High Performance XQuery Processing in PHP with Zorba

Vikram Vaswani
November 2012

Vikram Vaswani

Founder, Melonfire

- Product design and implementation with open source technologies
- 20+ products launched for customers worldwide

Author and active PHP community participant

- 7 books for McGraw-Hill USA
- 250+ tutorials for IBM dW, Zend Developer Zone and others

More information at http://vikram-vaswani.in

XML Support in PHP

- XML processing: SimpleXML and XMLReader
- XML transformation: XSLT and XML-DAS
- Event-based parsing: SAX
- Tree-based parsing: DOM
- Serialization: WDDX
- Dynamic documents: XMLWriter

What's Missing?

Introducing Zorba

- General purpose XQuery processor for PHP 5.3+
- Supports XQuery v1.0 and related specifications
- Available for Windows, Linux, and Mac OS X
- Open source; Apache 2.0 License
- Project supported by FLWOR Foundation and Oracle.

Supported XML Technologies

- XPath
- XQuery v1.0 and v3.0
- XQuery Scripting
- Xquery Data Definition
- XPath Full-Text 1.0
- XSLT 1.0
- XqueryX (limited)

PHP Installation

- Linux and MacOS X
  - Libxml2: http://xmlsoft.org/
  - Iconv: http://www.gnu.org/software/libiconv/
  - libUUID: http://linux.die.net/man/3/libuuid
  - ICU (C version): http://www.icu-project.org/
  - Xerces (C version): http://xerces.apache.org/xerces-c/
  - GCC and related make/cmake utilities
- Windows
  - https://launchpad.net/zorba/+download
- API Wrapper
  - http://zorba.s3.amazonaws.com/ZorbaPHP.zip

PHP Installation

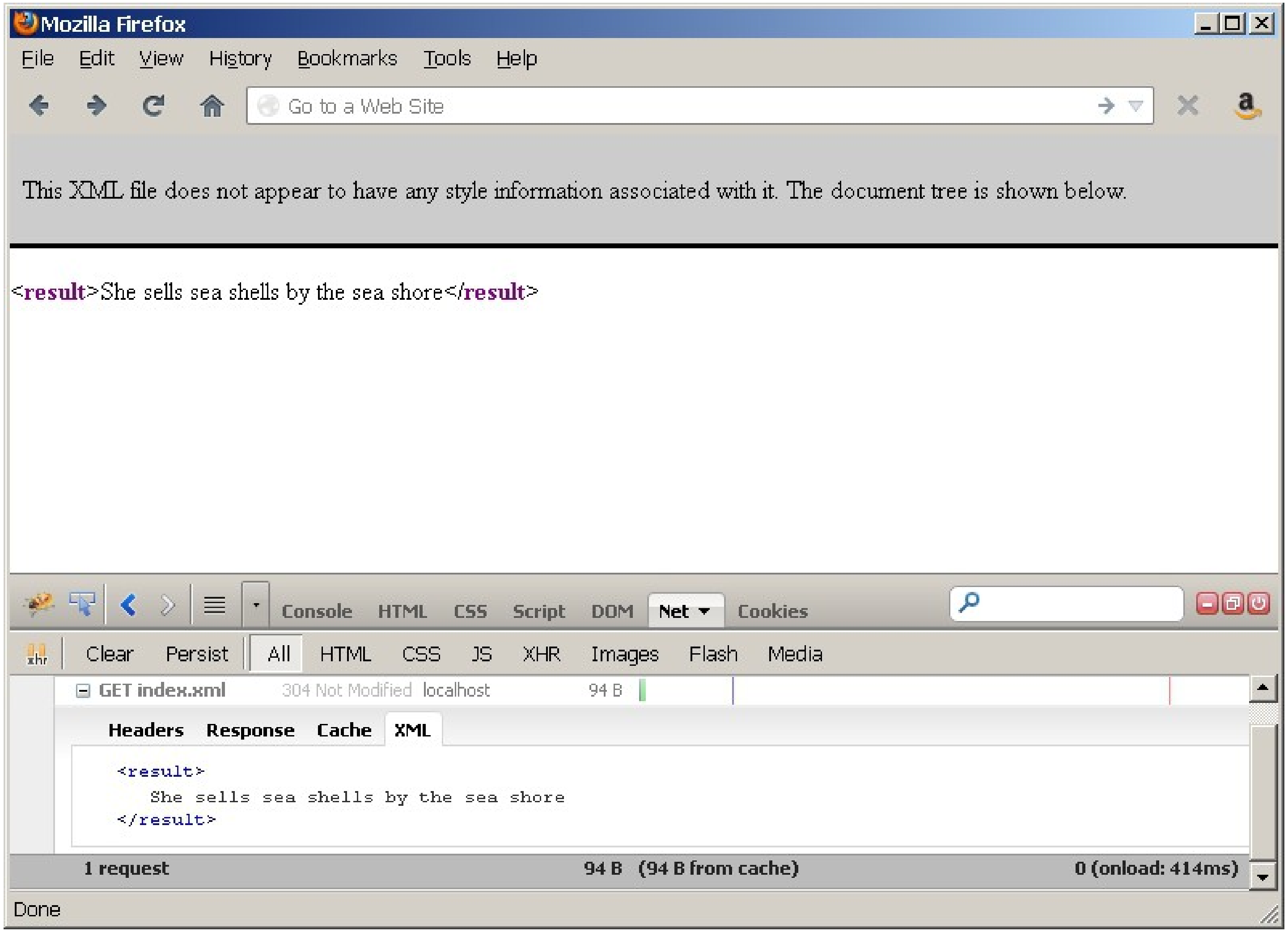
- Linux and MacOS X
  - Libxml2: http://xmlsoft.org/
  - Iconv: http://www.gnu.org/software/libiconv/
  - libUUID: http://linux.die.net/man/3/libuuid
  - ICU (C version): http://www.icu-project.org/
  - Xerces (C version): http://xerces.apache.org/xerces-c/
  - GCC and related make/cmake utilities
- Windows
  - https://launchpad.net/zorba/+download
- API Wrapper
  - http://zorba.s3.amazonaws.com/ZorbaPHP.zip

