

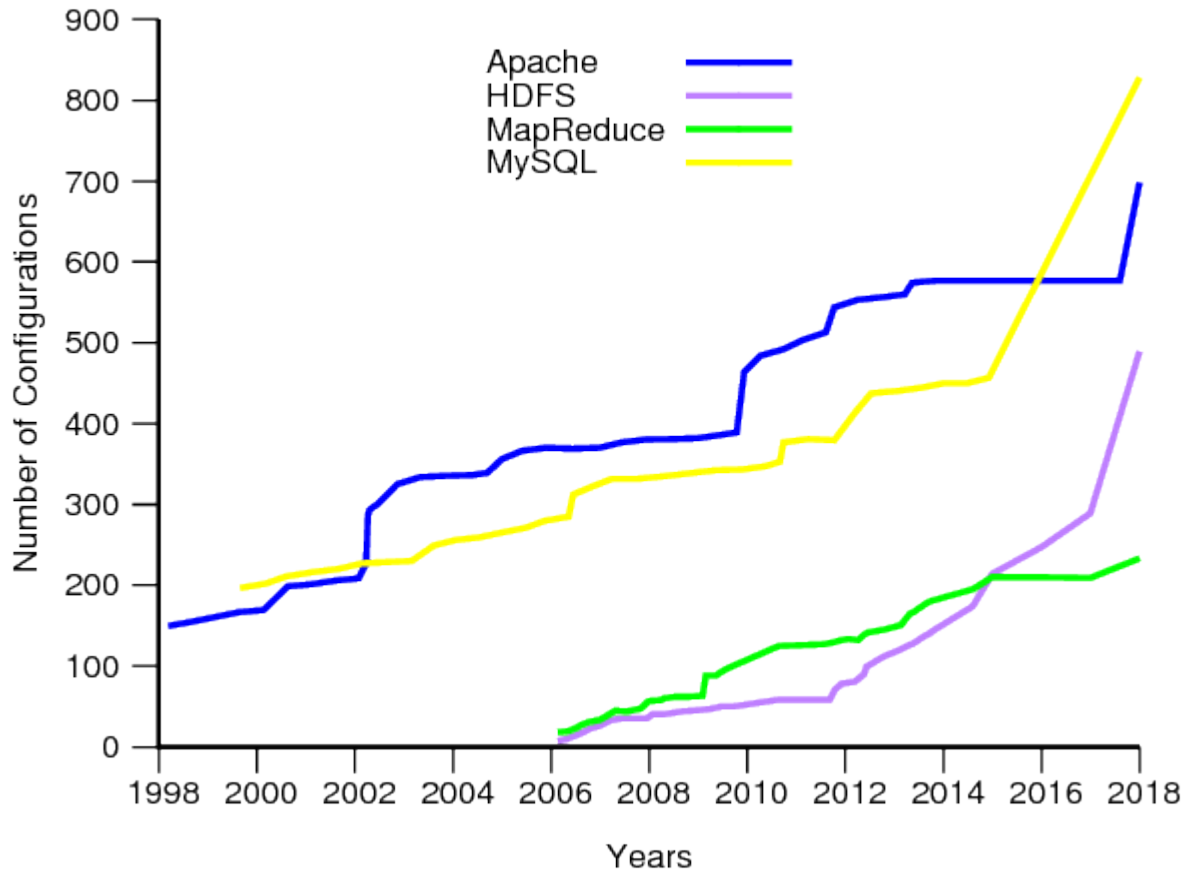
# Statically Inferring Performance Properties of Software Configurations

Chi Li, Shu Wang, Henry Hoffmann, Shan Lu



THE UNIVERSITY OF  
CHICAGO

# Configurations Explosion



Tianyin Xu, Long Jin, Xuepeng Fan, Yuanyuan Zhou, Shankar Pasupathy, and Rukma Talwadker . Hey, You Have Given Me Too Many Knobs! Understanding and Dealing with Over-Designed Configuration in System Software. In FSE, 2016

# Which configuration affects performance?

 **stackoverflow #36170959, Cassandra Performance Tuning**

*"Please let me know **what more settings I can tweak** to get maximum performance out of my cluster."*


 **stackoverflow #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?**"*

 **stackoverflow #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?

 **stackoverflow #37897438, Hbase Performance Tuning**

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

 **stackoverflow #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 currently many settings 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 ...

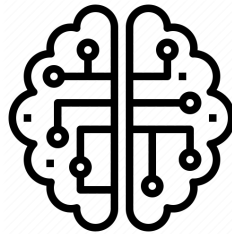
Does a configuration affect performance?

How does a configuration affect performance?

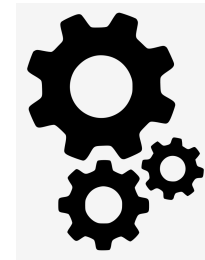
# Previous work ---- Auto-tuning



Profiling Data



Machine Learning and  
Control Theory



Optimal Settings

- 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?

reflects



Static program logic

Does a Performance Operation (PerfOp) depend on the configuration?

How does the PerfOp depend on the PerfConf?

