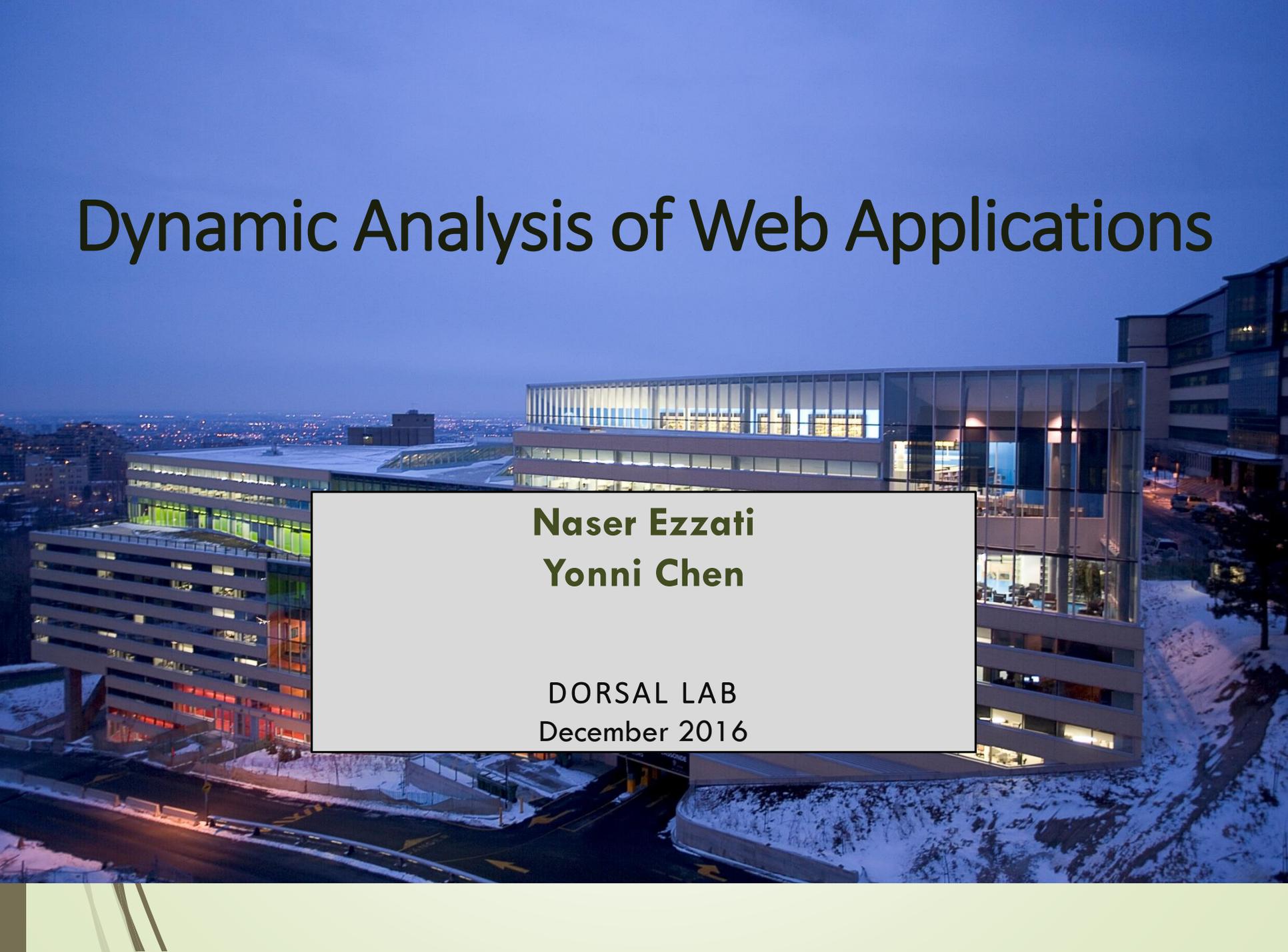


# Dynamic Analysis of Web Applications



**Naser Ezzati**  
**Yonni Chen**

DORSAL LAB  
December 2016

# Motivation

- Challenge: bottleneck identification and root cause analysis in web applications
  - Several components and layers are involved
  - Various debugging tools
  - We propose a unified way to analyze and monitor web applications
    - Trace-based approach (using LTTng)
    - LAMP stack
    - Nginx, PHP-FPM
    - Later: MEAN stack

## LAMP:



MongoDB

ExpressJS



AngularJS



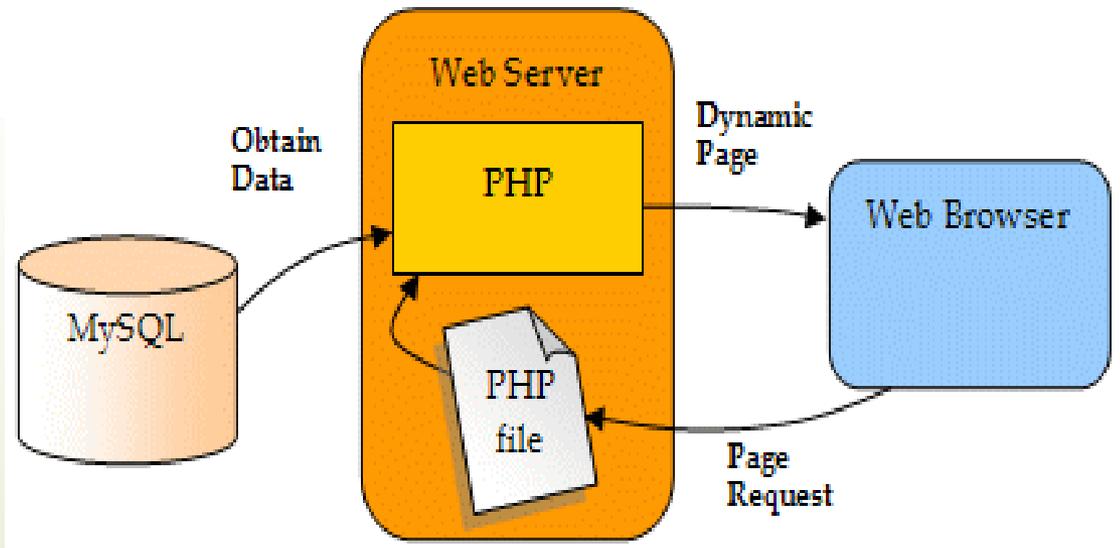
NodeJS

# LTTng-UST

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

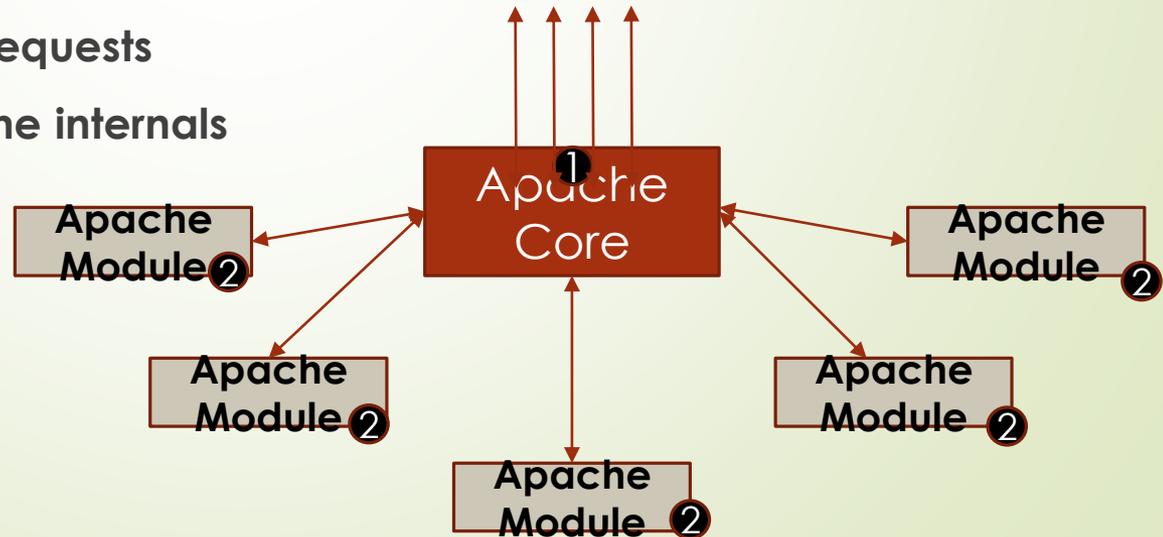
    [...]
    tracepoint(ust_tests_hello,
               tptest,
               i,
               &vals,
               flt);

