eZ publish and performance

LinuxTag 2004

June 24th, 2004. Karlsruhe, Germany

Derick Rethans <dr (at) ez (dot) no>

Questions

Questions?

Challenge: Improve eZ publish 3.3 performance

- Too much memory usage
- Too much CPU usage

Timing points:

Checkpoint	Elapsed	Rel. Elapsed	Memory	Rel. Memory
Script start	0.0000 sec	0.0517 sec	1,535.9063KB	1,827.2813KB
Module start 'content'	0.0517 sec	0.1870 sec	3,363.1875KB	6,943.0234KB
Module end 'content'	0.2387 sec	0.0402 sec	10,306.2109KB	863.1328KB
End	0.2789 sec		11,169.3438KB	0.8429KB
Total runtime:	**0.2791 sec**			

Accelerating: APC

- Acceleraters store PHP 'Byte Code'
- Shared Memory
- Faster Execution

Timing points:

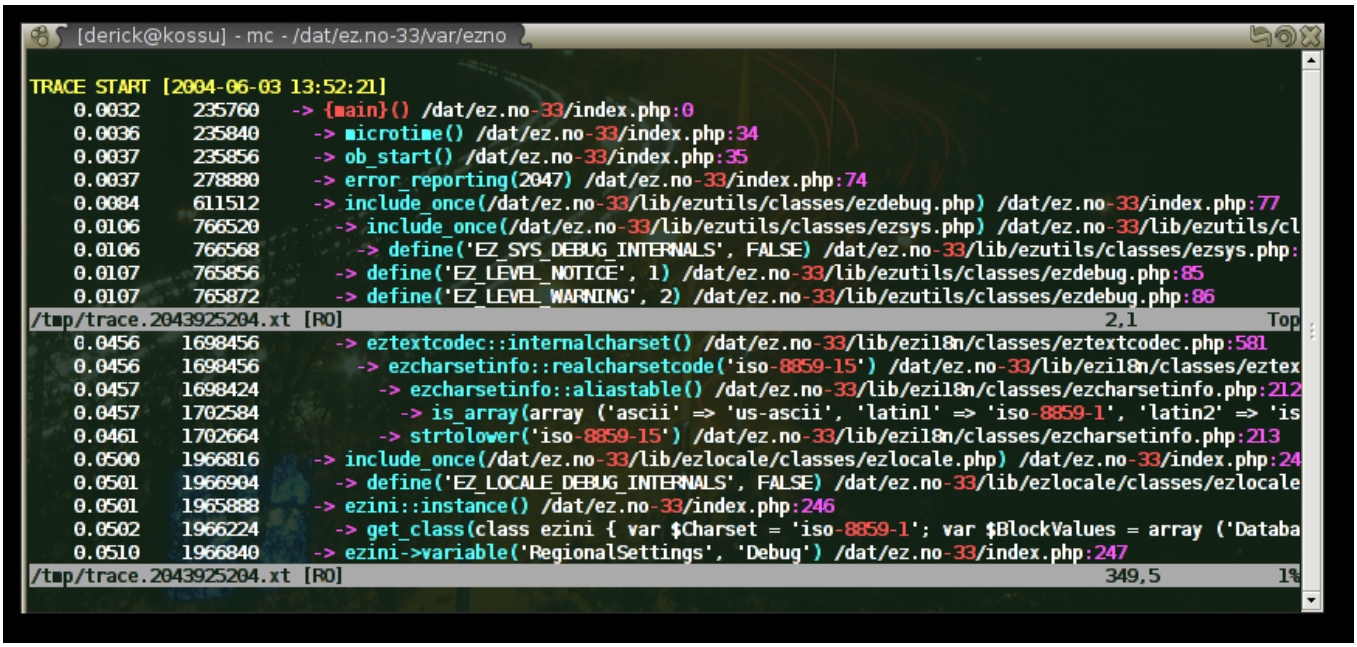
Checkpoint	Elapsed	Rel. Elapsed	Memory	Rel. Memory
Script start	0.0000 sec	0.0492 sec	182.4609KB	265.5313KB
Module start 'content'	0.0492 sec	0.0409 sec	447.9922KB	741.5781KB
Module end 'content'	0.0901 sec	0.0204 sec	1,189.5703KB	245.1563KB
End	0.1106 sec		1,434.7266KB	0.2394KB
Total runtime:	**0.1108 sec**			