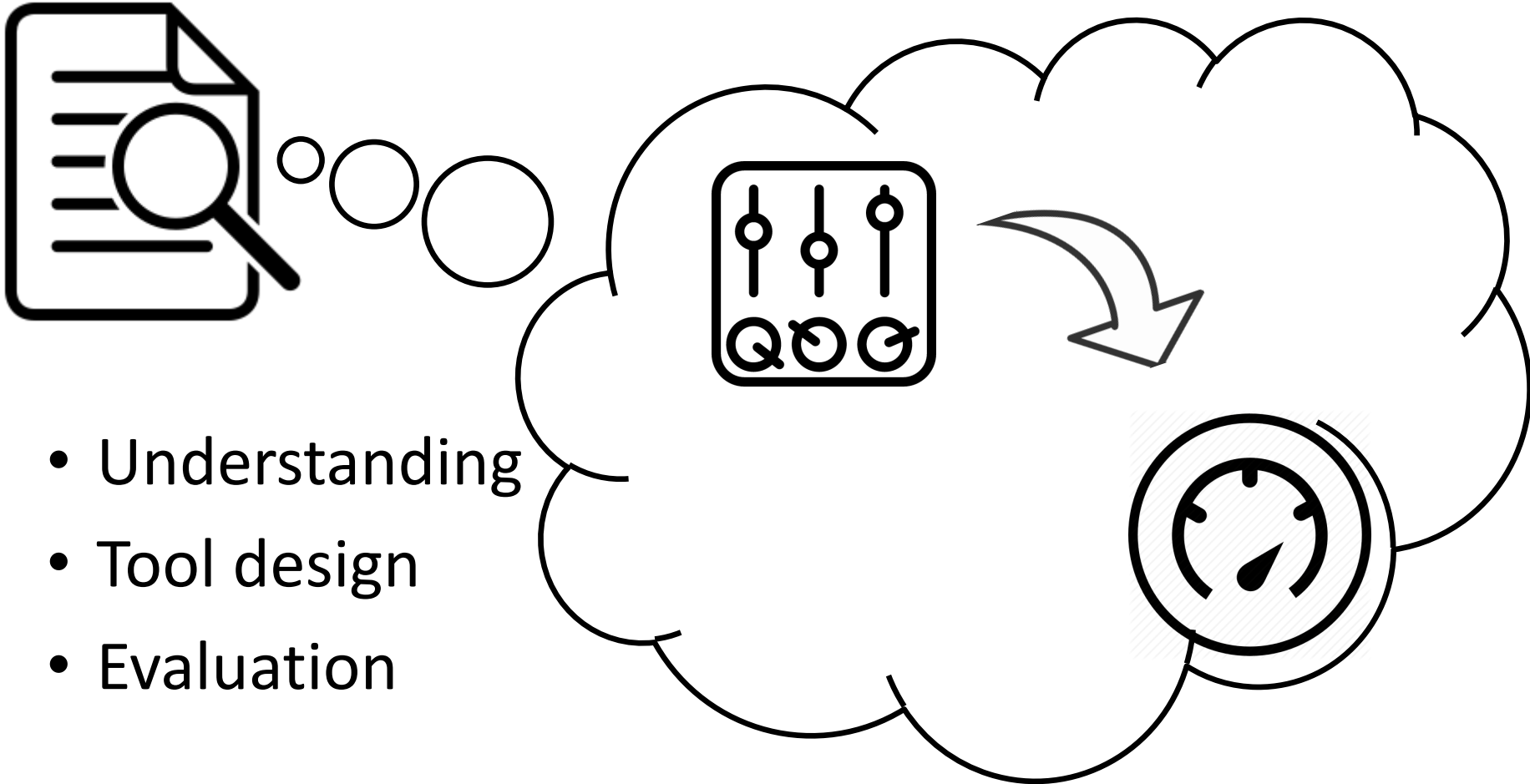
**PerfConf**

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

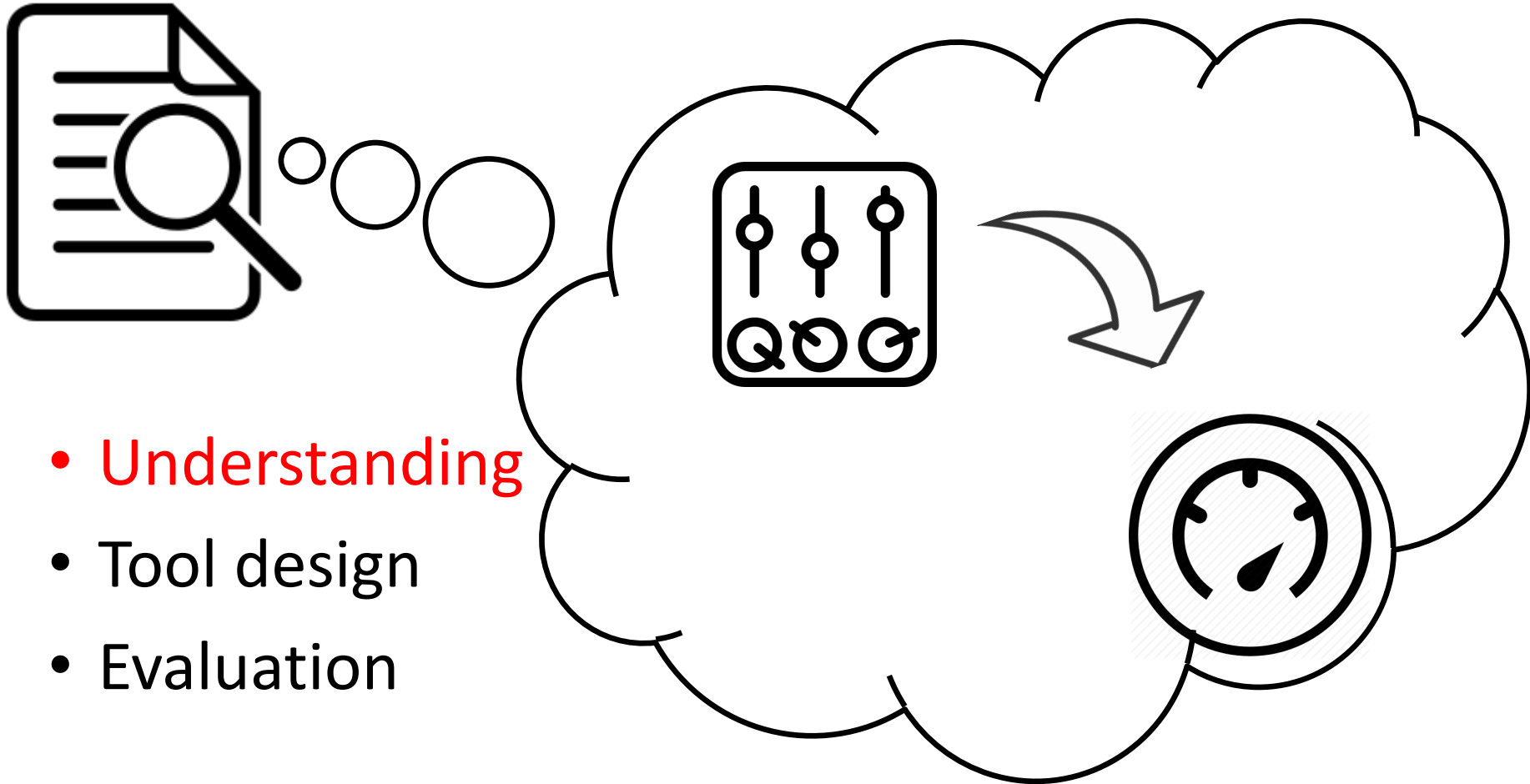