Basic Usage

```
<?php
require_once 'Zorba/XQueryProcessor.php';
$zorba = new XQueryProcessor();
try {
  $queryStr = <<<END
  let \$message := 'She sells sea shells by the sea shore'
  return
  <result>{\$message}</result>
END;
  $query = $zorba->importQuery($queryStr);
  $result = $zorba->execute();
  echo $result;
} catch (Exception $e) {
  die('ERROR:' . $e->getMessage());
}
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

Basic Usage


Mozilla Firefox
File Edit View History Bookmarks Tools Help
Go to a Web Site
Console HTML CSS Script DOM Net Cookies
Clear Persist All HTML CSS JS XHR Images Flash Media
Headers Response Cache XML

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<result>She sells sea shells by the sea shore</result>
```

```
GET index.xml    304 Not Modified  localhost    94 B
```

```
<result>
   She sells sea shells by the sea shore
</result>
```

```
1 request     94 B  (94 B from cache)     0 (onload: 414ms)
```

Done

Usage Examples

Arithmetic Operators

```
<?php
require_once 'Zorba/XQueryProcessor.php';
$zorba = new XQueryProcessor();
$queryStr = <<<END
  declare variable \$x := 30;
  declare variable \$y := 12;
  let \$r := \$x + \$y
  return
  <result> {\$r} </result>
END;
$query = $zorba->importQuery($queryStr);
echo $zorba->execute();
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

Sequences

```php
<?php
require_once 'Zorba/XQueryProcessor.php';
$zorba = new XQueryProcessor();
$queryStr = <<<END
  let \$dolls := ('Echo', 'Victor', 'Sierra', 'November', 'Alpha')
  return
  <results>
  {
    for \$doll in \$dolls
    return
    <name>{\$doll}</name>
  }
  </results>
END;
$query = $zorba->importQuery($queryStr);
echo $zorba->execute();
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

String Functions

```
<?php
require_once 'Zorba/XQueryProcessor.php';
$zorba = new XQueryProcessor();
$queryStr = <<<END
  let \$dolls := ('Echo', 'Victor', 'Sierra', 'November', 'Alpha')
  return
  <results>
  {string-join(\$dolls, ', ')}
  </results>
END;
$query = $zorba->importQuery($queryStr);
echo $zorba->execute();
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

Math Functions