The Hunt for Speed

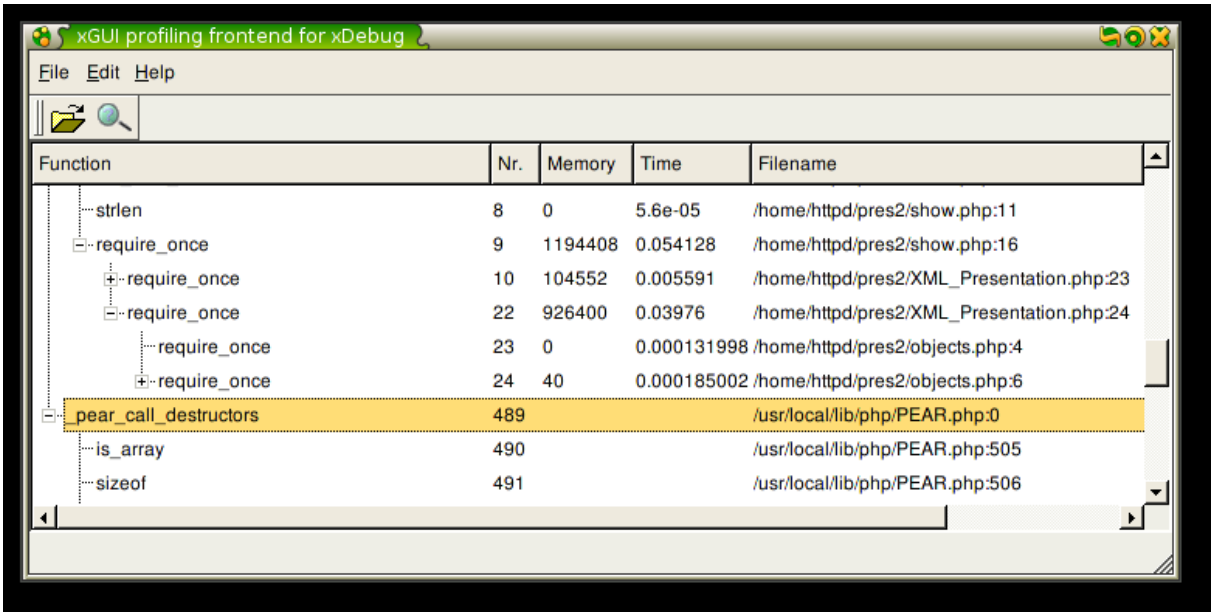
- Xdebug: a tool to debug PHP
- Tracing function calls
- Profiling Applications

Tracing the Problems

- Xdebug's Trace Features
- Trace function calls
- Provides details on memory usage

