Statically Inferring Performance Properties of Software Configurations

Chi Li, Shu Wang, Henry Hoffmann, Shan Lu
Configurations Explosion

Which configuration affects performance?

- **Stack Overflow #36170959, Cassandra Performance Tuning**
  
  "Please let me know what more settings I can tweak to get maximum performance out of my cluster."

- **Stack Overflow #47665640, Memory configurations**
  
  “I am finding that I am running out of memory when running my queries. I was able to figure out how to restrict cassandra to run in less than 4gb. Is there such a setting for hadoop?”

- **Stack Overflow #45565896, MapReduce Error: Java heap space**
  
  “Besides those parameters in the configuration, I do not change anything else, so I use the default values. How can I solve the Error: Java Heap Space”
How to performance-tune configurations?

---

**Stack Overflow #37897438, Hbase Performance Tuning**

“I have the following parameters in Hbase: ... Can anyone suggest any configuration changes to generate more IO per second?”

---

**Stack Overflow #7243670, Hbase performance**

“My major configurations are: ... Am I doing something wrong with the configuration? This is my last shot at Hbase. Please help”

---

**Jira #HBase-13919, Rationalize Client Timeout**

“There are current many setting that influence how/when an HBase client times out. This is hard to configure, hard to understand, and badly documented.”
Performance Misconfigurations

• Common
  – 65% of configuration issue reports
  – 35% of configuration posts on Stack Overflow

• Severe
  – 20% of MySQL misconfig. -> severe slowdown
  – 1/3 of Hadoop misconfig. -> memory issue (OOM)

Can we help?

Can we automatically answer ...

Does a configuration affect performance?

How does a configuration affect performance?
Previous work ---- Auto-tuning

- Expensive training and profiling
- Not working if workload/environment changes at run time

How can we do better?
Our Key Insights

**Dynamic behavior**

- Does a configuration affect performance?
- How does a Performance-sensitive Configuration (PerfConf) affect performance?

**Static program logic**

- Does a Performance Operation (PerfOp) depend on the configuration?
- How does the PerfOp depend on the PerfConf?

```java
int sortmb = job.getInt("io.sort.mb");
int maxUsage = sortmb * 1024 * 1024;
buffer = new Byte[maxUsage];
```
How to use program analysis to infer configurations' performance impact?

• Understanding
• Tool design
• Evaluation
How to use program analysis to infer configurations' performance impact?

- Understanding
- Tool design
- Evaluation
How can a Conf affect a Perf-Op?

Data Dependency

If Related

Control Dependency

Loop Related
How can a Conf affect performance?

- Data Dependency
  - Control Dependency
    - If Related
      - Constant
      - Approaching
      - Bouncy
      - Unrelated
    - Loop Related
      - Loop Bound
      - Loop Stride
      - Sync. Loop
      - Infinite Loop

What type of variables are compared in IF predicate?
How does Conf affect loop?
How can a Conf affect performance?

Data Dependency

Control Dependency

Loop Bound
loop stride
Sync. Loop
Infinite Loop

Loop Related

If Related

Constant
Approaching
Bouncy
Unrelated
Data Dependency

• Configuration affects the **impact of every instance** of PerfOp through parameters

<table>
<thead>
<tr>
<th>Code Example</th>
<th>Formula</th>
<th>Performance Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>PerfOp(Conf)</td>
<td>Perfomance = Conf</td>
<td></td>
</tr>
</tbody>
</table>

```
int sortmb = job.getInt("io.sort.mb");
int maxUsage = sortmb * 1024 * 1024;
buffer = new Byte[maxUsage];
```
How can a Conf affect performance?

<table>
<thead>
<tr>
<th>Data Dependency</th>
<th>Control Dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td>If Related</td>
<td>Constant</td>
</tr>
<tr>
<td>Related</td>
<td>Approaching</td>
</tr>
<tr>
<td>Loop Related</td>
<td>Bouncy</td>
</tr>
<tr>
<td>Loop Bound</td>
<td>Unrelated</td>
</tr>
<tr>
<td>Loop Stride</td>
<td>Sync. Loop</td>
</tr>
<tr>
<td>Infinite Loop</td>
<td></td>
</tr>
</tbody>
</table>
If Related Patterns

- Conf affects whether the PerfOp is executed

<table>
<thead>
<tr>
<th>Code Example</th>
<th>Formula</th>
<th>Performance Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>If ((V \leq C)) {</td>
<td>Performance (= \begin{cases} a, &amp; V \leq C \ b, &amp; V &gt; C \end{cases})</td>
<td><img src="image" alt="Performance Graph" /></td>
</tr>
</tbody>
</table>
How can a Conf affect performance?

- **Data Dependency**
  - Related
  - Loop Related
  - Loop Bound
  - Loop Stride
  - Sync. Loop
  - Infinite Loop
- **Control Dependency**
  - If Related
  - Approaching
  - Bouncy
  - Unrelated
  - Constant
Compared with Constant

• The if-else decision does not change over time

```java
if (maxFsObjects != 0) {
    lock();
}
```
How can a Conf affect performance?