**PerfOp**



# *How to use program analysis to infer configurations' performance impact?*

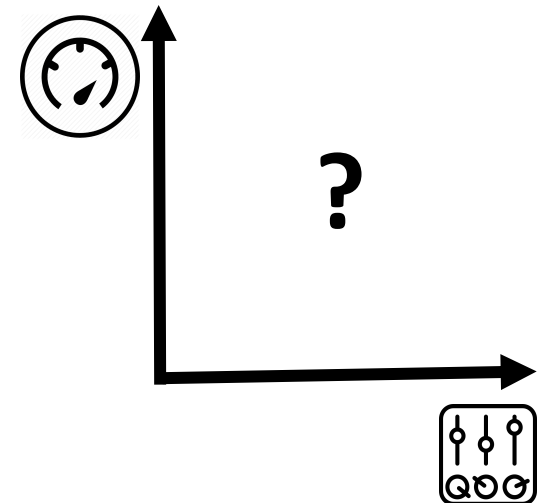
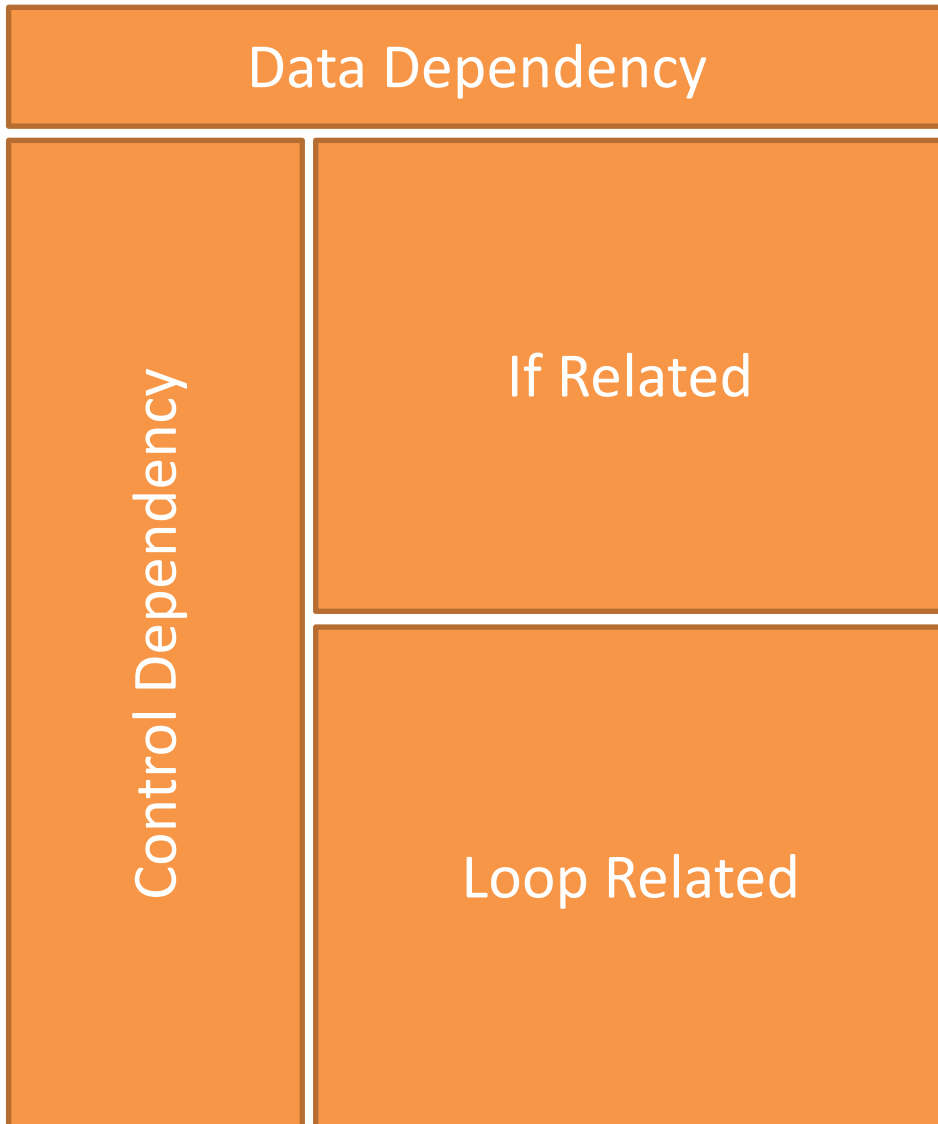