    [...]
}
```



# Apache

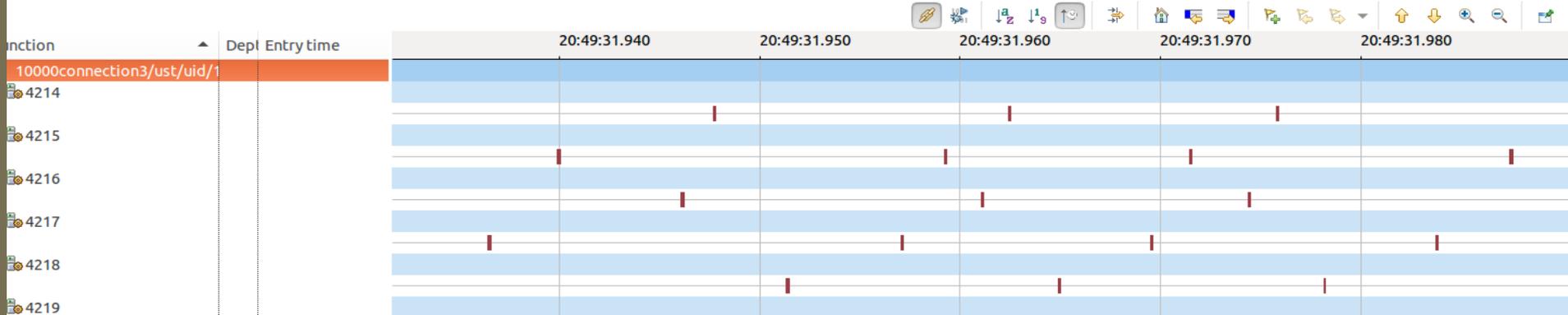
- **Apache LTTng module: a framework to hook LTTng probes to the Apache web server.**
- These probes provide runtime information about Apache and can be used to monitor web server behavior
- The module provides two sets of data
  1. **Web requests**
  2. **Apache internals**



# Apache Request 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=(null)
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=(null)
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= request
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= request
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=(null)
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= request
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

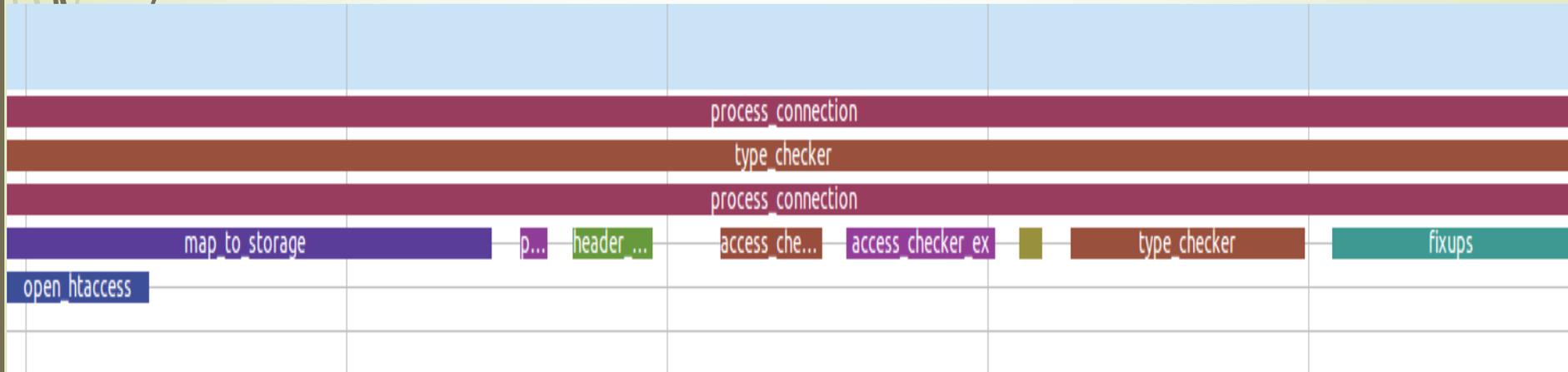
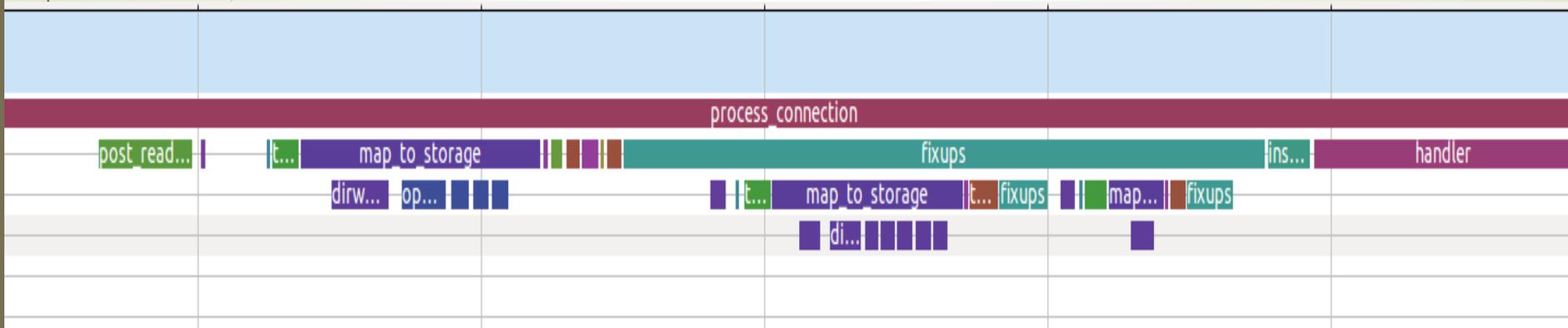
Properties Bookmark State Syst Critical Flo Progress LAMP call Mysql quer Flame Gra LAMP dep XML Time Apache m LAMP dep Analysis R lamptop



# Apache Module Tracing

Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	<b>ust_apache:apache_module</b>	<srch>
11:12:07.494 274 785	channel0_3	3	ust_php:request_exit	path=/usr/local/apache2/htdocs/drupal/index.php,uri=/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

# Apache Modules Tracing (2)





# PHP and LTTng

- ▶ LTTng probes in PHP
- ▶ Provide detailed information about the execution of php codes
- ▶ Monitor the entire php script execution process:
  - ▶ 13 tracepoints
    - ▶ start/close a request
    - ▶ call a function
    - ▶ execute a line
    - ▶ make a db connection
    - ▶ errors/exceptions
  - ▶ detailed information in the arguments
    - ▶ request info, function name, file name, class name, line number, etc.
- ▶ Trace Compass views:
  - ▶ Callstack, Flame Graph, Request top/list

# PHP Core Modification

```
61
62 /* We wrap the execute function to have fire the execute-entry/return and function-entry/return probes */
63 ZEND_API void lttng_execute_ex(zend_execute_data *execute_data)
64 {
65     int lineno;
66     const char *scope, *filename, *funcname, *classname;
67     scope = filename = funcname = classname = NULL;
68
69     /* we need filename and lineno for both execute and function probes */
70     classname = get_active_class_name(&scope);
71     filename = lttng_get_executed_filename();
72     funcname = get_active_function_name();
73     lineno = zend_get_executed_lineno();
74
75     tracepoint(ust_php, execute_entry, filename, lineno);
76
77     if (funcname != NULL)
78         tracepoint(ust_php, function_entry, funcname, filename, lineno, classname, scope);
79
80     execute_ex(execute_data);
81
82     if (funcname != NULL)
83         tracepoint(ust_php, function_exit, funcname, filename, lineno, classname, scope);
84
85     tracepoint(ust_php, execute_exit, filename, lineno);
86 }
87
88 ZEND_API void lttng_execute_internal(zend_execute_data *execute_data, zval *return_value)
89 {
90     int lineno;
91     const char *filename;
92     filename = lttng_get_executed_filename();
93     lineno = zend_get_executed_lineno();
94
95     tracepoint(ust_php, execute_entry, filename, lineno);
96
97     execute_internal(execute_data, return_value);
98
99     tracepoint(ust_php, execute_exit, filename, lineno);
100 }
```

Probe Name	Probe Description	Probe Arguments
<i>request-startup</i>	Fires when a request starts.	<i>char *file, char *request_uri, char *request_method</i>
<i>request-shutdown</i>	Fires when a request shutdown.	<i>char *file, char *request_uri, char *request_method</i>
<i>compile-file-entry</i>	Fires when the compilation of a script starts.	<i>char *compile_file, char *compile_file_translated</i>
<i>compile-file-return</i>	Fires when the compilation of a script finishes.	<i>char *compile_file, char *compile_file_translated</i>
<i>execute-entry</i>	Fires when an opcode array is to be executed. For example, it fires on function calls, includes, and generator resumes.	<i>char *request_file, int lineno</i>
<i>execute-return</i>	Fires after execution of an opcode array.	<i>char *request_file, int lineno</i>
<i>function-entry</i>	Fires when the PHP engine enters a PHP function or method call.	<i>char *function_name, char *request_file, int lineno, char *classname, char *scope</i>
<i>function-return</i>	Fires when the PHP engine returns from a PHP function or method call.	<i>char *function_name, char *request_file, int lineno, char *classname, char *scope</i>
<i>exception-thrown</i>	Fires when an exception is thrown.	<i>char *classname</i>
<i>exception-caught</i>	Fires when an exception is caught.	<i>char *classname</i>

# To Use:

- ./configure --enable-ltng
- make
- sudo make install

```
/home/naserez $> ltng list -u | grep php
ust_php:php_mysql_close (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
ust_php:php_mysql_connect (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
ust_php:php_exception_thrown (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
ust_php:php_exception_caught (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
ust_php:php_error (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
ust_php:request_exit (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
ust_php:request_entry (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
ust_php:execute_exit (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
ust_php:execute_entry (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
ust_php:compile_file_exit (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
ust_php:compile_file_entry (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
ust_php:function_exit (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
ust_php:function_entry (loglevel: TRACE_DEBUG_LINE (13)) (type: tracepoint)
```

# Views

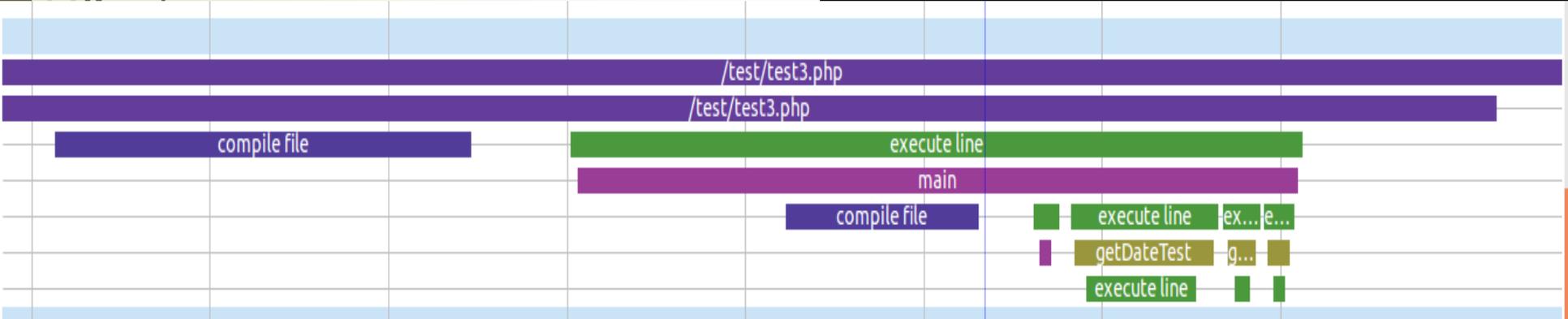
➔ Callstack