- Data Dependency
  - Constant
  - Approaching
  - Bouncy
  - Unrelated
- Control Dependency
  - If Related
- Loop Related
  - Loop Bound
  - Loop Stride
  - Sync. Loop
  - Infinite Loop
Compared with Bouncy Variable

- The if-else decision keeps changing over time

```java
  currentSize += put.heapSize();
  writeBuffer.add(put);
  if (currentSize > bufferSize) {
    writeBuffer.clear();
    currentSize = 0;
  }
```
How can a Conf affect performance?

- Data Dependency
  - If Related
  - Constant
  - Approaching
  - Bouncy
  - Unrelated
- Control Dependency
  - Loop Bound
  - Loop Stride
  - Sync. Loop
  - Infinite Loop
How can a Conf affect performance?

Data Dependency

Control Dependency

Loop Related

If Related

Constant

Approaching

Bouncy

Unrelated

Loop Bound

Sync. Loop

Infinite Loop

Loop Stride
Affect Loop Stride

- Conf used as a loop stride in the loop-exit condition

<table>
<thead>
<tr>
<th>Code Example</th>
<th>Formula</th>
<th>Performance Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>for (; i &lt; N; i+=Conf) {</td>
<td>Performance = aN/Conf</td>
<td></td>
</tr>
<tr>
<td>PerfOp();</td>
<td></td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

while (bytesRemaining > 0) {
    splits.add(makeSplit());
    bytesRemaining -= splitSize;
}
How can a Conf affect performance?

Data Dependency

- Constant
- Approaching
- Bouncy
- Unrelated

Control Dependency

- If Related
- Loop Bound
- LoopStride
- Infinite Loop

More details in the paper
• Understanding
• Tool design
• Evaluation
LearnConf Overview

- Binary
  - PerfConf - PerfOp Identification
    - PerfConf-PerfOp pair
    - Pattern Analysis
      - Pattern-specific Analysis
      - Performance Properties
        - List of PerfConf
          - Pattern: Data Dependency
            - Pat. Spe. Properties:
              - User request: MapTask
              - Linear: Yes
              - Slope: 1024 * 1024
              - Range effect: No
              - Related PerfConf: No
              - Monotonic: Yes
Identify Configuration Variable

- Identify configuration-loading API
  - Add return var. to configuration variable set
- Track data-dependence chain
  - Tag more variables as configuration variables

```java
int sortmb = job.getInt("io.sort.mb");
int maxUsage = sortmb * 1024 * 1024;
buffer = new Byte[maxUsage];
```
Identify PerfOps

• Latency related
  • Sleep(), lock(), IO, etc.

• Memory related
  • new byte[], List.add(), etc.

```java
int sortmb = job.getInt("io.sort.mb");
int maxUsage = sortmb * 1024 * 1024;
buffer = new Byte[maxUsage];
```

Memory Intensive Operation
Identify PerfConf

If a PerfOp depends on the Configuration Variable, ...

```
int sortmb = job.getInt("io.sort.mb");
int maxUsage = sortmb * 1024 * 1024;
buffer = new Byte[maxUsage];
```

List of PerfConf
- io.sort.mb
- ...

Categorize PerfConf-PerfOp dependency

- **Data Dependency Pattern**
  - Conf used in the parameter of the PerfOp

- **If Pattern**
  - Conf used in an if-predicate

- **Loop Pattern**
  - Conf used in a loop-exit condition

```java
int sortmb = job.getInt("io.sort.mb");
int maxUsage = sortmb * 1024 * 1024;
buffer = new Byte[maxUsage];
```
Pattern-Specific Analysis

List of PerfConf

- Pattern: Data Dependency

Pat. Spe. Properties:
- User request: MapTask
- Linear: Yes
- Slope: 1024 * 1024
- Range effect: No
- Related PerfConf: No
- Monotonic: Yes

int sortmb = job.getInt("io.sort.mb");
int maxUsage = sortmb * 1024 * 1024;
buffer = new Byte[maxUsage];
• Understanding
• Tool design
• Evaluation
Methodology

• Benchmarks
  – Four widely used distributed systems
  – Each contains around 100~150 configurations
Identify Correct PerfConf

- Correctly identify 60 out of 71 true PerfConfs
- 9 false positives
- 4 true PerfConfs \textbf{not} in previous work that can lead to OOM or timeout failures!

<table>
<thead>
<tr>
<th></th>
<th>Identified</th>
<th>False Positive</th>
<th>False Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>MapReduce</td>
<td>16</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>HBase</td>
<td>19</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>HDFS</td>
<td>13</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Cassandra</td>
<td>21</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>69</td>
<td>9</td>
<td>11</td>
</tr>
</tbody>
</table>
Identify Correct Pattern
More Result

• Input Analysis
• Slope Analysis
• Configuration Setting Range Analysis
• Configuration Relation Analysis
• Monotonicity Analysis
• Applying LearnConf for Performance Tuning

More results in the paper
Conclusion

Data Dependency

Control Dependency

Loop Related

If Related

Constant

Approaching

Bouncy

Unrelated

Loop Bound

Loop Stride

Sync. Loop

Infinite Loop

Thanks

Chi Li
lichi@uchicago.edu