A	redis2	33975	33975/socket (fd=1)
open	17.799	N/A	lttng	984739	/lib/x86_64-linux-gnu/liblttng-ctl.so.0 (fd=3)
socket	16.764	N/A	watch-snapshot	33975	33975/socket (fd=1)
open	15.492	N/A	watch-snapshot	984739	/lib/x86_64-linux-gnu/libc.so.6 (fd=3)
open	15.093	N/A	lttng	984739	/root/.lttng/cfg-1
open	13.398	N/A	watch-snapshot	984739	/lib/x86_64-linux-gnu/libarchive.so.1 (fd=1)
open	13.644	N/A	lttng	984739	/lib/x86_64-linux-gnu/liblttng-1.so.1 (fd=3)
open	12.786	N/A	lttng	984739	/usr/x86_64-linux-gnu/libcuru.so.0 (fd=3)

# Dual Tracing: phase 1

1

Minimal Tracing



- **Apache module to generate overview**
  - No kernel!
- **Minimal tracing: only two tracepoints**
  - 2 \* 100ns for each request
- **Trace is relayed to another server**
  - No Disk I/O (or limited disk I/O)
    - The trace is still written to disk but we limit the size
  - No trace processing (very minimal)
    - Low overhead
  - Continues monitoring
  - Production server
    - Distributed or Embedded system

```
$ lttng create --live # optional: -U  
net://<server>  
  
$ lttng enable-event -k -a  
$ lttng enable-event -u -a  
$ lttng start  
$ lttng view  
$ lttng stop  
$ lttng destroy
```

# Dual Tracing: phase 2

2

# Detail Tracing

- **Max tracing: all tracepoints are enabled**
    - Kernel
      - Cputop, iolatencytop, iousagetop, irqstats, memtop, schedtop, schedstats
    - Userspace:
      - Apache module analysis
      - PHP call stack and function profiling
      - Database query details
  - **Useful for detail analysis around a problem**
    - When only required
    - Enabled by triggers
      - High latency
        - For requests beyond the threshold
      - Error in application
      - Segmentation faults
    - No Disk I/O (or limited disk I/O)
      - Memory only tracing
      - On-demand flushing memory to disk

## Detail

```
$ lttnng create --snapshot # optional: -U net://  
$ lttnng enable-event -k -a  
$ lttnng enable-event -u -a  
$ lttnng start  
...  
$ lttnng snapshot record  
...  
$ lttnng snapshot record  
...  
$ lttnng snapshot record
```

# Detail Report (1)

Total queries: 291

Per-(db,table) query statistics (usec)

Table:drupaldb.watchdog	Count	Min	Average	Max	Stdev	Return values
- insert	1	149179.201	149179.201	149179.201		? {'success': 1}
Total:	1					

Table:drupaldb.unknown	Count	Min	Average	Max	Stdev	Return values
- replace	19	36086.314	170591.767	359199.685	105436.274	{'success': 19}
- set	19	15.273	43.039	100.838	30.074	{'success': 19}
- begin	7	8.583	33.02	72.436	25.446	{'success': 7}
- select	3	120.933	172.93	265.593	80.449	{'success': 3}
Total:	48					

Table:drupaldb.sessions	Count	Min	Average	Max	Stdev	Return values
- insert	1	52499.202	52499.202	52499.202		? {'success': 1}
- select	1	364.921	364.921	364.921		? {'success': 1}
Total:	2					

Table:drupaldb.users	Count	Min	Average	Max	Stdev	Return values
- select	6	73.978	81.508	93.247	6.828	{'success': 6}
- update	2	48813.695	74040.974	99268.253		? {'success': 2}
Total:	8					

Table:my_wiki.`objectcache`	Count	Min	Average	Max	Stdev	Return values
- select	69	36.693	100.19	360.522	77.248	{'success': 69}
- delete	6	48989.703	88323.852	166113.703	57627.832	{'success': 6}
Total:	75					

Table:my_wiki.`module_deps`	Count	Min	Average	Max	Stdev	Return values
- select	4	82.057	664.357	2210.545	1031.91	{'success': 4}
Total:	4					

Table:my_wiki.`page`, `revision`, `text`	Count	Min	Average	Max	Stdev	Return values
- select	5	425.976	598.256	829.949	151.928	{'success': 5}
Total:	5					

Table:my_wiki.`page`	Count	Min	Average	Max	Stdev	Return values
- select	7	131.854	469.556	741.87	232.575	{'success': 7}
Total:	7					