```
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 ?>
```

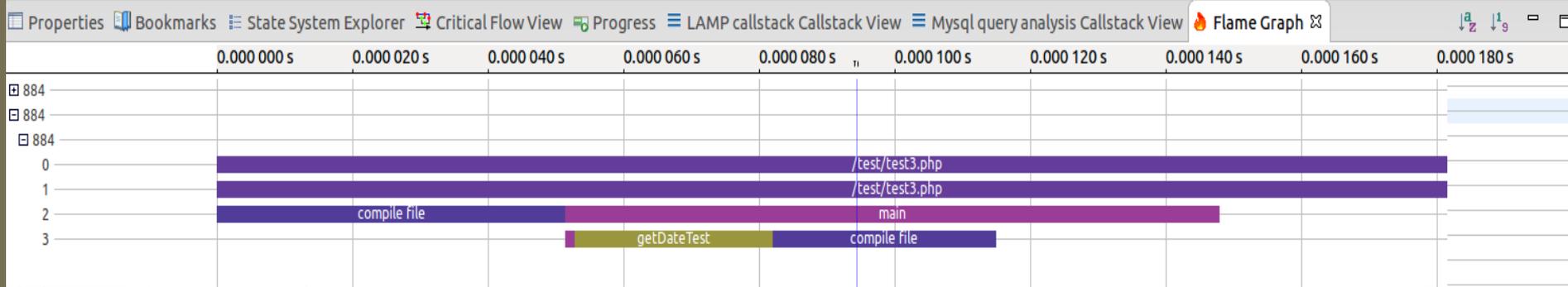
/usr/local/apache2/htdocs/test/test3.php [FORMA

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

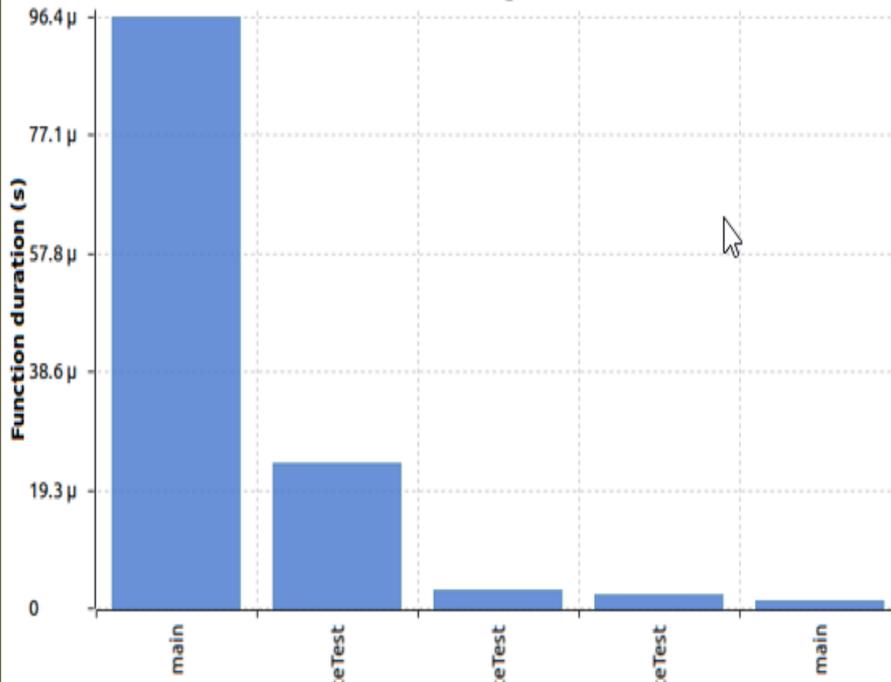
/usr/local/apache2/htdocs/test/include.php [FOR



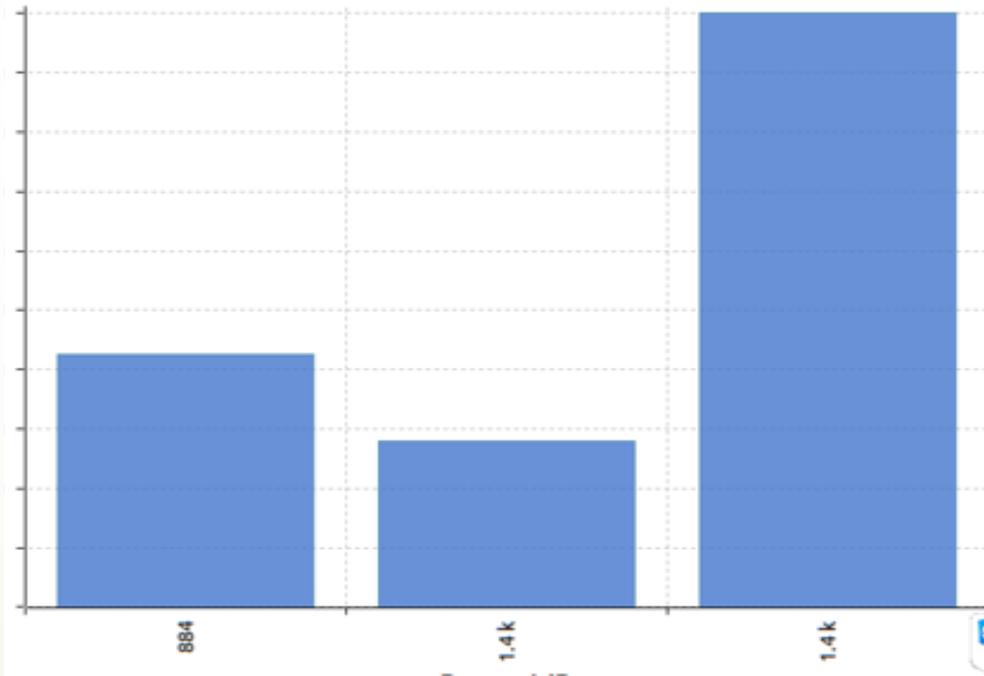
# Views(2): Flame Graph



PHP function log [884]

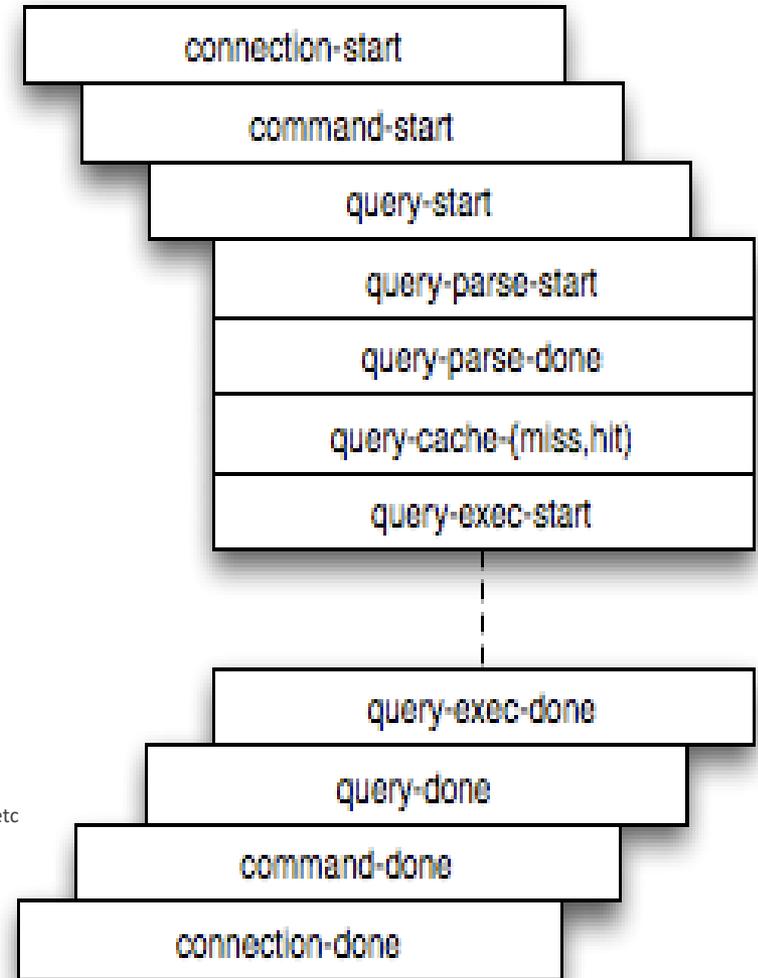


PHP requests statistics



# MySQL and LTTng

- ▶ LTTng probes in MySQL server
- ▶ Provide information about the execution of queries
- ▶ 60 tracepoints, 200 locations in mysql
- ▶ Monitor the full query execution process
  - ▶ start a connection
  - ▶ execute
    - ▶ row-level operations
    - ▶ cache miss, hit
    - ▶ Locks used
    - ▶ Sort methods
    - ▶ Network I/O
  - ▶ close a connection
  - ▶ detailed information in the arguments
    - ▶ connection ID, db name, user, host, query string, etc



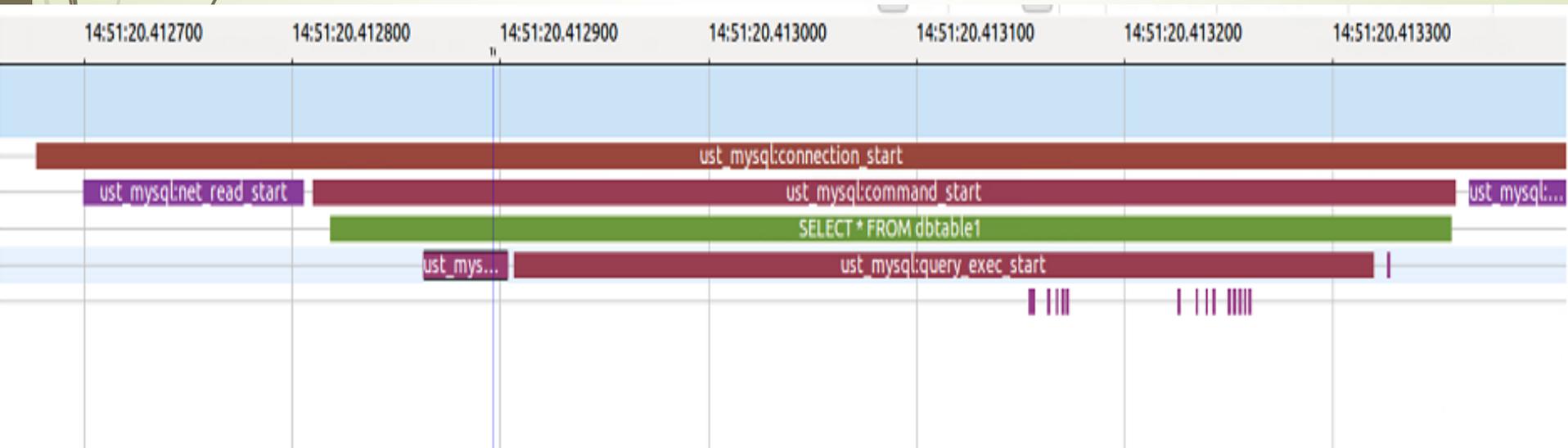
# To Use:

- `./configure --enable-lttnng`
- `make`
- `make install`

