UIMA scale-out with MapReduce using Apache Hadoop

- UIMA meets Hadoop -

Speaker: Marc Hofer, mail@marc-hofer.de
Supervisors: Thilo Goetz, tgoetz@de.ibm.com
           Prof. Dr. Theel, h.theel@fhtw-berlin.de
           Prof. Dr. Stanierowski, stani@fhtw-berlin.de
Agenda

• Introduction
  • UIMA – Unstructured Information Management Architecture
• Goal
• Technology
  • MapReduce
• UIMA tasks
• Performance results
  • One of the UIMA tasks in detail
  • Summary
• Conclusion
Introduction:
UIMA – Unstructured Information Management Architecture

• „A component framework for analyzing unstructured content such as text, audio and video„
• Amount of unstructured data is growing not only on the internet, but also in enterprises
• Challenges: analyzing and structuring large volumes of unstructured information.
• Goals:
  • Finding information
  • Finding information quickly
• Experiments refer to text analysis
Goal

• Analyzing large quantities of text-data with UIMA

• Approach: Linking Hadoop with UIMA
  • Does it work?
  • Complexity of administration of a Hadoop cluster
  • Duration of deployment of UIMA tasks
  • Overhead of Hadoop
  • Restrictions of Hadoop in combination with UIMA
  • Performance tests
Technology: MapReduce

Not directly applicable on UIMA, and not “necessary” (we only use the mapper)

Pseudo-Code:

```java
run(){
Default Hadoop Configuration
}
//called only once per node
Configure(JobConfiguration){
Initializing UIMA-Framework
}
//called for each document
Map(exactly one Document){
Complete Processing of the Document
Writing Result Data to Hadoop Distributed File System
}
```
UIMA tasks

• Emphasis on regular expressions
  • Accent on CPU power

• Without regular expressions
  • “Opposite” of “Emphasis on regular expressions”, focus rather on hard disk and memory
  • WhitespaceTokenizer
  • Hidden Markov Model (HMM) tagger

• Most realistic use case
  • Combination of antecedents
Cluster setup – hardware and software

- CPU: Intel Core 2 Duo E6600, 2400 MHz
- Storage device: Western Digital WDC WD800ADFS-75SLR2, 80 GBytes
- Memory (RAM): 2 * 1024 Mbytes, DDR2 (PC2-6400)
- Network card: Broadcom NetXtreme 57xx Gigabit Controller, 1000 Mbps
- Motherboard: Dell 0HR330
- Five routers: 100Mbps interfaces

- OS: Linux 2.6.18.2-34-default (x86)
- Java: JDK 1.6.0 06
- UIMA 2.2.1
- Hadoop 0.16.4
Cluster setup – processed data

- Wikipedia XML dump partly split up
- Files between 4,000 and 25,500 lines
- Numbers of lines were chosen by chance (realistic scenario)
- Smallest file: 46,215 Bytes, largest file: 3,918,466 Bytes, average file size: 954,539 Bytes
- In total: 6,000 files, 5,727,231,903 Bytes (~5,33 GB)
Performance results – most realistic use case

<table>
<thead>
<tr>
<th>Number of machines</th>
<th>Master is slave</th>
<th>Dedicated Master</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>27 min 58.707 sec</td>
<td>27 min 50.202 sec</td>
</tr>
<tr>
<td>10</td>
<td>51 min 0.098 sec</td>
<td>56 min 6.759 sec</td>
</tr>
<tr>
<td>5</td>
<td>1 hr 39 min 4.550 sec</td>
<td>2 hrs 2 min 24.276 sec</td>
</tr>
<tr>
<td>2</td>
<td>4 hrs 4 min 3.330 sec</td>
<td>8 hrs 4 min 16.973 sec</td>
</tr>
<tr>
<td>HadoopUIMA-standalone</td>
<td>8 hrs 5 min 58.076 sec</td>
<td></td>
</tr>
<tr>
<td>UIMA-standalone</td>
<td>14 hrs 40 min 8.217 sec</td>
<td></td>
</tr>
</tbody>
</table>

- Scales well
- Differences between dedicated and master is slave runs
- Hadoop uses 2 cores
Performance results - summary

Dedicated master

- Most realistic use case
- Without regular expressions
- Emphasis regular expressions

Processing time (sec) vs Number of machines
Conclusion

• It works!
• Integration of (uncomplex) UIMA tasks is straightforward
• Scales well

• Further advancement:
  • Small files
  • Complex UIMA tasks or whole workflows
  • And so on
Suggestions, questions?

- mail@marc-hofer.de

Thank you for your attention.