

MySQL in an Enterprise Search Architecture

Clay Webster

CNET Networks, Inc.

(clay@cnet.com)



April 21, 2005

Co-presented by O'Reilly Media, Inc. and MySQL AB

CNET Case Study

- Our Problem
- How Does MySQL Fit?
- How CNET Made a Search Service
- Strengths
- Our Issues
- Conclusions



CNET Networks, Inc.



Release 1.0
the conversation starts here



The Problem

- Search Vendor Changes
- Product End-of-Life
- Very High Replacement Costs
- Learning New Tech and Integration
- Yep. We need it “Faster, Better, Cheaper”.



Some of Our Requirements

- Sorting and Good Text Searching
- Fast Content Updates and Distribution
- Standalone Service Speaking HTTP and Providing XML
- Adequate Capacity and Scalability
- Inexpensive
- Rapid Development



Why Think of MySQL?

Technology

- “Fulltext Searching” and Sorting
- Replicate Content
- Source Code
- Robust and Customer-Hardened

Economics

- Search Vendor RFP
- Low Cost
- Easy Learning Curve



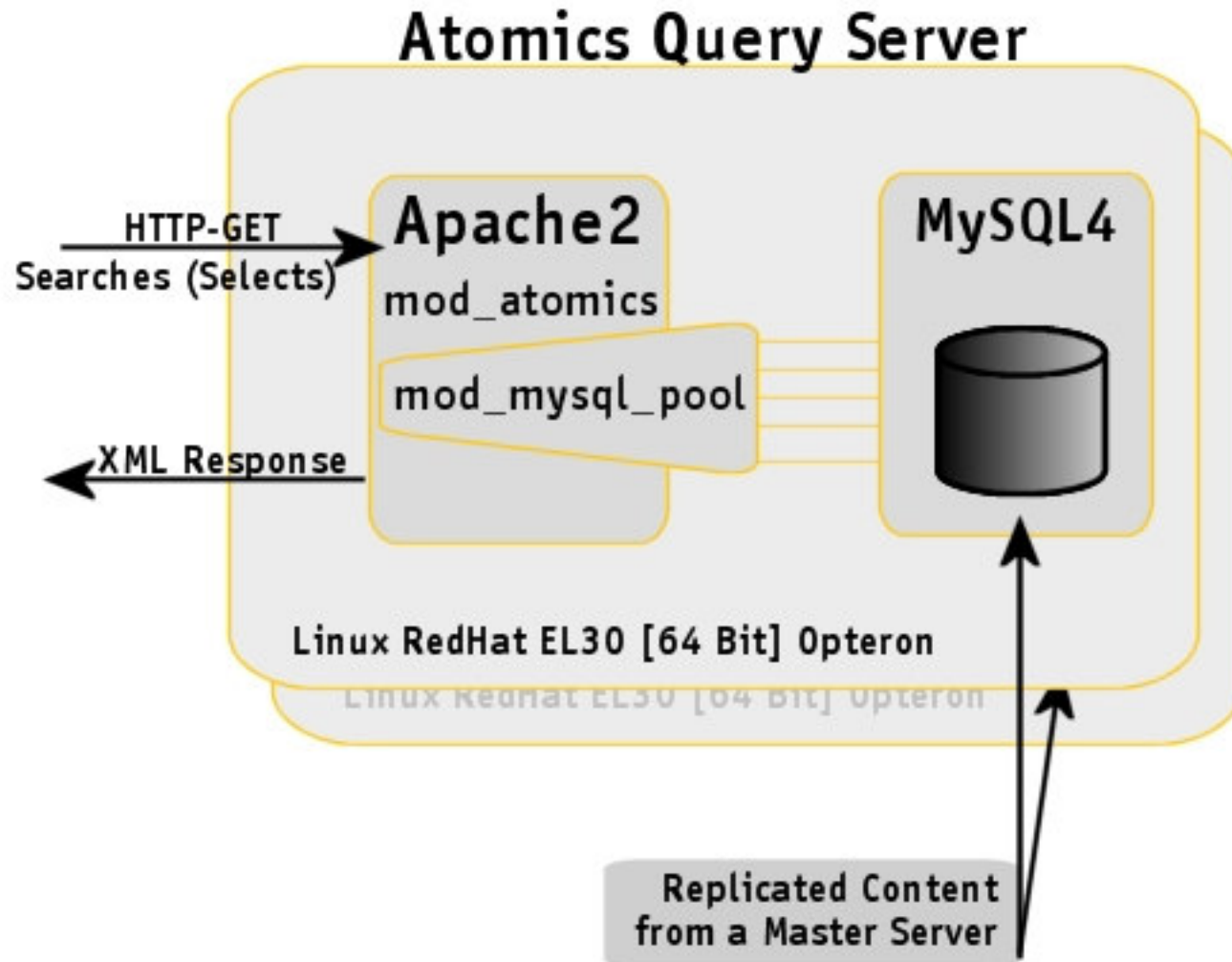
Our Internal Product

ATOMICS

Apache TO MySQL In CNET Search



How CNET Made a “Search Service”



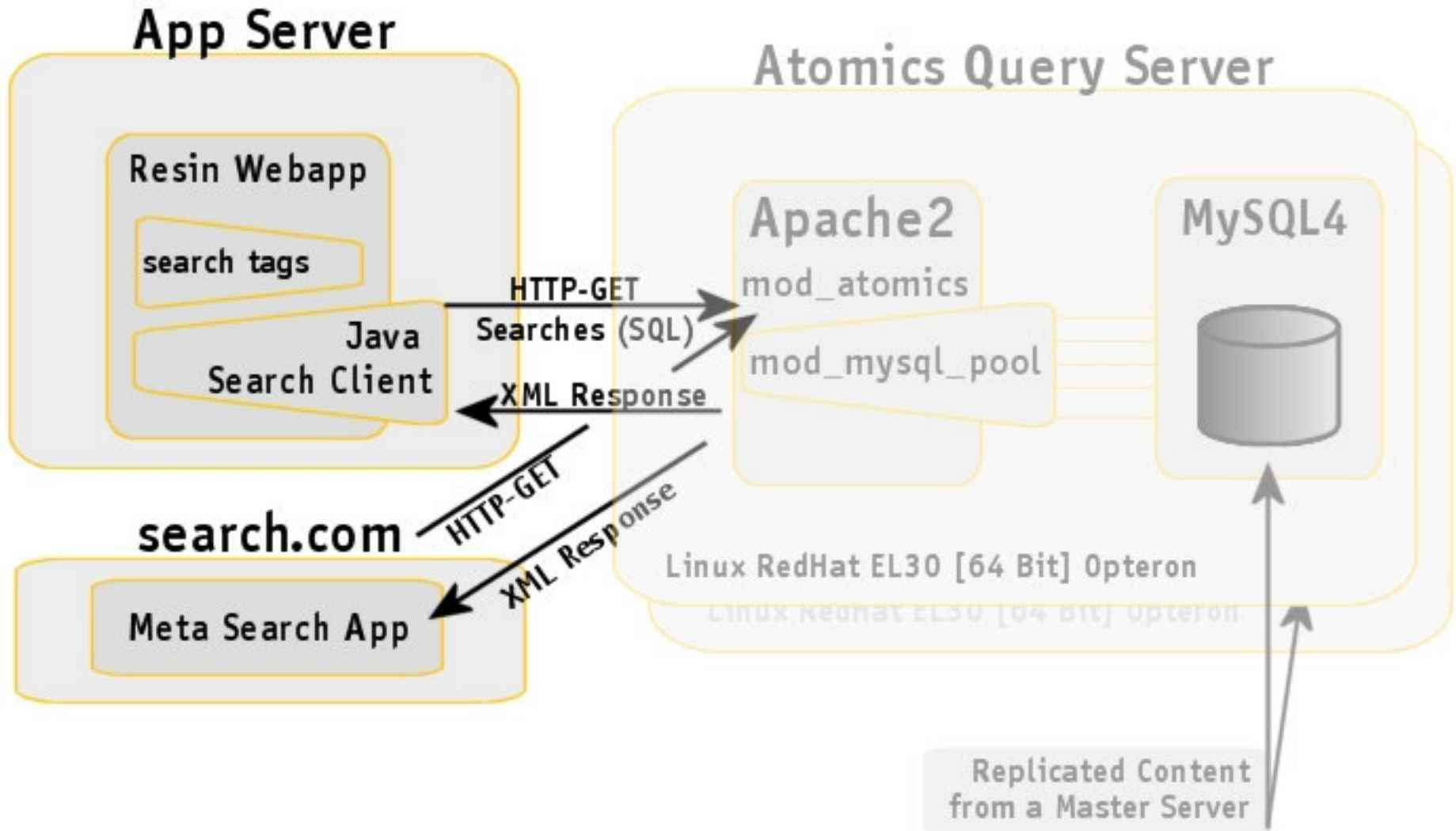
Why Apache At All?