```
/home/naserez $> lttnng 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_rlock_done (loglevel: TRACE DEBUG LINE (13)) (type: tracepoint)
ust_mysql:handler_rlock_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)
```

# Views

- Example1: Select \* from dbtable1;



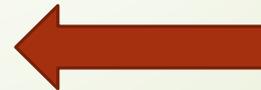
# Example2

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

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

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

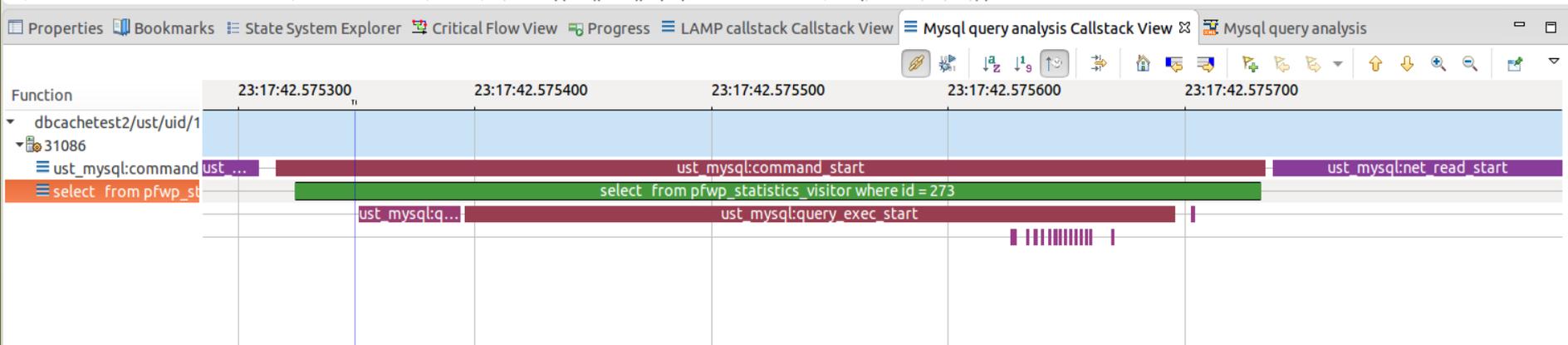
```
select * from pfw_statistics_visitor where id = 273;  
(cache_hit) T: 42,708 ns
```



# Example2: cache miss

➔ query\_cache\_miss

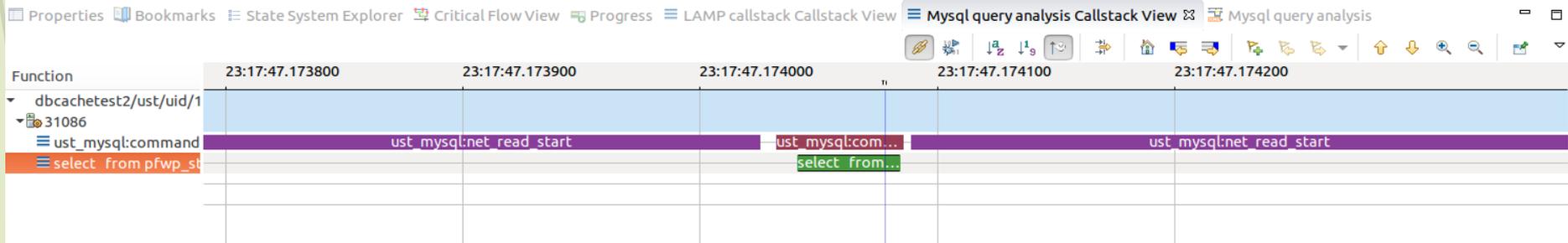
Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	<srch>	<srch>
23:17:39.369 848 397	ss_2	2	ust_mysql:command_done	result=0, thread_id=17787, context._vtid=31086
23:17:39.369 856 327	ss_2	2	ust_mysql:net_read_start	context._vtid=31086
23:17:42.575 308 310	ss_0	0	ust_mysql:net_read_done	result=0,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



# Example2: cache hit

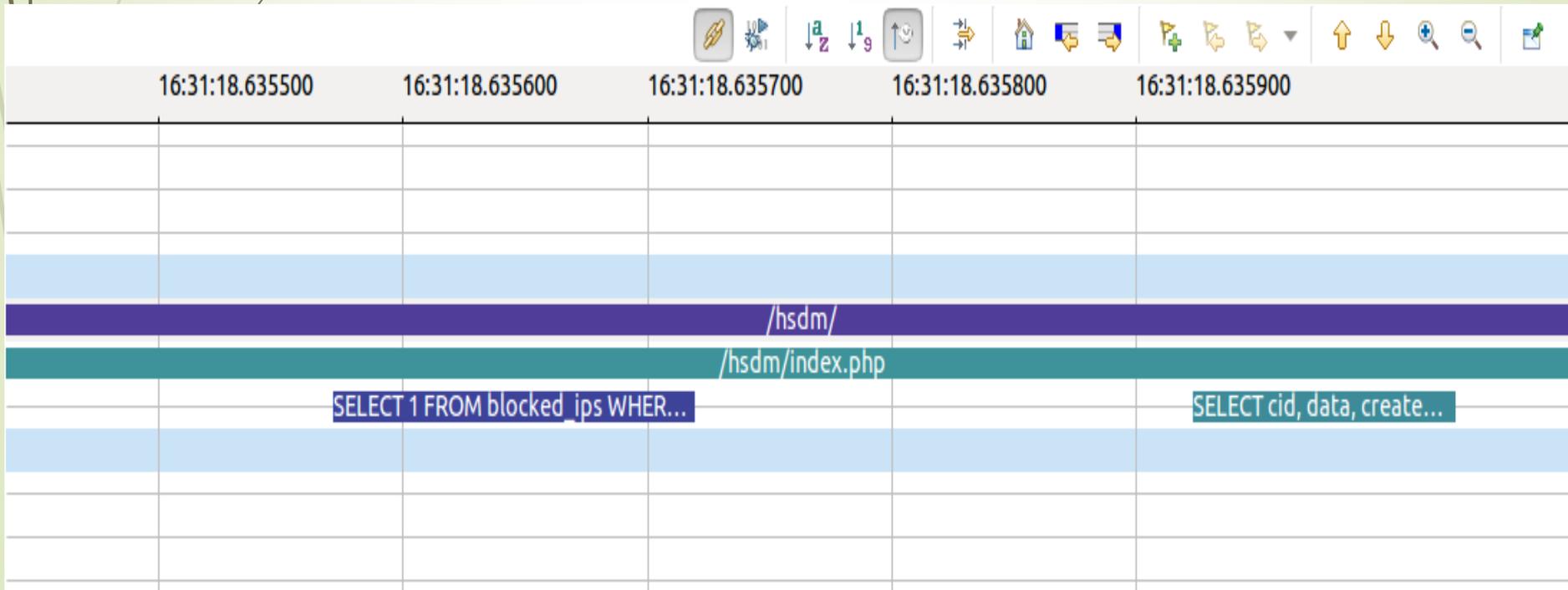
➔ query\_cache\_hit

Timestamp	Channel	CPU	Event type	Contents
<srch>	<srch>	<srch>	<srch>	<srch>
23:17:42.575 733 761	ss_0	0	ust_mysql:command_done	result=0, thread_id=17787, context._vtid=31086
23:17:42.575 737 349	ss_0	0	ust_mysql:net_read_start	context._vtid=31086
23:17:47.174 025 126	ss_3	3	ust_mysql:net_read_done	result=0, len=53, context._vtid=31086
23:17:47.174 032 424	ss_3	3	ust_mysql:command_start	thread_id=17787, command=3, priv_user=root, host_or_ip=localhost, context._vtid=31086