# *How to use program analysis to infer configurations' performance impact?*

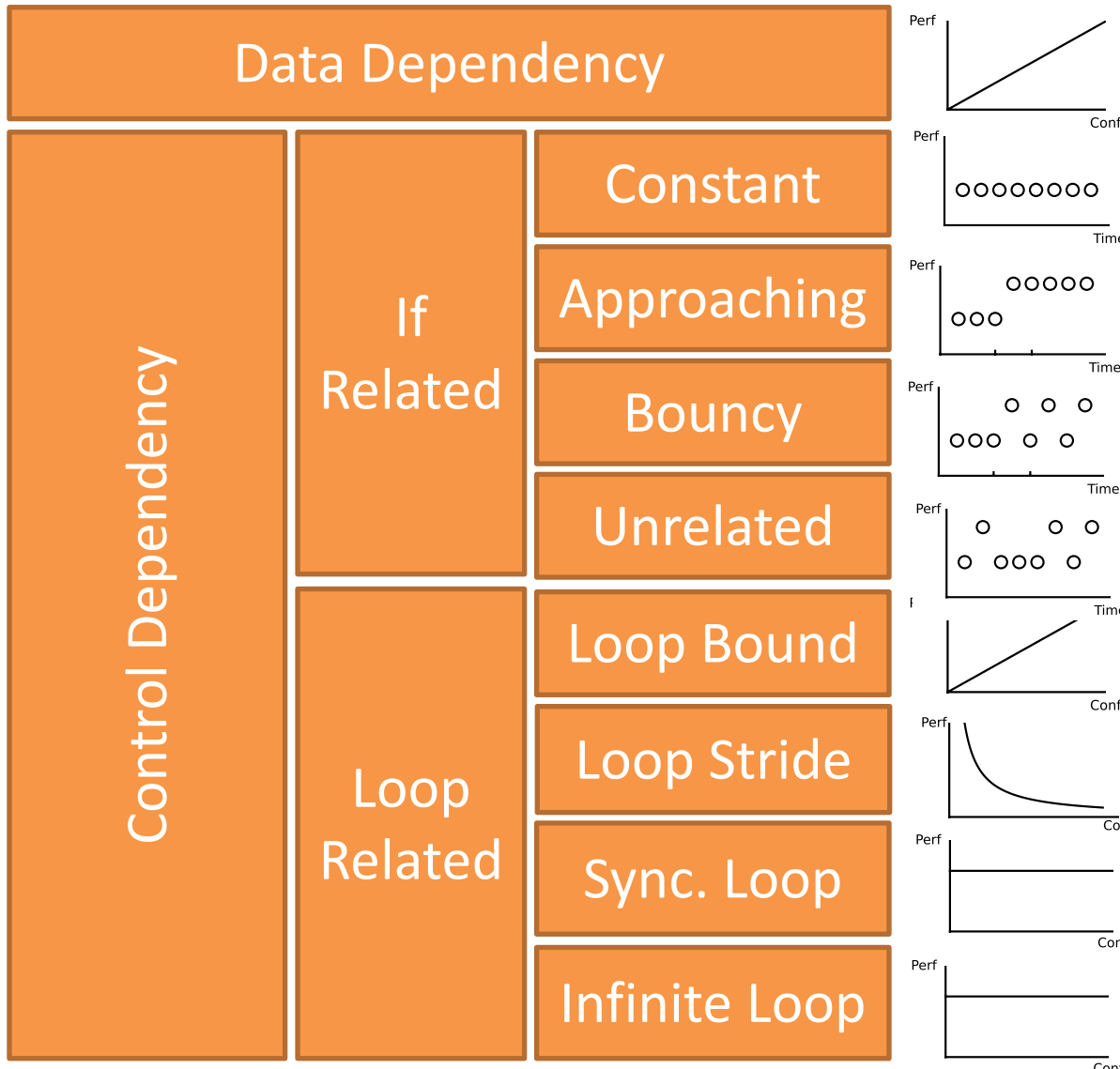


- **Understanding**
- Tool design
- Evaluation

# How can a Conf affect a Perf-Op?



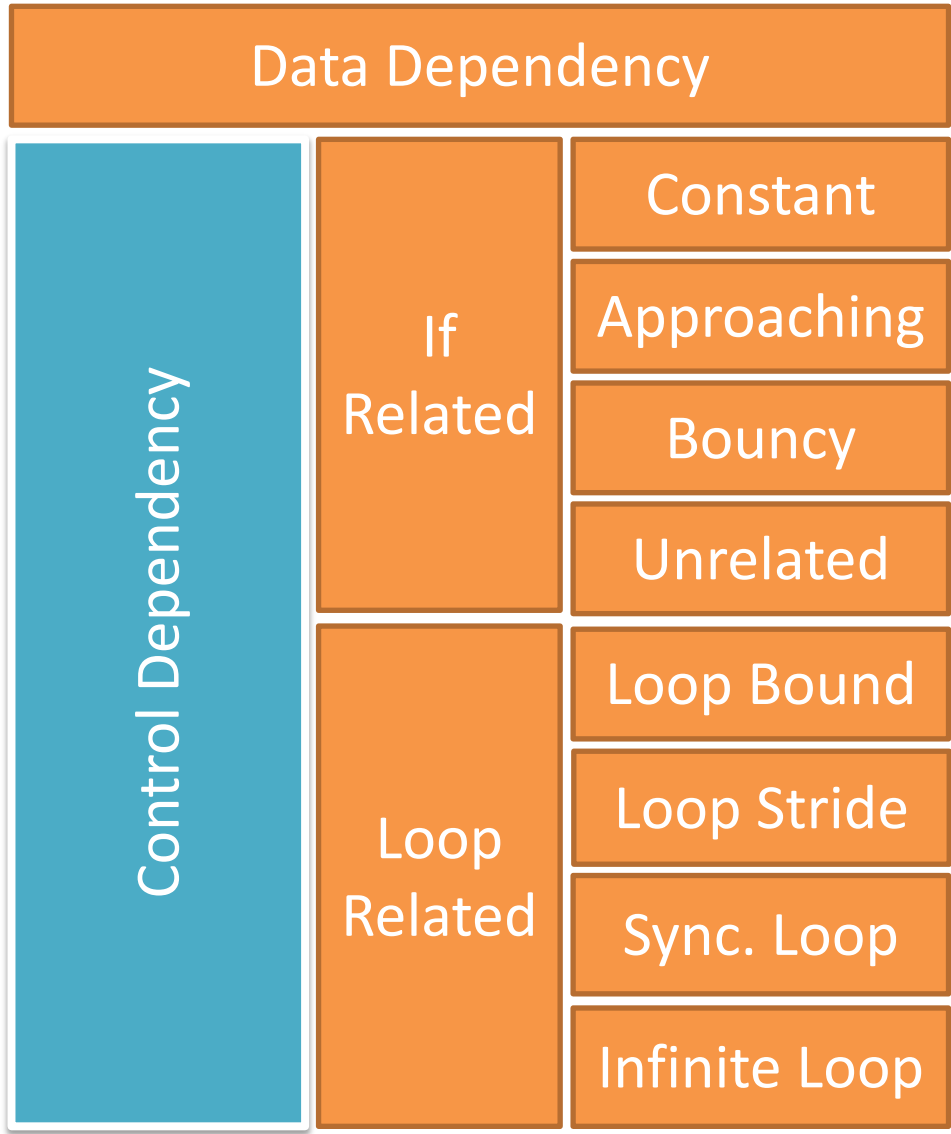
# How can a Conf affect performance?



