Shed Lights into Your Web Applications

Naser Ezzati
Polytechnique Montreal

Tracing Summit 2017
Prague, Czech
Motivations

• Challenges: root cause analysis of web applications performance problems
  - Several components and layers are involved
    • Web server problem?
    • Code problem?
    • Bad database design? No table indexes?
    • System resource limitation?
  - Various debugging tools
  - Unified way to analyse them
    - Trace-based approach
    - LAMP stack
      • MEAN stack
Userspace tracing

- You can trace your application
  - tracepoints
  - LTTng-UST
  - FTrace

```c
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

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Channel</th>
<th>CPU</th>
<th>Event type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>20:49:37.86 561</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_close_connection</td>
<td>connection_id=16, context._vld=4236</td>
</tr>
<tr>
<td>20:49:37.86 584</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_suspend_connection</td>
<td>connection_id=16, context._vld=4236</td>
</tr>
<tr>
<td>20:49:37.86 592</td>
<td>s3_6</td>
<td>6</td>
<td>ust_apache_accept_connection</td>
<td>connection_id=20, client_ip=132.207.72.9, client_hostname=(null), local_ip=132.207.72.37, local_hostname=(null)</td>
</tr>
<tr>
<td>20:49:37.88 784</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_accept_connection</td>
<td>connection_id=16, client_ip=132.207.72.9, client_hostname=(null), local_ip=132.207.72.37, local_hostname=(null)</td>
</tr>
<tr>
<td>20:49:37.86 630</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_request_entry</td>
<td>connection_id=20, client_ip=132.207.72.9, method=GET, uri=/drupal, protocol=HTTP/1.0, request_info=reques</td>
</tr>
<tr>
<td>20:49:37.86 640</td>
<td>s3_6</td>
<td>6</td>
<td>ust_apache_request_entry</td>
<td>connection_id=20, client_ip=132.207.72.9, method=GET, uri=/drupal, protocol=HTTP/1.0, request_info=reques</td>
</tr>
<tr>
<td>20:49:37.86 724</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_request_exit</td>
<td>connection_id=16, status=301, context._vld=4236</td>
</tr>
<tr>
<td>20:49:37.86 749</td>
<td>s3_6</td>
<td>6</td>
<td>ust_apache_request_exit</td>
<td>connection_id=20, status=301, context._vld=4244</td>
</tr>
<tr>
<td>20:49:37.86 751</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_close_connection</td>
<td>connection_id=16, context._vld=4236</td>
</tr>
<tr>
<td>20:49:37.86 763</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_suspend_connection</td>
<td>connection_id=16, context._vld=4236</td>
</tr>
<tr>
<td>20:49:37.86 777</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_accept_connection</td>
<td>connection_id=16, client_ip=132.207.72.9, client_hostname=(null), local_ip=132.207.72.37, local_hostname=(null)</td>
</tr>
<tr>
<td>20:49:37.86 781</td>
<td>s3_6</td>
<td>6</td>
<td>ust_apache_close_connection</td>
<td>connection_id=16, client_ip=132.207.72.9, client_hostname=(null), local_ip=132.207.72.37, local_hostname=(null)</td>
</tr>
<tr>
<td>20:49:37.86 798</td>
<td>s3_6</td>
<td>6</td>
<td>ust_apache_close_connection</td>
<td>connection_id=16, client_ip=132.207.72.9, client_hostname=(null), local_ip=132.207.72.37, local_hostname=(null)</td>
</tr>
<tr>
<td>20:49:37.86 809</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_request_entry</td>
<td>connection_id=16, client_ip=132.207.72.9, method=GET, uri=/drupal, protocol=HTTP/1.0, request_info=reques</td>
</tr>
<tr>
<td>20:49:37.86 907</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_request_entry</td>
<td>connection_id=16, client_ip=132.207.72.9, method=GET, uri=/drupal, protocol=HTTP/1.0, request_info=reques</td>
</tr>
<tr>
<td>20:49:37.86 933</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_request_exit</td>
<td>connection_id=16, status=301, context._vld=4236</td>
</tr>
<tr>
<td>20:49:37.86 947</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_request_exit</td>
<td>connection_id=16, status=301, context._vld=4236</td>
</tr>
<tr>
<td>20:49:37.87 884</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_request_exit</td>
<td>connection_id=16, status=301, context._vld=4236</td>
</tr>
<tr>
<td>20:49:37.89 370</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_accept_connection</td>
<td>connection_id=4, client_ip=132.207.72.9, request_info= request:connection_id=4, client_ip=132.207.72.9, method=GET, uri=/drupal, protocol=HTTP/1.0, request_info= reques</td>
</tr>
<tr>
<td>20:49:37.89 408</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_accept_connection</td>
<td>connection_id=4, client_ip=132.207.72.9, request_info= request:connection_id=4, client_ip=132.207.72.9, method=GET, uri=/drupal, protocol=HTTP/1.0, request_info= reques</td>
</tr>
<tr>
<td>20:49:37.89 528</td>
<td>s3_3</td>
<td>3</td>
<td>ust_apache_close_connection</td>
<td>connection_id=4, client_ip=132.207.72.9, request_info= request:connection_id=4, client_ip=132.207.72.9, method=GET, uri=/drupal, protocol=HTTP/1.0, request_info= reques</td>
</tr>
</tbody>
</table>

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

Diagram showing network activity with timestamps from 20:49:31.940 to 20:49:31.980
Apache Modules Tracing

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Channel</th>
<th>CPU</th>
<th>Event type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:12:07.491479 193</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>complete name-handler, src=mod_php7.c, result=0</td>
</tr>
<tr>
<td>11:12:07.492487 607</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>exit name-handler, result=0</td>
</tr>
<tr>
<td>11:12:07.492865 299</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>complete name-process_connection, result=0</td>
</tr>
<tr>
<td>11:12:07.493285 217</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>exit name-protocol_get</td>
</tr>
<tr>
<td>11:12:07.493784 452</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>entry name-protocol_get, result=0</td>
</tr>
<tr>
<td>11:12:07.493974 781</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>exit name-protocol_get, result=0</td>
</tr>
<tr>
<td>11:12:07.494291 250</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>entry name-protocol_get</td>
</tr>
<tr>
<td>11:12:07.494291 447</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>exit name-protocol_get, result=0</td>
</tr>
<tr>
<td>11:12:07.494291 727</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>entry name-log_transaction</td>
</tr>
<tr>
<td>11:12:07.494291 728</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>request_exit id=0, status=200</td>
</tr>
<tr>
<td>11:12:07.494292 703</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>invoke name-log_transaction, src=mod_log_config.c</td>
</tr>
<tr>
<td>11:12:07.494312 568</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>complete name-log_transaction, src=mod_log_config.c, result=0</td>
</tr>
<tr>
<td>11:12:07.494312 599</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>exit name-log_transaction, result=0</td>
</tr>
<tr>
<td>11:12:07.494312 820</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>entry name-suspend_connection</td>
</tr>
<tr>
<td>11:12:07.494320 402</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>exit name-suspend_connection, result=0</td>
</tr>
<tr>
<td>11:12:07.494320 695</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>entry name-resume_connection</td>
</tr>
<tr>
<td>11:12:07.494320 867</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>exit name-resume_connection, result=0</td>
</tr>
<tr>
<td>11:12:07.494337 677</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>entry name-process_connection</td>
</tr>
<tr>
<td>11:12:07.494337 678</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>exit name-process_connection</td>
</tr>
<tr>
<td>11:12:07.494337 679</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>invoke name-process_connection, src=mod_regenerate.c</td>
</tr>
<tr>
<td>11:12:07.494348 398</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>complete name-process_connection, src=mod_regenerate.c, result=1</td>
</tr>
<tr>
<td>11:12:07.494383 849</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>invoke name-process_connection, src=http_core.c</td>
</tr>
<tr>
<td>11:12:07.494393 461</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>entry name-protocol_get</td>
</tr>
<tr>
<td>11:12:07.494393 462</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>exit name-protocol_get, result=0</td>
</tr>
<tr>
<td>11:12:07.494393 510</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>entry name-create_request</td>
</tr>
<tr>
<td>11:12:07.494393 511</td>
<td>channel0_3</td>
<td>3</td>
<td>st_apache:apache_module</td>
<td>invoke name-create_request, src=core.c, result=0</td>
</tr>
</tbody>
</table>
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

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

Installation

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