23:17:47.174 041 273	ss_3	3	ust_mysql:query_start	query=select * from pfw_statistics_visitor where id = 273, thread_id=17787, db=my_wiki, priv_user=root
23:17:47.174 077 954	ss_3	3	ust_mysql:query_cache_hit	query=select * from pfw_statistics_visitor where id = 273, limit_found_rows=1, context._vtid=31086
23:17:47.174 083 981	ss_3	3	ust_mysql:query_done	result=0, thread_id=17787, context._vtid=31086
23:17:47.174 085 586	ss_3	3	ust_mysql:command_done	result=0, thread_id=17787, context._vtid=31086
23:17:47.174 089 048	ss_3	3	ust_mysql:net_read_start	context._vtid=31086
23:17:51.879 308 878	ss_3	3	ust_mysql:net_read_done	result=0, len=53, context._vtid=31086



# Integrated Analysis for Web Applications

➤ Request Dependency Graph

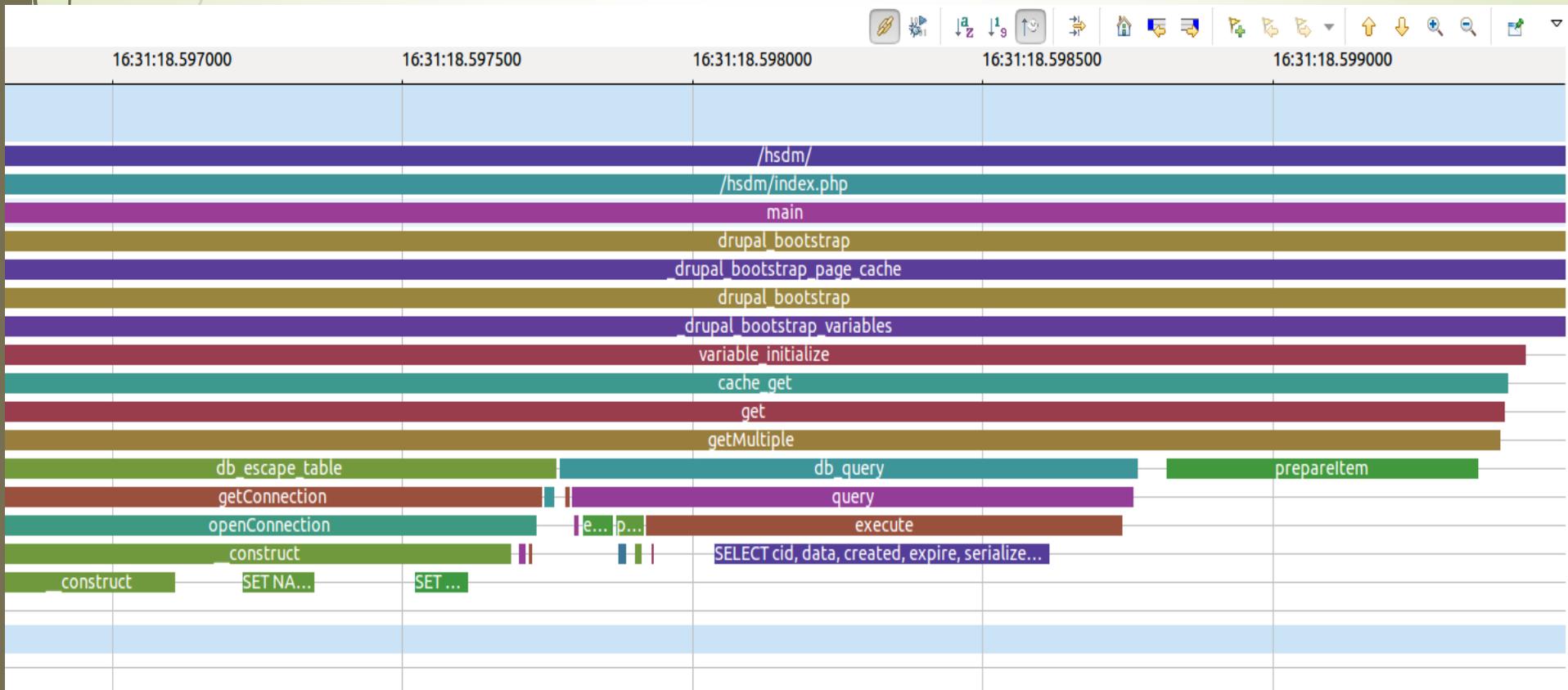


The screenshot displays a request dependency graph with a toolbar at the top and a table of request events below. The toolbar includes icons for editing, zooming, and navigation. The table has five columns representing time intervals from 16:31:18.635500 to 16:31:18.635900. The events shown are:

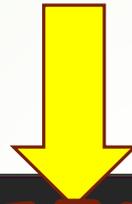
Time	Request	SQL Query
16:31:18.635500		
16:31:18.635600		
16:31:18.635700	/hsdm/	
16:31:18.635800	/hsdm/index.php	
16:31:18.635900		SELECT cid, data, create...

# Integrated Analysis for Web Applications: Call Graph

- Call graph: apache requests, php function calls and mysql queries.



# TOP Requests



Time Range: [2016-09-14 20:40:37.751341541, 2016-09-14 20:41:29.596591398]

Apache Top:

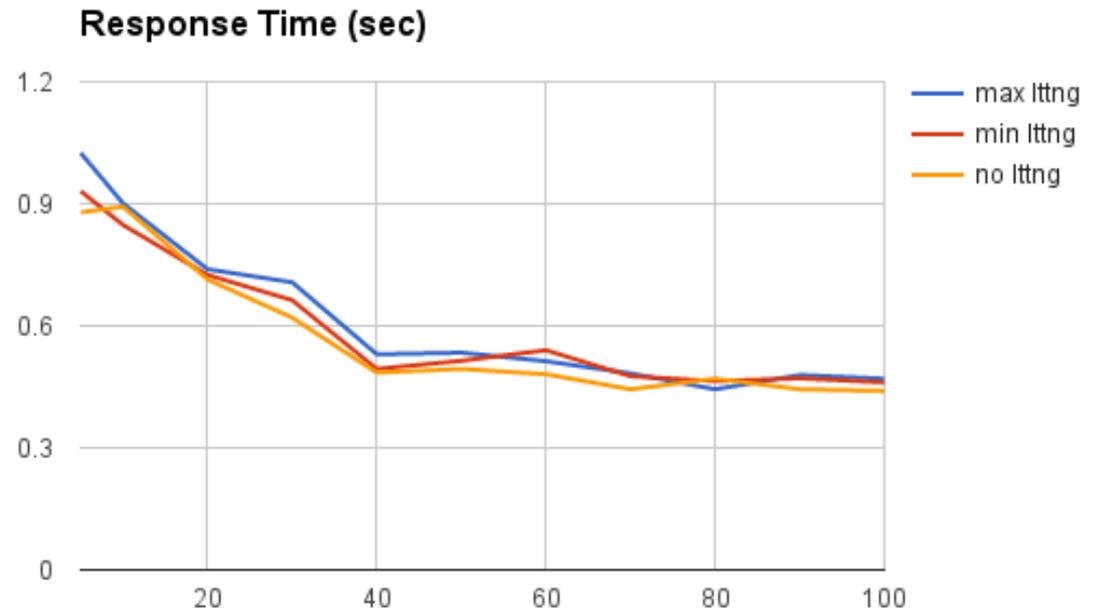
Time Range	Response Time	(% Spent in php and mysql)	TID	Method	URI
[20:41:20.984309151, 20:41:24.693434577]	3709125.426	(apache: 0.020 %, php: 86.325 %, mysql: 13.655 %)	9111	GET	132.207.72.37/polyfab/wiki/load.php