```
<?php
require_once 'Zorba/XQueryProcessor.php';
$zorba = new XQueryProcessor();
$queryStr = <<<END
  let \$set := (1 to 20)
  return
  <results>
  {
    for \$i in \$set
    where (\$i mod 2 eq 0)
    return
    <value> {\$i} </value>
  }
  </results>
END;
$query = $zorba->importQuery($queryStr);
echo $zorba->execute();
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

Data Filtering

(XML source)

```xml
<?xml version="1.0"?>
<data>
  <record>
    <code>ABW</code>
    <name>Aruba</name>
    <continent>North America</continent>
    <year></year>
    <population>103000</population>
    <gnp>828.00</gnp>
    <capital>
      <name>Oranjestad</name>
      <population>29034</population>
    </capital>
  </record>
  ...
</data>
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

Data Filtering (Simple)

```php
<?php
require_once 'Zorba/XQueryProcessor.php';
$zorba = new XQueryProcessor();
$dm = $zorba->getXmlDataManager();
$dm->loadDocument('data.xml', file_get_contents('data.xml'));
$queryStr = <<< END
  for \$record in doc('data.xml')//record
  let \$name := \$record/name/text()
  let \$code := \$record/code/text()
  return
  <name code="{\$code}">{\$name}</name>
END;
$query = $zorba->importQuery($queryStr);
echo $zorba->execute();
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

Data Filtering (Complex)

```php
<?php
require_once 'Zorba/XQueryProcessor.php';
$zorba = new XQueryProcessor();
dm = $zorba->getXmlDataManager();
$dm->loadDocument('data.xml', file_get_contents('data.xml'));
$queryStr = <<< END
  for \$record in doc('data.xml')//record
  let \$name := \$record/name/text()
  let \$code := \$record/code/text()
  let \$population := \$record/population/text()
  where contains(\$record/continent/text(), 'Europe')
  where (xs:integer(\$population) gt 1000000)
  order by \$population descending
  return
  <name code="{\$code}" population="{\$population}">{\$name}</name>
END;
$query = $zorba->importQuery($queryStr);
echo $zorba->execute();
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

Node Manipulation

(XML source, before)

```
<?xml version='1.0'?>
<heroes>
  <hero>
    <name>Spiderman</name>
    <alterego>Peter Parker</alterego>
  </hero>
  <hero>
    <name>Superman</name>
    <alterego>Clark Kent</alterego>
  </hero>
  <hero>
    <name>The Flash</name>
    <alterego>Wally West</alterego>
  </hero>
</heroes>
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

Node Manipulation

```
<?php
require_once 'Zorba/XQueryProcessor.php';
$zorba = new XQueryProcessor();
$dm = $zorba->getXmlDataManager();
$dm->loadDocument('heroes.xml', file_get_contents('heroes.xml'));
$queryStr = <<< END
  delete node doc('heroes.xml')//heroes/hero[last()],
  replace node doc('heroes.xml')//heroes/hero[last()]
  with
  <hero>
    <name>The Incredible Hulk</name>
    <alterego>Bruce Banner</alterego>
  </hero>
END;
$query = $zorba->importQuery($queryStr);
echo $zorba->execute();
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

Node Manipulation

```
(XML source, after)
<?xml version="1.0" encoding="UTF-8"?>
<heroes>
  <hero>
    <name>Spiderman</name>
    <alterego>Peter Parker</alterego>
  </hero>
  <hero>
    <name>Superman</name>
    <alterego>Clark Kent</alterego>
  </hero>
  <hero>
    <name>The Incredible Hulk</name>
    <alterego>Bruce Banner</alterego>
  </hero>
</heroes>
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

REST Parser

```
<?php
require_once 'Zorba/XQueryProcessor.php';
$zorba = new XQueryProcessor();
$queryStr = <<< END
  import module namespace zorba-rest =
    "http://www.zorba-xquery.com/zorba/rest-functions";
  for \$post in
    zorba-rest:get("http://user:pass@api.del.icio.us/v1/posts/recent")//posts/post
  where contains(\$post/@tag, 'book')
  return
  <result>
    <href> {string(\$post/@href)} </href>
    <description> {string(\$post/@description)} </description>
    <tags> {string(\$post/@tag)} </tags>
  </result>
END;
$query = $zorba->importQuery($queryStr);
echo $zorba->execute();
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

REST Parser

(After)

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <href>http://www.everythingphpmysql.com/</href>
  <description>How to do Everything with PHP & MySQL - Vikram Vaswani</description>
  <tags>book development php mysql web beginner</tags>
</result>
<result>
  <href>http://www.mysql-tcr.com/</href>
  <description>MySQL: The Complete Reference - Vikram Vaswani</description>
  <tags>mysql php book</tags>
</result>
```