What type of variables are compared in IF predicate?

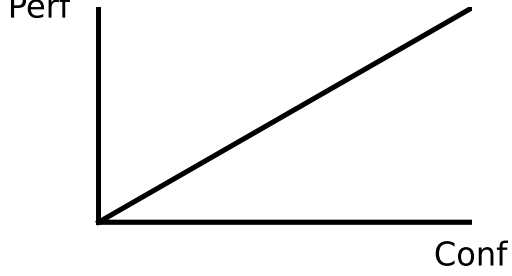
How does Conf affect loop?

# How can a Conf affect performance?



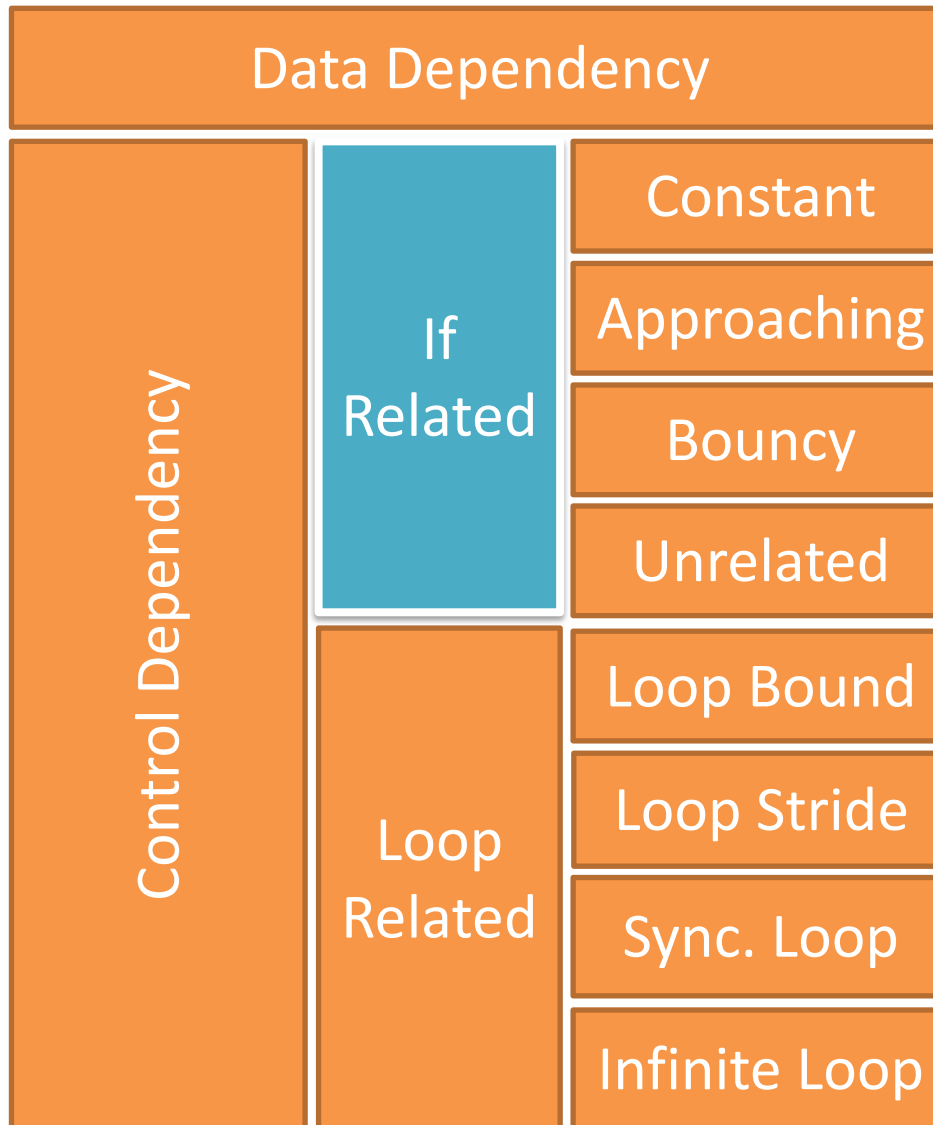
# Data Dependency

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

Code Example	Formula	Performance Graph
PerfOp(Conf)	$Performance = Conf$	

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

# How can a Conf affect performance?



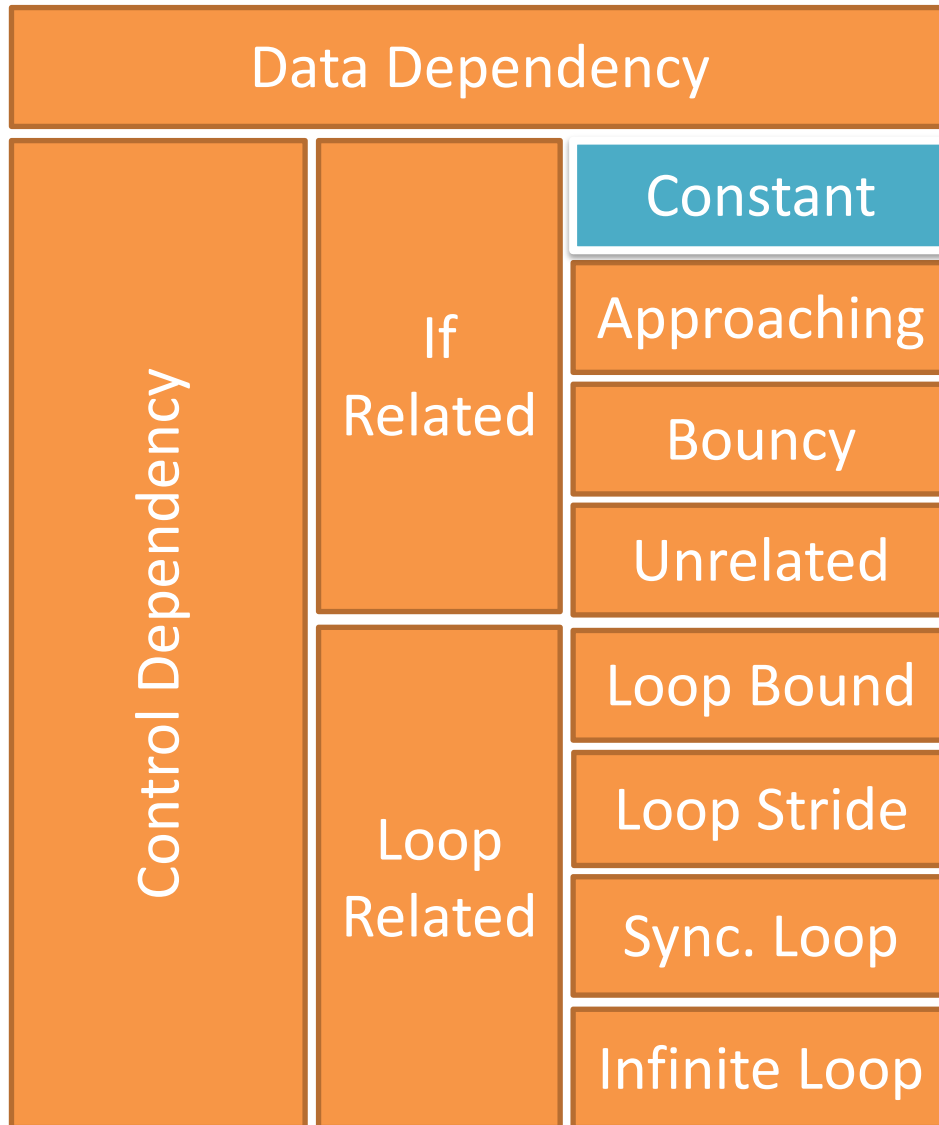
# If Related Patterns

- Conf affects whether the PerfOp is executed

Code Example	Formula	Performance Graph
<pre>If (V &lt;= C) {     PerfOpA } else {     PerfOpB }</pre>	$\text{Performance} = \begin{cases} a, & V \leq C \\ b, & V > C \end{cases}$	