# Detail Report (2)

```
[1475704637.279272611] (+0.000030843) contest12 latency_tracker_ttfb: { cpu_id = 0 }, { pid = 33975, hostname = "4chin1" }, { comm = "apache2", tid = 33975, delay = 159135757, far
"0000:0000:0000:0000:0000:0001", sport = 80, daddr4 = "", daddr6 = "0000:0000:0000:0000:0000:0000:0000:0001", dport = 46036 }
Timerange: [2016-10-05 21:57:17.206009096, 2016-10-05 21:57:17.315099881]

Per-TID Usage
#####
Process          Migrations   Priorities
#####
watch-snapshot. (904739)      0  [20]
watch-snapshot. (904738)      0  [20]
chromium-browse (904715)      0  [20]
chromium-browse (904664)      0  [20]
chromium-browse (904713)      0  [20]
Chrome_IOThread (904690)      0  [20]
chromium-browse (904718)      0  [20]
exe (904727)                0  [20]
mysqld (569710)              0  [20]
watch-snapshot. (843840)      0  [20]
apache2 (33975)              0  [20]
Compositor (904722)          0  []
cat (903956)                 0  [20]
migration/3 (23)              0  [-100]
mysqld (568709)              0  [20]
Chrome_ChildIOT (904717)     0  [20]
Chrome_ChildIOT (904720)     0  [20]
Chrome_ChildIOT (904728)     0  [20]
Chrome_CacheThr (904689)     0  [20]
watchquagga (531270)         0  [20]
kworker/u8:7 (893971)        0  [20]
mysqld (32188)               0  [20]
nginx (571700)               0  [20]
handle-watcher- (904729)     0  [20]
bgpd (533899)                0  [20]
CompositorTileW (904692)     0  [20]
lttng-relayd (547220)        0  [20]
bgpd (541956)               0  [20]
python3 (563325)             0  [20]
swapper/1 (0)                 0  []

Per-CPU Usage
#####
100.00 % CPU 0
10.64 % CPU 1
23.08 % CPU 2
80.16 % CPU 3

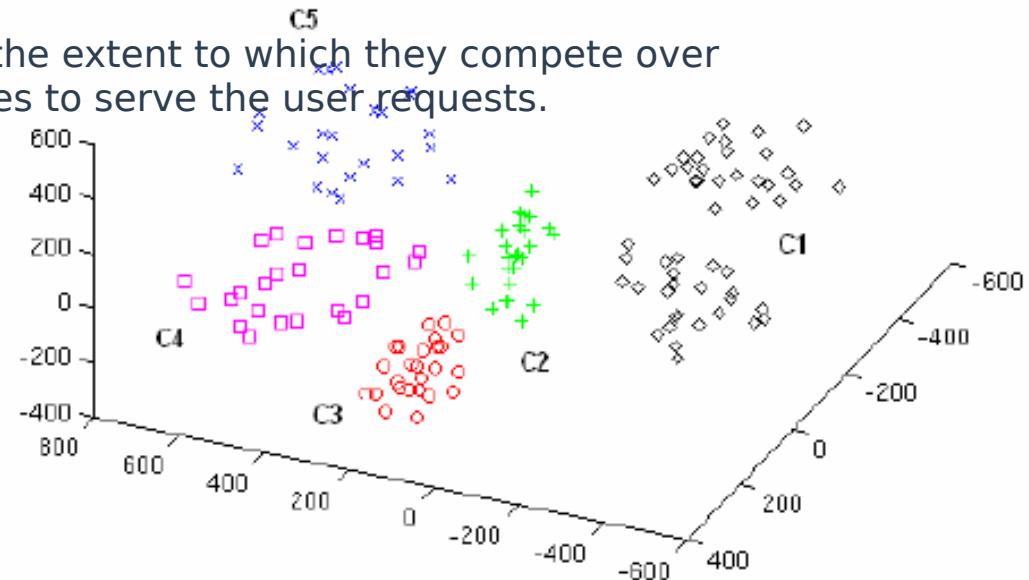
Total CPU Usage: 53.47%
Timerange: [2016-10-05 21:57:17.206009096, 2016-10-05 21:57:17.315099881]

Top system call latencies open (usec)
Begin           End       Name    Duration (usec)  Size  Proc          PID   Filename
[21:57:17.290045348, 21:57:17.290100498] open      55.150  N/A  Chrome_IOThread 904690 /dev/shm/.org.chromium.Chromium.Mog5RJ (fd=97)
[21:57:17.314092856, 21:57:17.314129124] socket    36.268  N/A  lttng          904739 socket (fd=3)
[21:57:17.293075275, 21:57:17.293099164] open      23.889  N/A  redis-server  570939 /proc/502/stat (fd=5)
[21:57:17.224349312, 21:57:17.224371857] socket    22.545  N/A  apache2        33975 socket (fd=13)
[21:57:17.304907877, 21:57:17.304925856] open      17.979  N/A  lttng          904739 /usr/lib/x86_64-linux-gnu/liblttng-ctl.so.0 (fd=3)
[21:57:17.251597805, 21:57:17.251614569] socket    16.764  N/A  apache2        33975 socket (fd=13)
[21:57:17.292726004, 21:57:17.292741406] open      15.402  N/A  watch-snapshot. 904738 /lib/x86_64-linux-gnu/libc.so.6 (fd=3)
[21:57:17.313395044, 21:57:17.313410137] open      15.093  N/A  lttng          904739 /root/.lttngrc (fd=3)
[21:57:17.295445952, 21:57:17.295460142] open      14.190  N/A  watch-snapshot. 904738 /usr/lib/locale/locale-archive (fd=3)
[21:57:17.305151063, 21:57:17.305164707] open      13.644  N/A  lttng          904739 /lib/x86_64-linux-gnu/librt.so.1 (fd=3)
[21:57:17.306684049, 21:57:17.306696835] open      12.786  N/A  lttng          904739 /usr/lib/x86_64-linux-gnu/liburcu.so.4 (fd=3)
```

# Dual Tracing: phase 3

## 3-Clustering

- Based on behavioral characteristics of the problematic requests
  - Similarity measured by their system resources consumption behavior
- Bugs are easier to understand, predict and maintain:
  - If they are clustered into closely related groups based on the underlying root causes.
    - Grouping the faulty requests based on the extent to which they compete over consuming the existing system resources to serve the user requests.
  - Learn from them and label them
  - Use in the production server



# Resources

**LTTng:**

<http://lttng.org/>

**Trace Compass:**

<http://tracecompass.org>

**Modules to Install:**

<https://github.com/naser>

**Thank you**

n.ezzati@polymtl.ca