JSON Parser

(JSON source)

```
{
  "books": [{
    "title":"Phantom Prey",
    "author":"John Sandford",
    "price":"7.99"
  }, {
    "title":"A Most Wanted Man",
    "author":"John le Carre",
    "price":"8.99"
  },{
    "title":"Beach Babylon",
    "author":"Imogen Edward-Jones",
    "price":"12.99"
  }]
}
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

JSON Parser

```
<?php
require_once 'Zorba/XQueryProcessor.php';
$zorba = new XQueryProcessor();
$queryStr = <<< END
  import module namespace json =
      "http://www.zorba-xquery.com/zorba/json-functions";
    let \$x := json:parse('{$json}')
    for \$i in \$x//item
    where (xs:decimal(\$i/pair[@name='price']/text()) lt 10.00)
    return
    <result> {\$i/pair[@name='title']/text()} </result>
END;
$query = $zorba->importQuery($queryStr);
echo $zorba->execute();
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

JSON Parser

```
<?php
require_once 'Zorba/XQueryProcessor.php';
$zorba = new XQueryProcessor();
$queryStr = <<< END
  import module namespace json =
      "http://www.zorba-xquery.com/zorba/json-functions";
    let \$x := json:parse('{$json}')
    for \$i in \$x//item
    where (xs:decimal(\$i/pair[@name='price']/text()) lt 10.00)
    return
    <result> {\$i/pair[@name='title']/text()} </result>
END;
$query = $zorba->importQuery($queryStr);
echo $zorba->execute();
```

Credit: Vikram Vaswani, "Building XQuery-powered applications with PHP and Zorba"
http://www.ibm.com/developerworks/library/x-zorba/index.html

Performance

Scenario

- Data
  - 1 day of forecast data in XML format: 727MB
- Workflow
  - Process data for a specific site
  - Send selected temperatures to clients
  - Display chart

Credit: William Candillon, "Cutting Edge Data Processing with PHP & XQuery"
http://www.slideshare.net/wcandillon/cutting-edge-data-processing-with-php-xquery

SimpleXML

```php
<?php
$siteId = 3;

$forecasts = simplexml_load_file('forecasts.xml');
$forecasts = $forecasts->xpath(
  "/forecast-list/forecast[@site-id='$siteId']");

foreach($forecasts as$forecast) {
  $time = $forecast->xpath("@time-step");
  $value = $forecast->xpath(
    "//weather-elements/weather-element"
    ."[@name = 'ScreenTemperature']/text()");
  echo "<temperature time='" . $time[0] ."'value='" = . $value[0] . "' />\n";
}
?>
```

Credit: William Candillon, "Cutting Edge Data Processing with PHP & XQuery"
http://www.slideshare.net/wcandillon/cutting-edge-data-processing-with-php-xquery