Mostly all “CNET” Reasons:

- Our HTTP Approach
- XML Out
- Logging and Status
- Other Goodies



How Do We Query?



atomics admin page - Mozilla Firefox

File Edit View Go Bookmarks Tools Help Fair, 52°F 32°F 63°F

http://c10-gdl-at-qry3.cnet.com:8531/admin/


Atomics Admin (artistindexdb)

ATOMICS
Apache To MySQL In CNET Search

Atomics	[STATUS]
c10-gdl-at-qry3.cnet.com:8531	
Real Apache:	[HARDWARE] [TRAFFIC] [PROBLEMS] [PROC STATUS] [INFO]
c10-gdl-at-qry3.cnet.com:8531	
MySQL Server:	[HARDWARE] [TRAFFIC] [PROBLEMS] [PROC STATUS] [STATUS] [MYSQL HISTORY]
c10-gdl-at-qry3.cnet.com:11010	

Make a Query	[SCHEMA EXPLORER] [TABLE STATUS] [CONFIG] [OPEN TABLES] [FULL SQL INTERFACE]
<pre>SELECT * FROM artistindex ORDER BY docId DESC</pre>	
<input type="button" value="Search"/>	

Assistance	[DOCUMENTATION] [FILE A BUGZILLA] [SEND EMAIL]
<i>Fri Apr 15, 2005 03:37:38 PM</i>	



Atomics SQL Interfaces - Mozilla Firefox

File Edit View Go Bookmarks Tools Help Fair, 52°F 32°F 63°F

http://c10-gdl-at-gry3.cnet.com:8531/admin/form.html

Atomics SQL Interfaces

ATOMICS
Apache To MySQL In CNET Search

/select mode

SQL Statement	<pre>SELECT SQL_CALC_FOUND_ROWS productId, addDate, name, shortDescription, MATCH (name,expressURL) AGAINST ('love song') as score FROM artistindex WHERE MATCH (name,expressURL) AGAINST ('love song') ORDER BY score DESC, addDate DESC</pre>
Return Number Found	<input checked="" type="checkbox"/> (SQL SELECT statement should contain SQL_CALC_FOUND_ROWS.)
Protocol Version	1
Start Row	0
Maximum Rows Returned	20
<input type="button" value="Search"/>	

/raw mode

SQL Statement	<pre>SELECT COUNT(*) FROM artistindex</pre>
Protocol Version	1
<input type="button" value="Search"/>	

Atomics Search Results - Mozilla Firefox

File Edit View Go Bookmarks Tools Help Fair, 52°F 32°F 63°F

http://c10-gdl-at-qry3.cnet.com:8531/select/?q=SELECT+SQL_CALC_FOUND_ROWS+prod

Atomics Search Results

ATOMICS
Apache To MySQL In CNET Search

Status: 0
Records Found: 222
Number of Fields: 5
Records Returned: 20
Query time: 5 (ms)

productId	addDate	name	shortDescription	score
100270965		Love Spit Love 2365	Love Spit Love - How Soon Is Now? [Originally From "The Smiths"]	8.8227500915527
100049511		Paal Nilssen-Love		8.8227500915527
100041826		Marcus Love & Love Inc.		8.8227500915527
100040402		Love Spit Love		8.5022525787354
100588944	2005-01-06 19:46:23	Song Destruction	A guy playing an instrument. A guy recording some music. A guy that prefers to record instrumental music. A guy that just does this as a hobby.	8.2230520248413
100214124		The Culling Song	The Culling Song is a 5-piece rock band from Oxford, Ohio.	8.2230520248413
100375464		a streetlight song	alternate prog rock band... 3 piece... emotional and epic songs... use of standard rock and experimental methods and instruments...	8.2230520248413