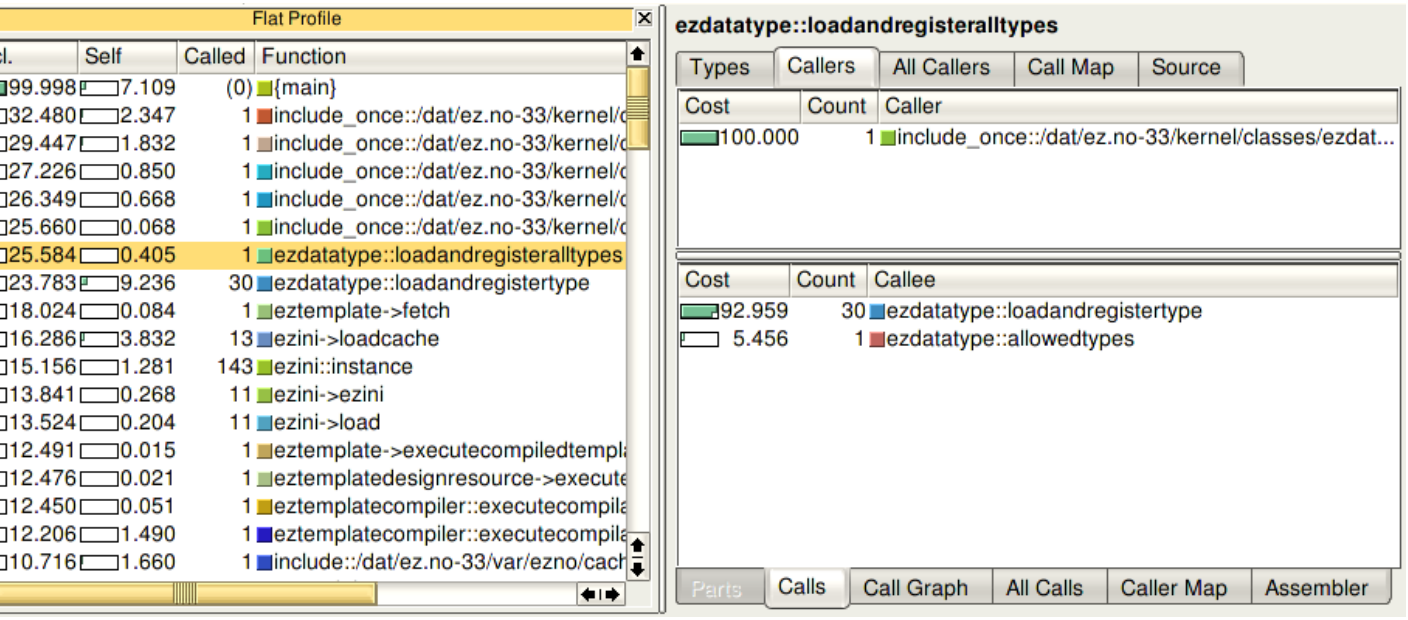


- Simple Tracefile Browser



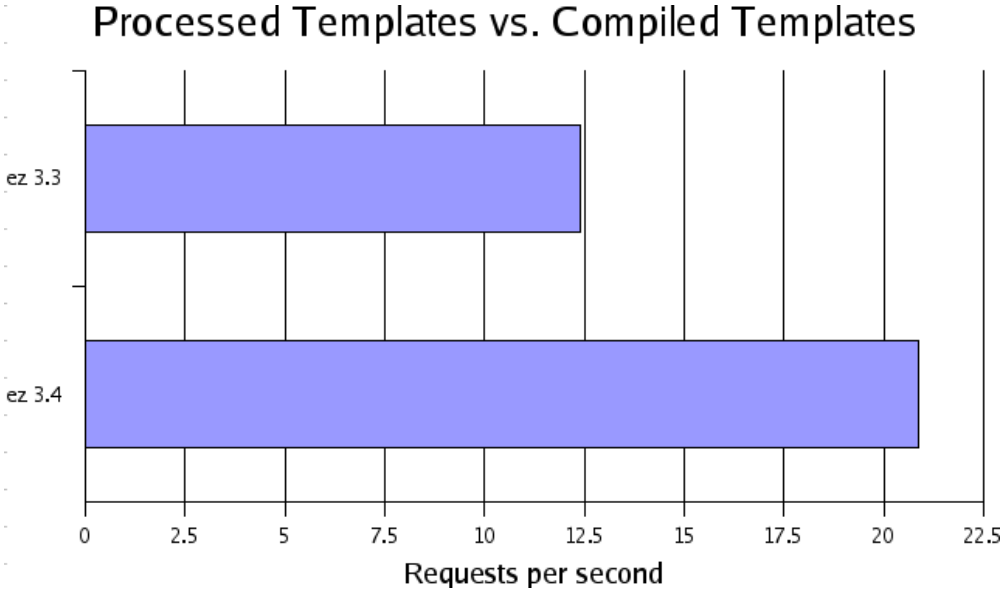
Finding the Bottleneck

- Xdebug's Profiling Features
- GUI analysing with KCachegrind
- Provides details on CPU usage



Speeding up

- Processed Templates est mort
- Viva la Compiled Templates



Reducing Memory Usage

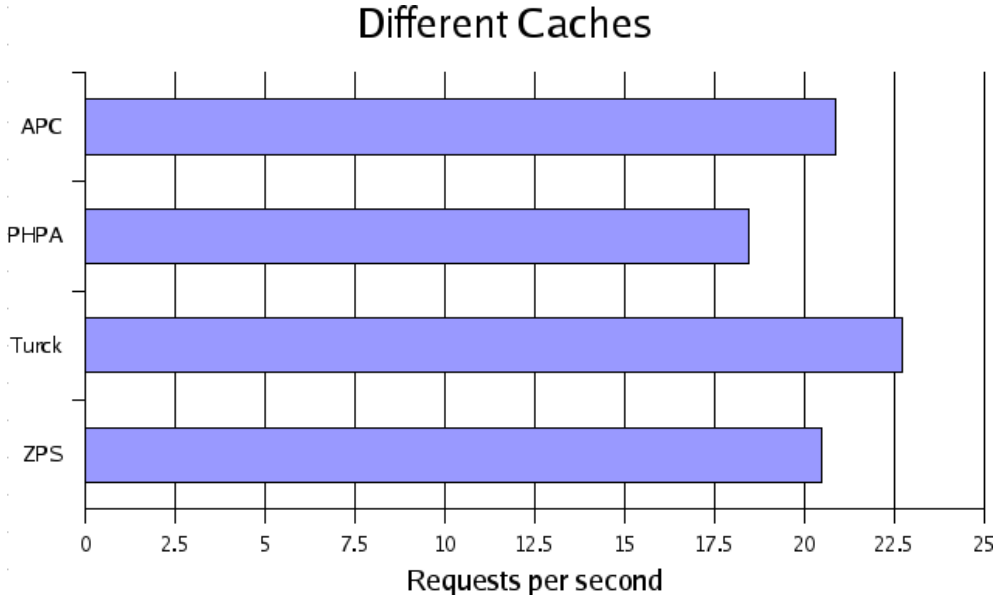
- Processed vs. Compiled Templates
- Loading Data Types on demand
- From 11MB to 7MB

```
function &create( $dataTypeString )
 {
   $types =& $GLOBALS["eZDataTypes"];
   $def = null;
+  if ( !isset( $types[$dataTypeString] ) )
+  {
+    eZDataType::loadAndRegisterType($dataTypeString);
+  }
+
   if ( isset( $types[$dataTypeString] ) )
   {
     $className = $types[$dataTypeString];
@@ -709,6 +716,4 @@
   var $Name;
 }

-eZDataType::loadAndRegisterAllTypes();
```