[20:40:54.261234556, 20:40:57.787529428]	3526294.872	(apache: 0.022 %, php: 90.633 %, mysql: 9.344 %)	8955	GET	132.207.72.37/polyfab/wiki/load.php
[20:41:14.983114689, 20:41:18.172910580]	3189795.891	(apache: 1.529 %, php: 58.017 %, mysql: 40.454 %)	8929	GET	132.207.72.37/polyfab/wiki/index.php/Ressources
[20:41:09.021468218, 20:41:11.921474018]	2900005.800	(apache: 0.030 %, php: 91.352 %, mysql: 8.617 %)	8927	GET	132.207.72.37/polyfab/wiki/load.php
[20:40:50.227325472, 20:40:53.077147564]	2849822.09	(apache: 0.025 %, php: 66.012 %, mysql: 33.964 %)	8950	GET	132.207.72.37/polyfab/wiki/load.php
[20:41:20.989529057, 20:41:23.818049619]	2828520.50	(apache: 0.148 %, php: 79.904 %, mysql: 19.948 %)	8931	GET	132.207.72.37/polyfab/wiki/load.php
[20:40:54.262046633, 20:40:57.076147767]	2814101.13	(apache: 0.035 %, php: 88.118 %, mysql: 11.847 %)	8956	GET	132.207.72.37/polyfab/wiki/load.php
[20:41:03.544898170, 20:41:06.304035476]	2759137.306	(apache: 0.076 %, php: 65.918 %, mysql: 34.006 %)	8916	GET	132.207.72.37/polyfab/wiki/index.php/Informations
[20:40:48.142448383, 20:40:50.868722822]	2726274.439	(apache: 0.130 %, php: 64.437 %, mysql: 35.433 %)	8946	GET	132.207.72.37/polyfab/wiki/index.php/Accueil
[20:41:17.789501232, 20:41:20.249221766]	2459720.534	(apache: 0.050 %, php: 82.078 %, mysql: 17.873 %)	9041	GET	132.207.72.37/polyfab/wiki/load.php
Total:	10				

Php Top:

Time Range	Response Time	(% Spent in mysql)	TID	Method	URI	SQL TID(s)	No of SQL Query
[20:41:20.984898904, 20:41:24.693296362]	3708397.458	(mysql: 13.658 %)	9111	GET	/polyfab/wiki/load.php	4407,4414	50