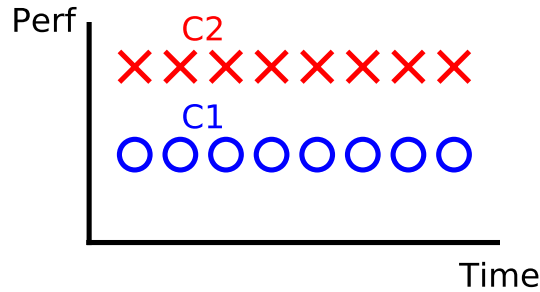


# How can a Conf affect performance?



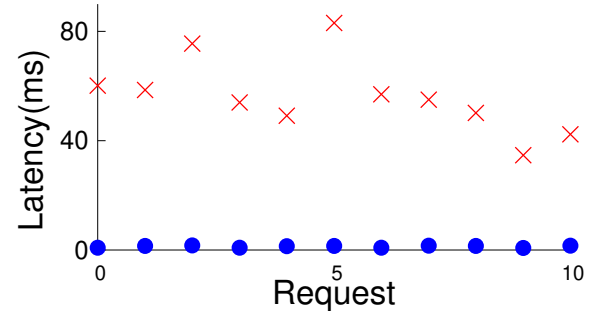
# Compared with Constant

- The if-else decision does not change over time

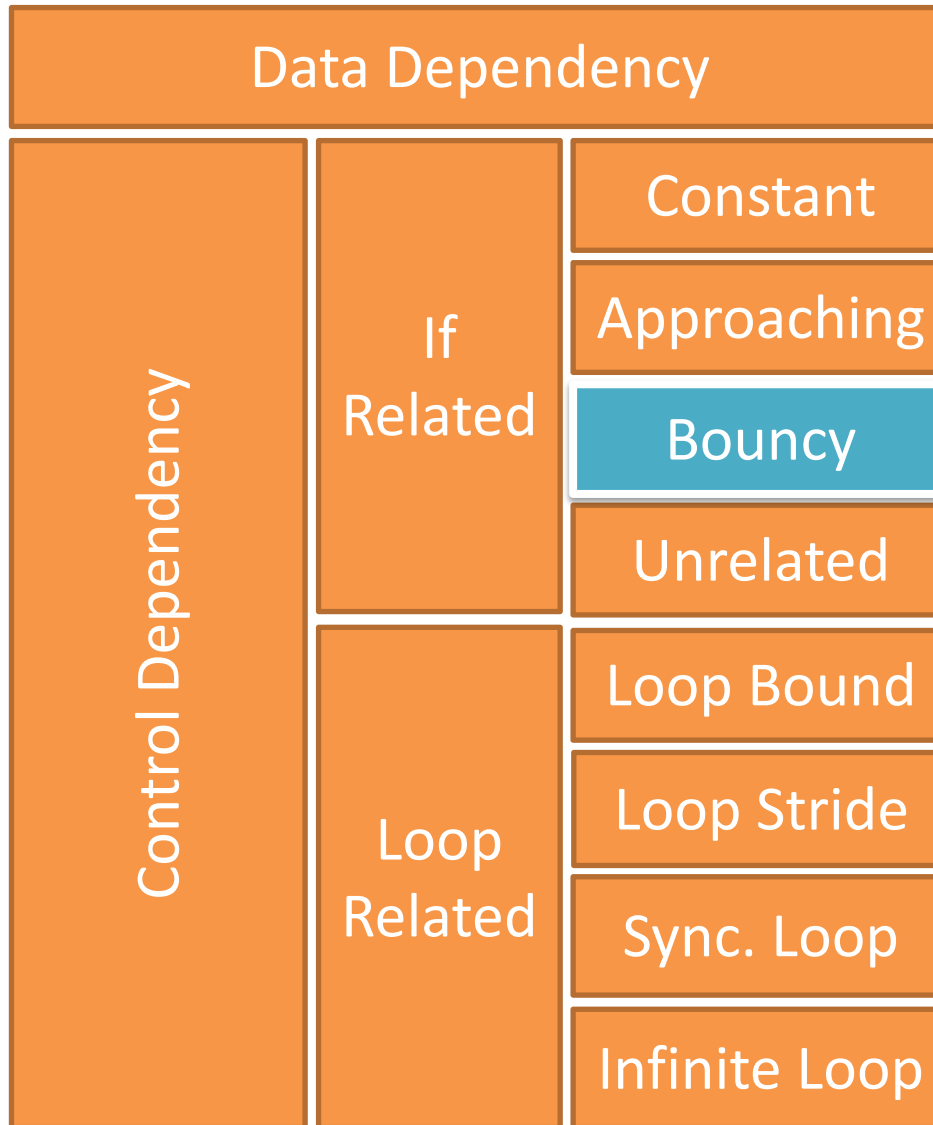


```

if (maxFsObjects != 0) {
    lock();
}
    
```