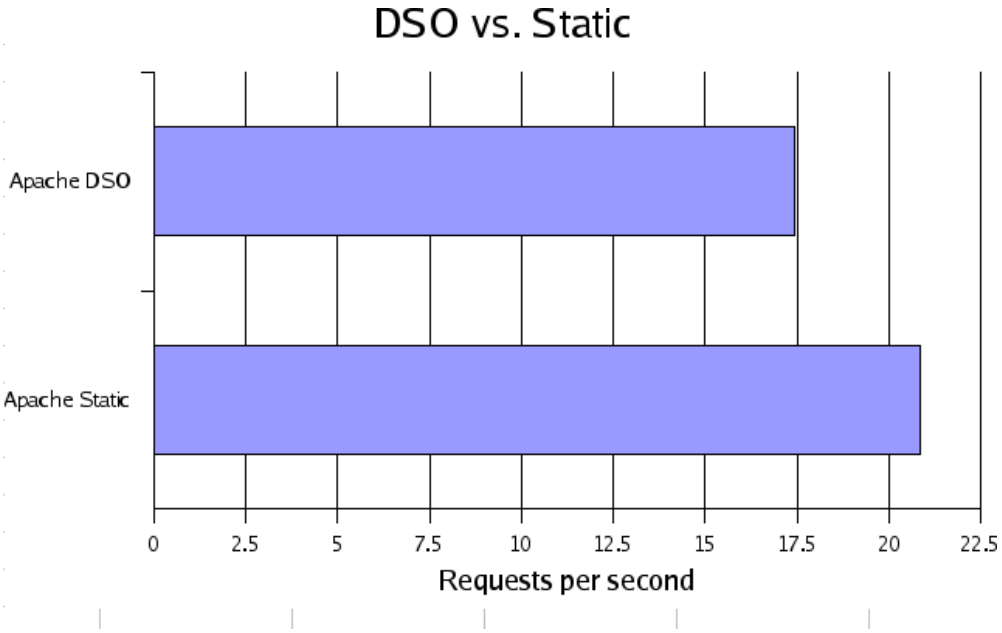
Accelerating More

- Turck MM Cache
- PHP Accelerator
- Zend Performance Suite



Apache Is Slow!

- When you're using PHP as DSO module
- So use static compiled modules
- or compile with --prefer-non-pic



- Optimized ./configure
- Memory Limit
- Shorter Path
- Use include() in a smarter way
- Ramdisk

Static Squids

- Reverse Proxy
- Tied into Apache
- Static Generated
- Huge Performance Boost
- No 'Dynamic' Content

Preview Caching

- Generated on Publish
- Static Generated
- First 'hit' is fast

```
[ContentSettings]
PreViewCache=enabled
```

No Indexes for Now!

- Search Engine
- Indexing takes Time
- Delay Indexing to Cron Job
- Publishing is Faster

```
[SearchSettings]
DelayedIndexing=enabled
```

For the Future

- Static .html generation part of eZ publish
- Further Optimizations (Template Compiler, Memory Usage)
- Clustered Object Caching
- SRM

Resources

These Slides: http://pres.derickrethans.nl

Xdebug site: http://xdebug.org

Non-pic: http://news.php.net/php.internals/9828

eZ publish: http://ez.no

Questions?: dr (at) ez (dot) no

Index