[20:40:54.261875801, 20:40:57.787383105]	3525507.304	(mysql: 9.347 %)	8955	GET	/polyfab/wiki/load.php	4371,4376	58
[20:41:15.031814970, 20:41:18.172828398]	3141013.428	(mysql: 41.082 %)	8929	GET	/polyfab/wiki/index.php/Ressources	4396,4397,4398	203
[20:41:09.022227546, 20:41:11.921356875]	2899129.329	(mysql: 8.620 %)	8927	GET	/polyfab/wiki/load.php	4390,4395	49
[20:40:50.227976347, 20:40:53.077093164]	2849116.817	(mysql: 33.972 %)	8950	GET	/polyfab/wiki/load.php	4363,4365,4366	80
[20:41:20.993614041, 20:41:23.817945338]	2824331.297	(mysql: 19.977 %)	8931	GET	/polyfab/wiki/load.php	4410,4413	58
[20:40:54.262955699, 20:40:57.076061031]	2813105.332	(mysql: 11.851 %)	8956	GET	/polyfab/wiki/load.php	4372,4375	49
[20:41:03.546893396, 20:41:06.303932328]	2757038.932	(mysql: 34.032 %)	8916	GET	/polyfab/wiki/index.php/Informations	4377,4378,4379	159
[20:40:48.145928370, 20:40:50.868668015]	2722739.645	(mysql: 35.479 %)	8946	GET	/polyfab/wiki/index.php/Accueil	4358,4359,4360	167
[20:41:17.790576746, 20:41:20.249076101]	2458499.355	(mysql: 17.882 %)	9041	GET	/polyfab/wiki/load.php	4401,4405	49
Total:	10						

# Performance

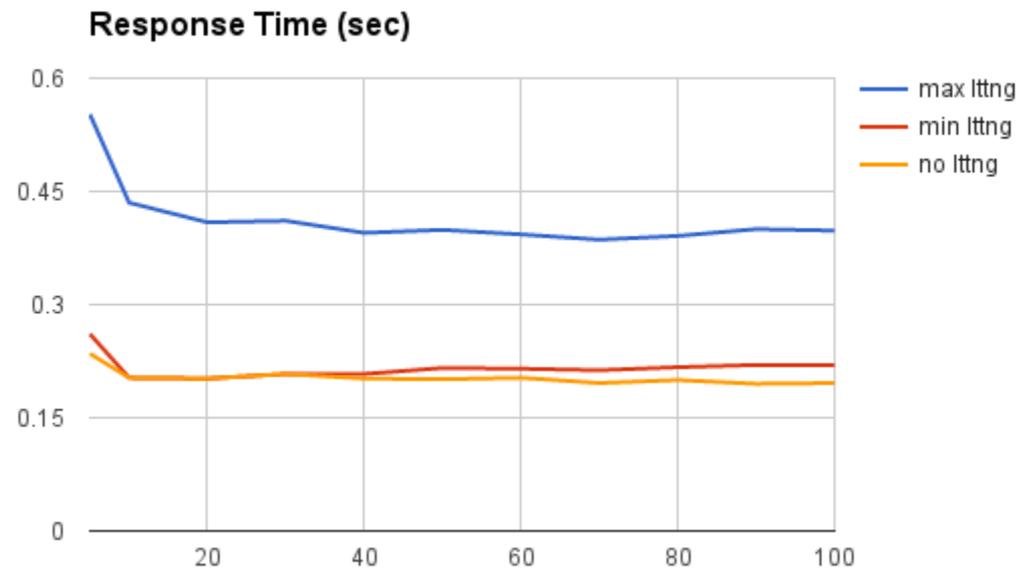
- `ab -c [5-100] -n 5000 http://132.207.72.37/drupal`
- 28000 lines of code



# Performance (2)

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

min LTTng : all events except execution entry/exit

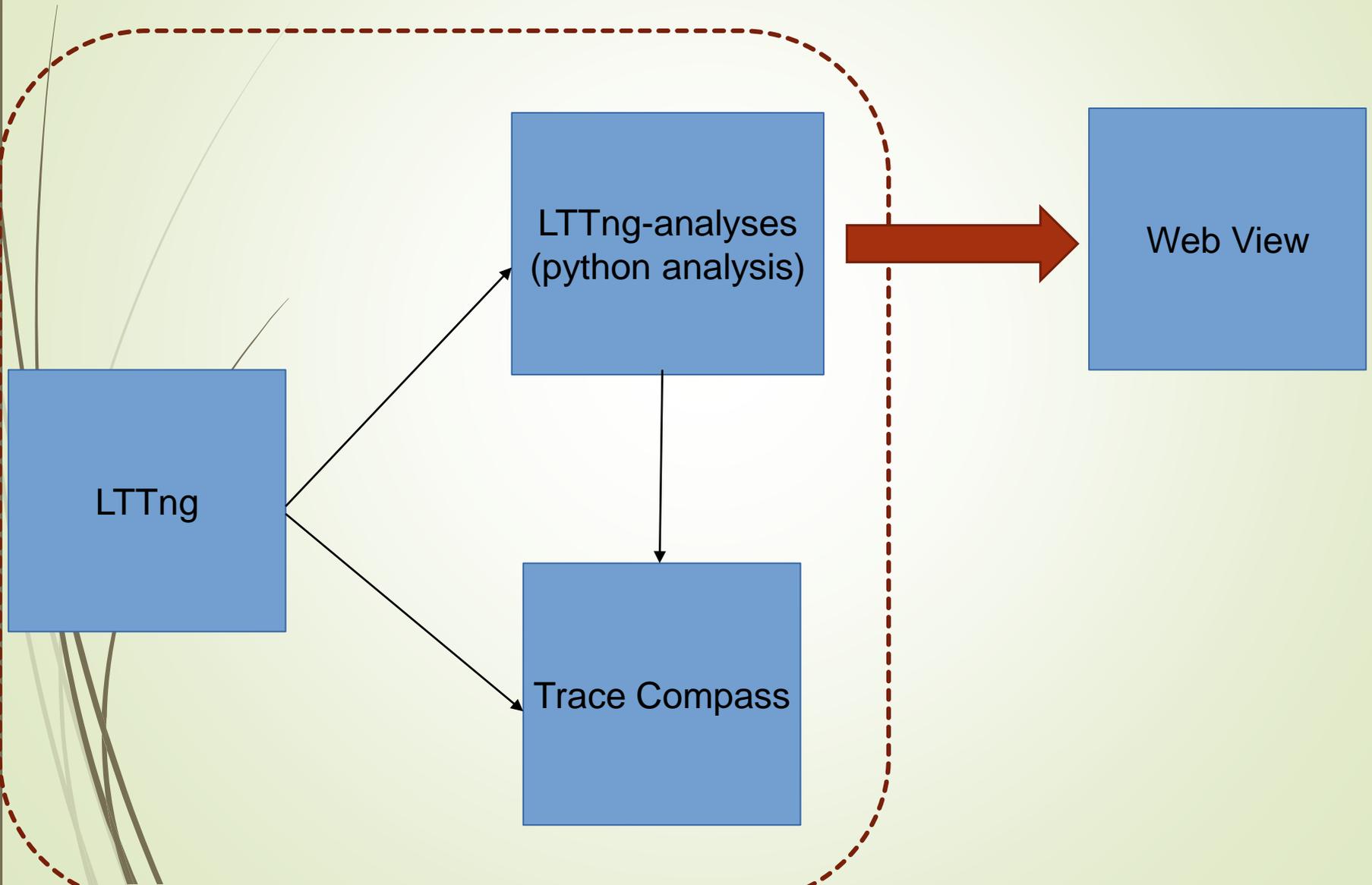




# Demo

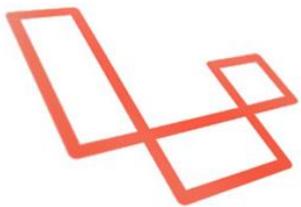
- hsdm slow page loads
  - media wiki
- 

# LTTng Web View



# Main technologies

- Web server base on **Laravel's framework**
- Client side using **Angular 2**
- **ChartJS** for the visualization
- **D3** for flamegraph chart



laravel



Chart.js