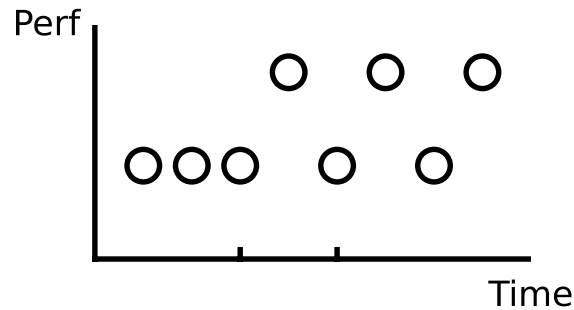


# How can a Conf affect performance?



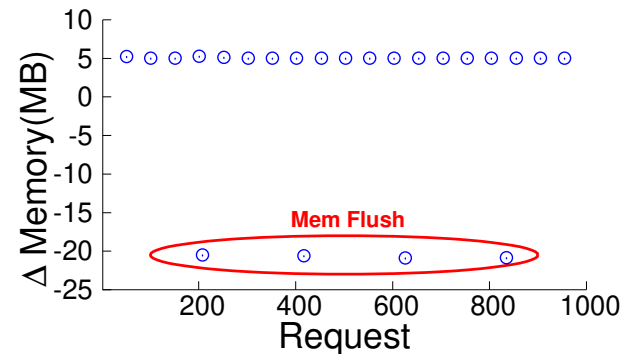
# Compared with Bouncy Variable

- The if-else