Atomics Search Results - M

File Edit View Go Bookmarks

Atomics Search

Status: 0
Records Found: 222
Number of Fields: 5
Records Returned: 20
Query time: 5 (ms)

productId
100270965
100049511
100041826
100040402
100588944
100214124
100375464

Source of: http://c10-gdl-at-qry3.cnet.com:8531/select?q=SELECT+SQL_CALC_FOUND_ROWS+prod...

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="/admin/tabular.xsl"?>
<response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://pi.cnet.com/cnet-search/response.xsd">
  <responseHeader>
    <status>0</status>
    <numFound>222</numFound>
    <numFields>5</numFields>
    <numRecords>20</numRecords>
    <QTime>5</QTime>
  </responseHeader>
  <responseBody>
    <record>
      <field type="integer">
        <name>productId</name>
        <value>100270965</value>
      </field>
      <field type="date">
        <name>addDate</name>
        <value null="1"/>
      </field>
      <field type="string">
        <name>name</name>
        <value>Love Spit Love 2365</value>
      </field>
      <field type="string">
        <name>shortDescription</name>
        <value>Love Spit Love - How Soon Is Now?
(Originally From "The Smiths")</value>
      </field>
      <field type="float">
        <name>score</name>
        <value>8.8227500915527</value>
      </field>
    </record>
    <record>
      <field type="integer">
        <name>productId</name>
        <value>100049511</value>
      </field>
      <field type="date">

```

How Do We Populate DBs?

Our Index Building (from Scratch):

- ContentDB => MySQL Import File => Import, Build DB Indices

Our Incremental Changes:

- JDBC-based to MySQL which does INSERT, UPDATE, DELETE, or REPLACE
- Only to a Single Master Server
- Replication Distributes



Strengths

- Searches Can Be Blindingly Fast
 - “Natural Language” and BOOLEAN MODE
- Parallel Service
 - Content Replication to Client-Slaves
 - Auto-Recovery of Changes
 - HTTP Load Balancing Makes Capacity Upgrades Linear with Hardware Additions
- No Huge Vendor Capital Expense



Strengths ⁽²⁾

- It's a Database!
 - Changes are Reflected Immediately
 - Ultimately Flexible
 - Queries are Transparent and Clear
- We Get Upgrades
 - Apache and MySQL
- It Does Not Crash*



Strengths ⁽³⁾

- Widely Known Leveraged Tech
 - Open Components and Specifications
 - SQL, Schema Design, and MySQL
 - Apache, Apache modules
 - HTTP, XML, XPP3
 - Easy to Expand and Add Features



Our Issues

- Extra Work and Thought
 - Creativity
 - Transformation Example: Stemming
- “Natural Language” vs. BOOLEAN MODE

```
(4 * (MATCH (title) against ('warez' in boolean mode)) +  
 2 * (MATCH (author) against ('warez' in boolean mode)) +  
  MATCH (title,dek,body,author) against ('warez' in  
  boolean mode))  
) as score
```



Our Issues ⁽²⁾

- Complexity, Space, and Speed
- Retrieval Blind Spots
 - Minimum Word Lengths
 - Non-Alpha-Numerics (plus '_')
 - C++ vs. C#
 - AT&T
 - Wi-Fi



Conclusions?

- Atomics is a Workhorse and Very Capable
- Retrieval Issues
- Meets >80% Our Search Needs
- Scoring Algorithms – and Boolean Mode vs. Normal Mode
- Very Flexible





Co-presented by O'Reilly Media, Inc. and MySQL AB