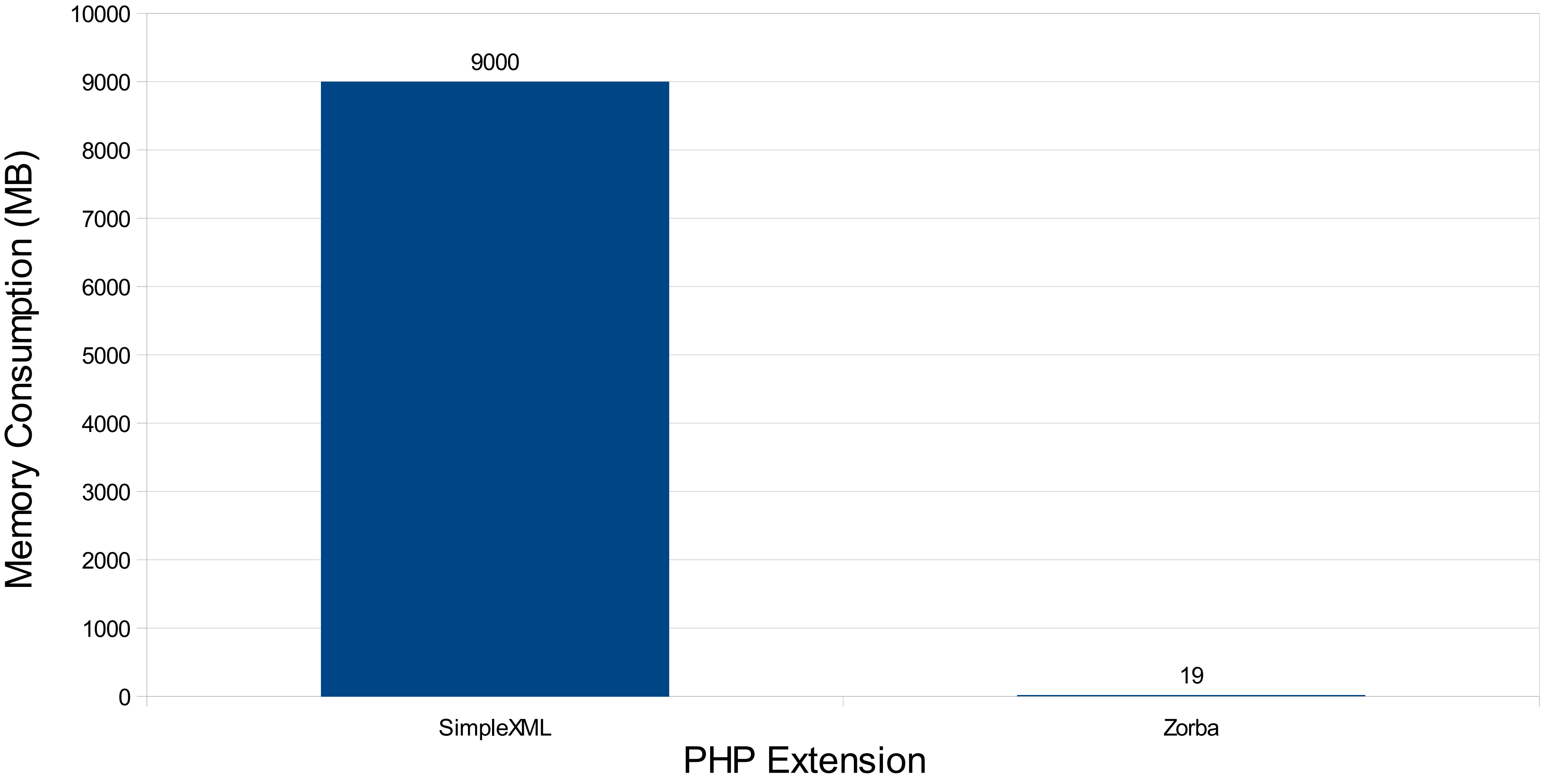
Zorba

```
<?php
require_once 'Zorba/XQueryProcessor.php';
$zorba = new XQueryProcessor();

$queryStr = <<<END
  for $forecast inz:parse-xml(file:read-text("forecasts.xml"),
    <opt:options>
    <opt:parseExternalParsedEntity opt:skipRootNodes="1"/>
    </opt:options>)
  where $forecast/@site-id = "3"
  let $time:= string($forecast/@time-step)
  let $value:= $forecast/weather-elements/weather-element
    [@name = 'ScreenTemperature']/text()
  return
  <temperaturetime="{$time}" value="{$value}" />
END;
$query = $zorba->importQuery($queryStr);
echo $zorba->execute();
```