```bash
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

```php
Hello
<p>Welcome!</p>
Today's date is:

```
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
Installation

./configure --enable-lttng

cmake .

make

sudo make install
Example

Select * from dbtable1;
Example 2

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

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

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

```sql
select * from pfwp_statistics_visitor where id = 273;
(cache_hit)  T: 42,708 ns
```
## 4- LAMP stack analysis

<table>
<thead>
<tr>
<th>Time</th>
<th>IP Address 1</th>
<th>IP Address 2</th>
<th>IP Address 3</th>
<th>IP Address 4</th>
<th>IP Address 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:31:18.635500</td>
<td>/hsdm/</td>
<td>/hsdm/index.php</td>
<td>SELECT 1 FROM blocked_ips WHERE...</td>
<td>SELECT cid, data, create...</td>
<td></td>
</tr>
</tbody>
</table>

### /hsdm/ analysis

- URL requested: /hsdm/index.php
- Query: SELECT 1 FROM blocked_ips WHERE...
- Additional data: SELECT cid, data, create...
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
Usecase: PHP Compile problem

- Multilevel lamp stack view
  - PHP compile problem
- Apache modules
- PHP compile time
- Mysql overhead view
- VM overhead view
Trace Compass Updates

- Incubator
- Multilevel Flame charts
- Multilevel Flame graphs
- VM overhead view
- Critical path using Perf events.
- Counters Analysis
Trace Compass Incubator

• The features that are under development, but still usable enough to be used and tested by users.
• Whose content relates to a specific trace type or domain of analysis (for example virtual machine analyses) and that no other plugin will depend on.
• The features will never be officially released with a specific version
• Some feature may eventually graduate to the Trace Compass project itself if required
  - https://wiki.eclipse.org/Trace_Compass/Contributor_Guidelines
Trace Compass Counters Analysis
Use-case:

PHP OPCODE

Performance Analysis
Response Time of a Web Application

What happens?

There are periodic slow-downs.
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
  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.
• Which one is the longest?
  - It depends!
  - Let's see what trace data gives us.
Compile Time Analysis

request time: 592 us

compile time: 27.4 us

10 11 echo $x.PHP_EOL;
12 ?>

<?php /*main4.php*/
3 $x = rand(0,1000);
6 $xy = 123;
9 include 'folder1/'.$xy.'.php';
11 echo $x.PHP_EOL;
Compile Time: UST Events

30

request time: 592 us

compile time: 27.4 us

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

But, let's go deeper!

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

Now looks much more!

Parse, Compile, Optimize (55us)  Execute (5.2us)
Solution: Opcode Cache (Opcache)

Cache at first run

Load from cache after
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

What happens?
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
Resources

Modules to install:
  https://github.com/naser

Traces and XML files:
  https://github.com/naser/tracingsumit2017

Trace Compass:
  http://tracecompass.org

LTTng:
  http://lttng.org/

Thank you
n.ezzati@polymtl.ca
Other Slides

- Python Analysis