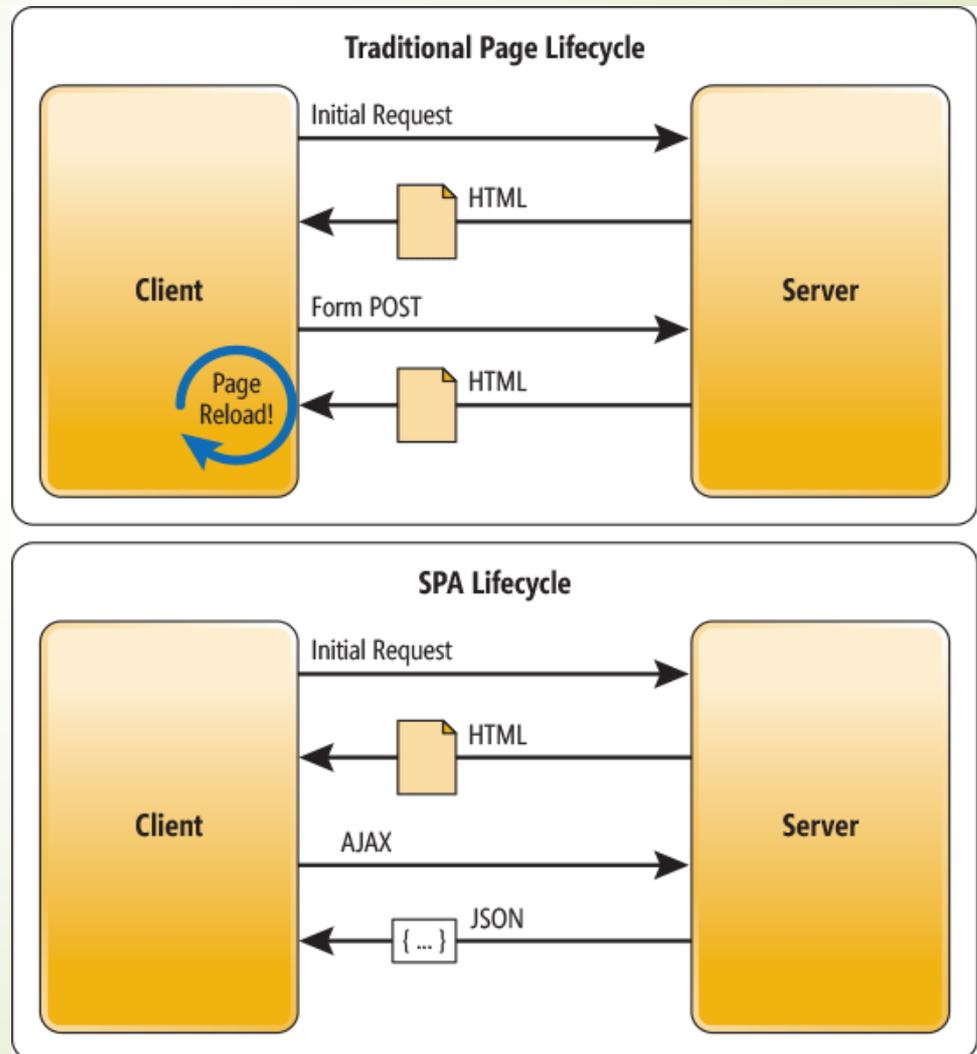
# Architecture

## Problems

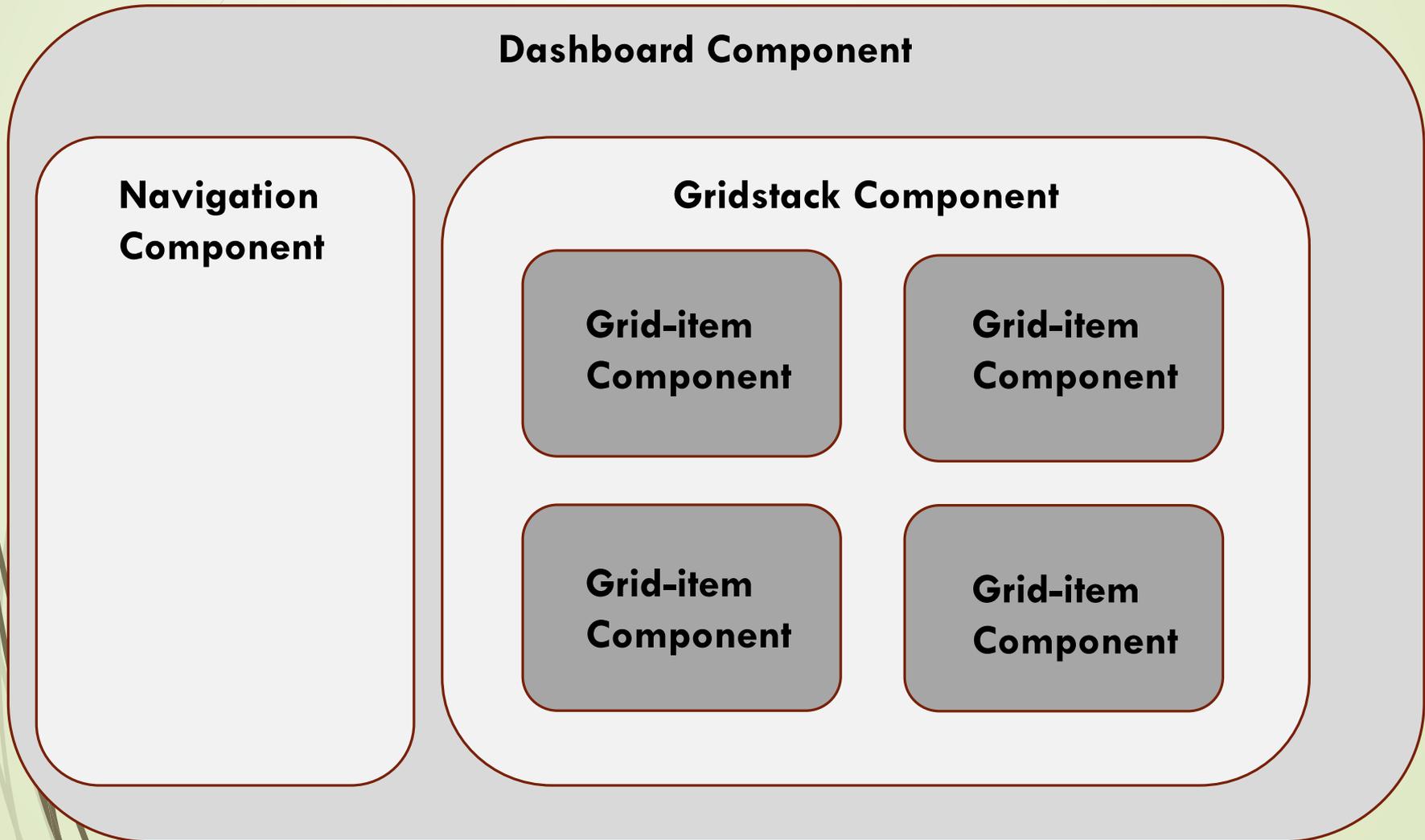
- Loading the DOM is **heavy**
- Some part of the UI must be **reusable**.
- Need a low impact on the **network traffic**

## Solutions

- **Single Page Application**
- DOM is loaded once.  
**Component** are updated via AJAX queries



# Architecture

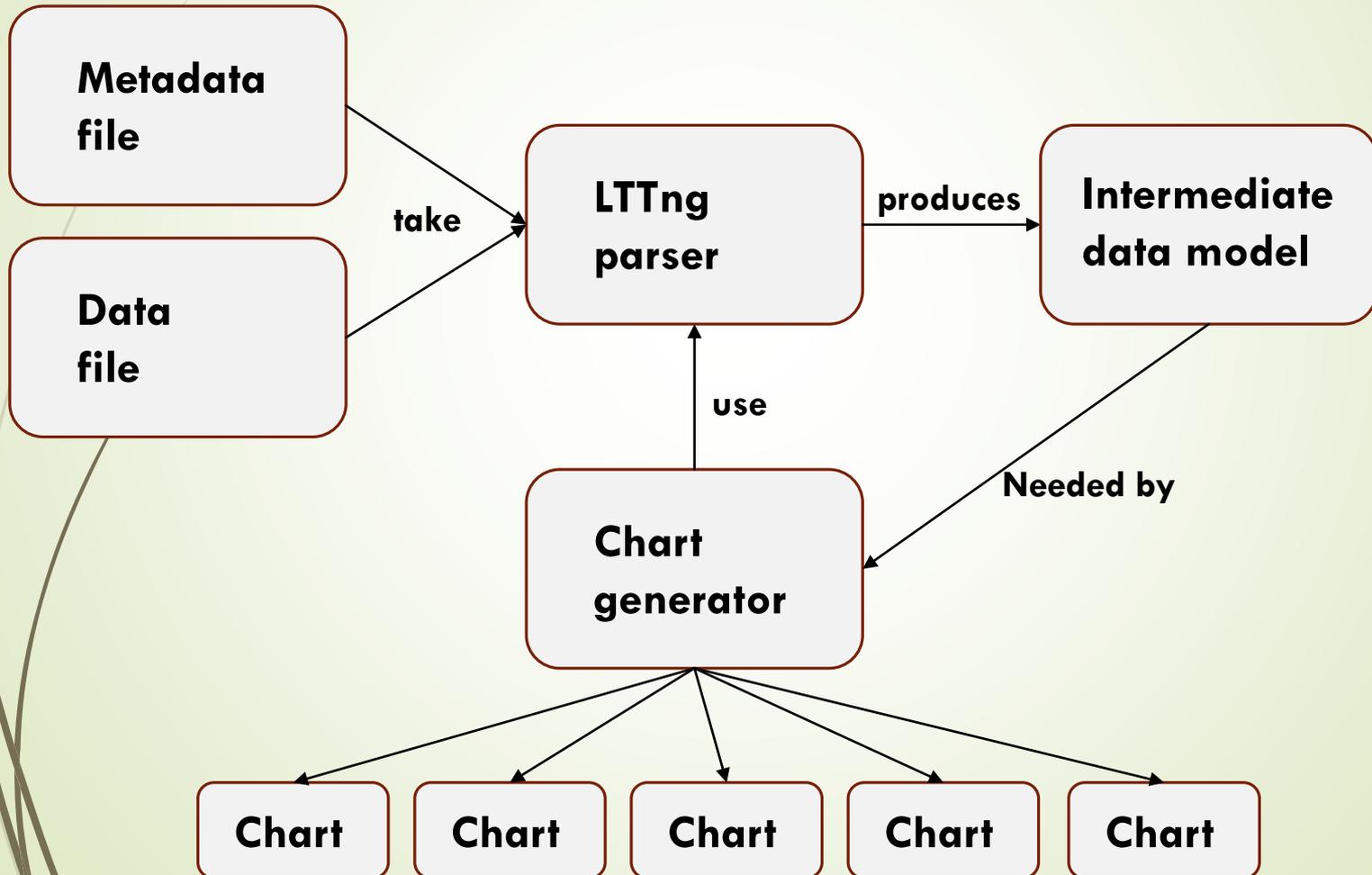




# Architecture

- **Previous component are here only for the layout**
  - Injecting **services** to components to retrieve data;
  - Services send an HTTP request to a **REST API endpoint**
  - Using a **child component** for visualization library;
- **Reducing network load**
  - Using **Gulp** for minifying and uglifying CSS and JS files
  - **POST request** with body object;
  - JSON compression

# Parser architecture





# Demo

- hsdm slow page loads
  - media wiki
- 



# Future Work

- **Official release**
  - **Php and mysql mainlines.**
- **Support other web and database**
- **Web view**
  - **Scalability**
    - **Support large traces**
    - **Nano-scale time units**
  - **Modularity and customization**
    - **Multiple visualization libraries**
  - **Deployment policies!**
    - **Where to host**



# Resources

LTTng-enabled lamp stack modules: <https://github.com/naser/>  
LTTng-analyses for lamp stack traces: <https://github.com/naser/lttnganalyses-for-lampstacktraces>  
XML files: <http://132.207.72.37/xml/>  
Web View: <https://github.com/cheninator/lttng-web-view>

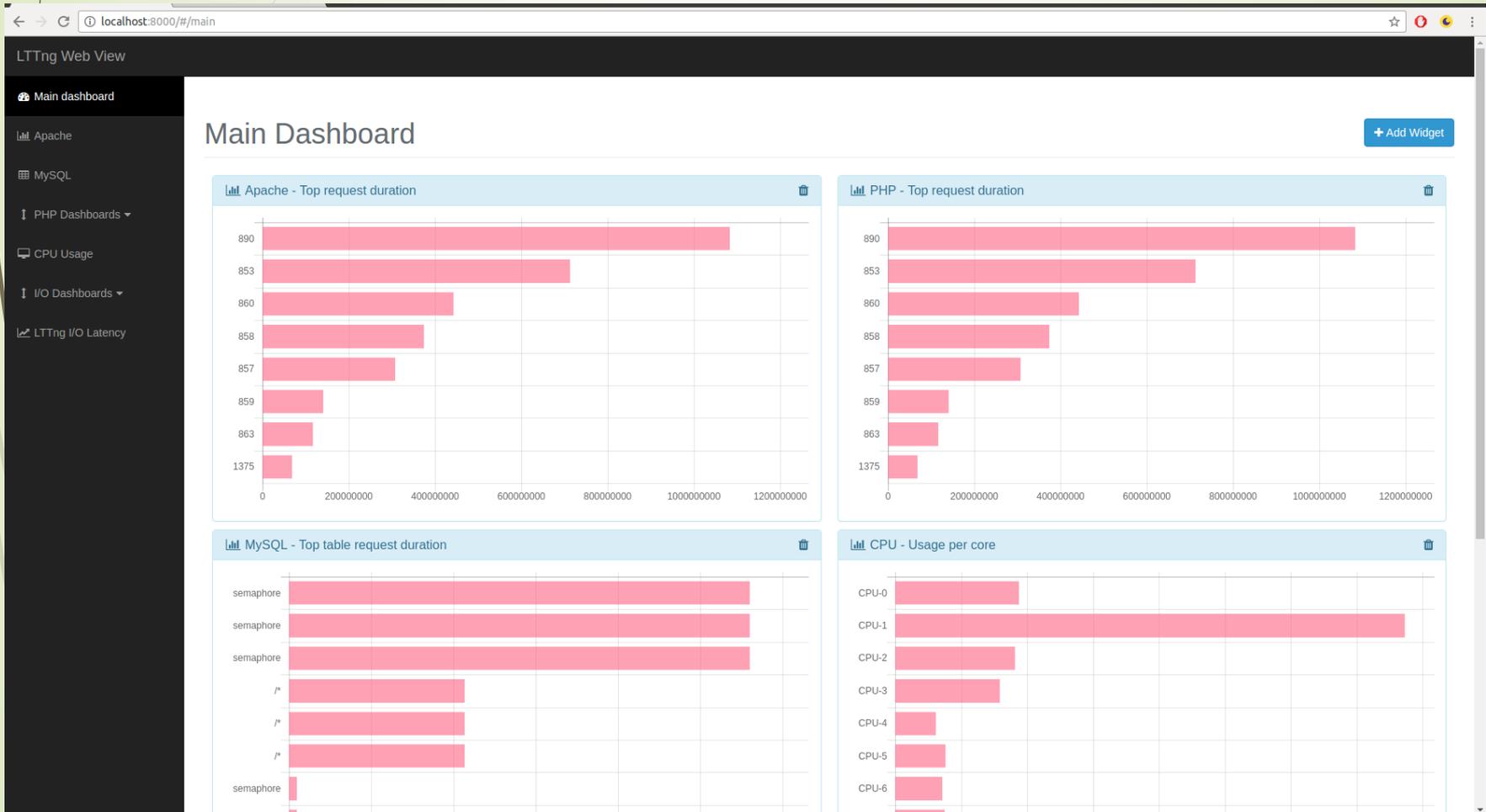


# Questions

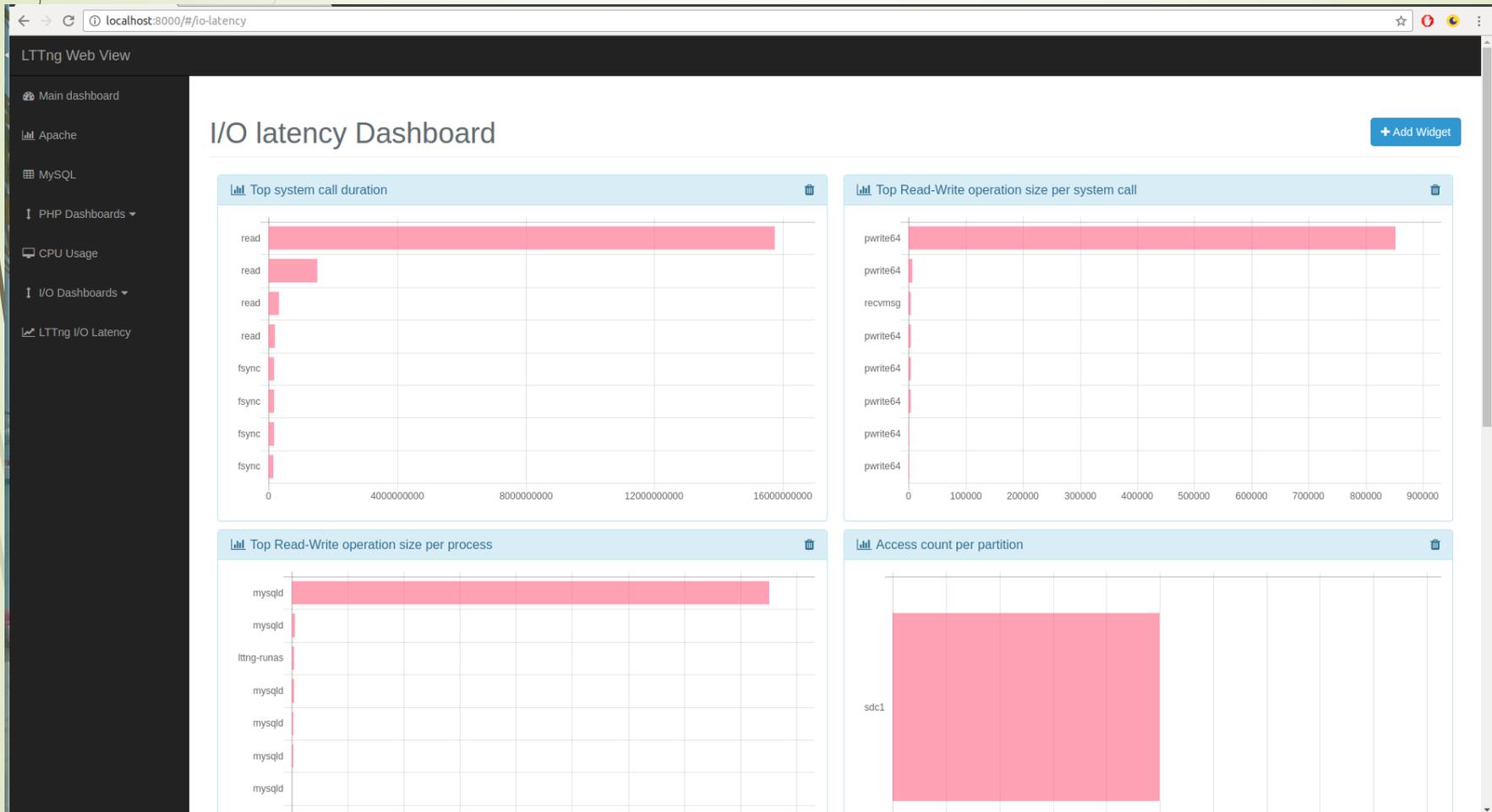


# Other Slides

# Some screenshot



# Some screenshot



# The possibilities