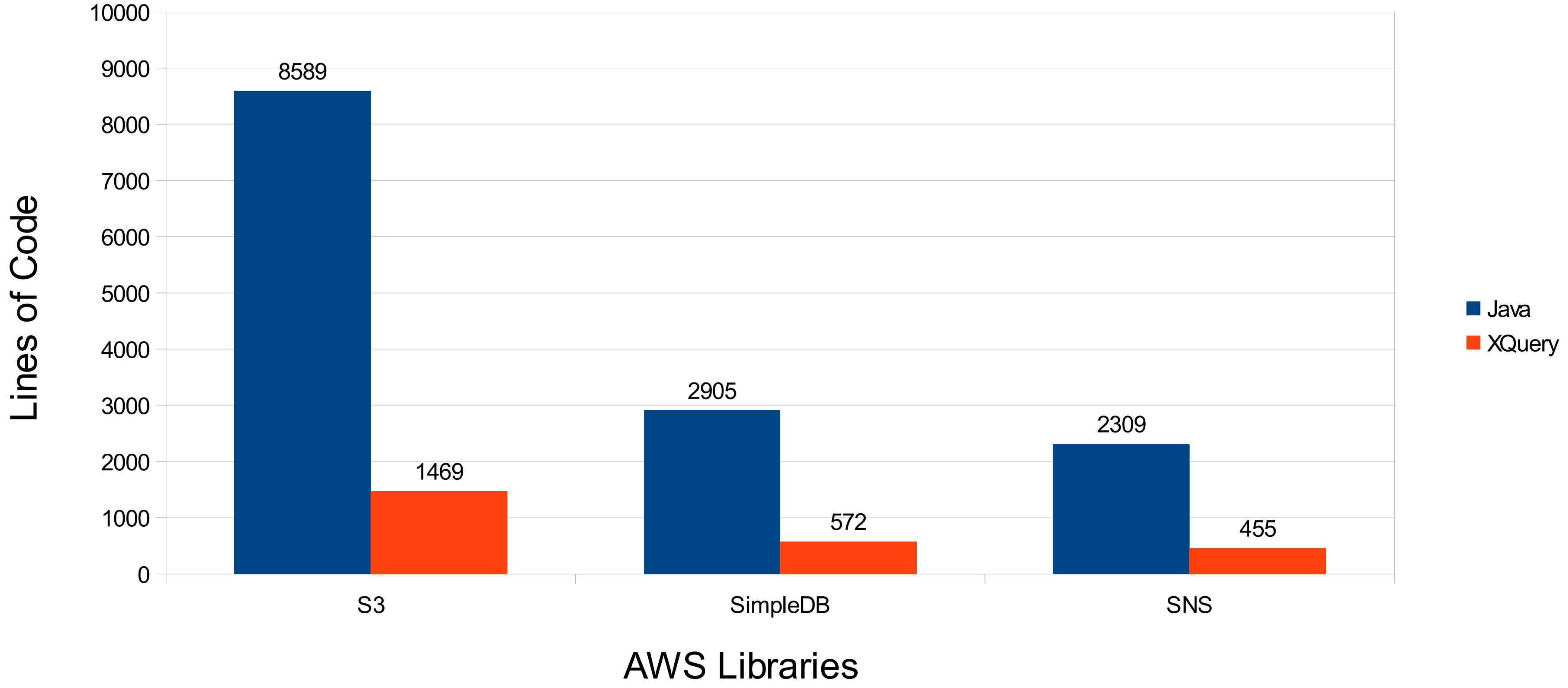
Credit: William Candillon, "Cutting Edge Data Processing with PHP & XQuery"
http://www.slideshare.net/wcandillon/cutting-edge-data-processing-with-php-xquery

Results



Credit: William Candillon, "Cutting Edge Data Processing with PHP & XQuery"
http://www.slideshare.net/wcandillon/cutting-edge-data-processing-with-php-xquery

AWS Libraries



Credit: William Candillon, "Cutting Edge Data Processing with PHP & XQuery"
http://www.slideshare.net/wcandillon/cutting-edge-data-processing-with-php-xquery

Resources

- Usage docs: zorba-xquery.com/html/documentation
- Live demo: zorba-xquery.com/html/demo
- Source code: code.launchpad.net/~zorba-coders/zorba/trunk
- Modules: zorba-xquery.com/html/modules
- Toolkit: xqdt.org
- Forums: zorba-users [at] googlegroups [dot] com
- News: zorba-xquery.com/html/blog
- Twitter: @wcandillon

Questions?

Contact Information

Email:

- vikram-vaswani.in/contact

Web:

- www.melonfire.com
- vikram-vaswani.in

Social networks:

- plus.google.com/100028886433648406825
- twitter.com/melonfire