decision keeps changing over time

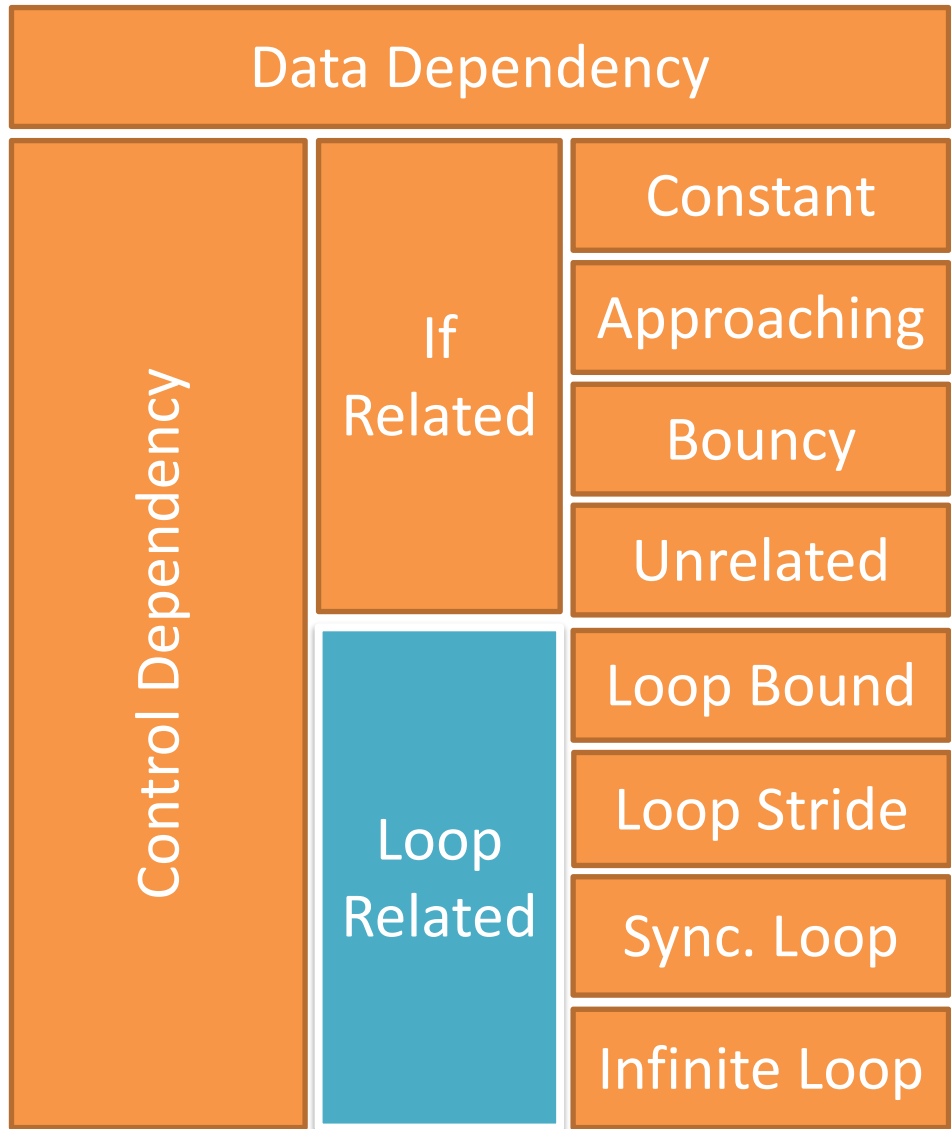


```

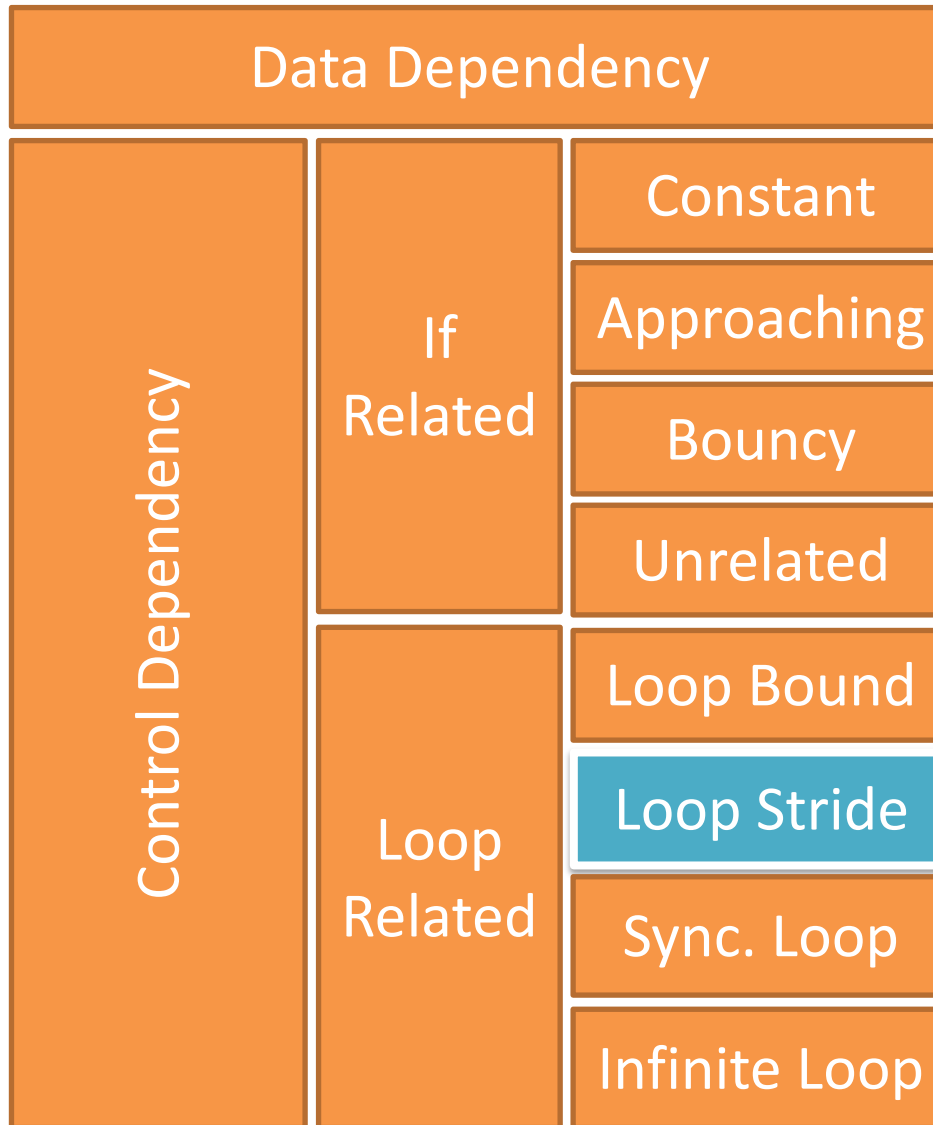
currentSize += put.heapSize();
writeBuffer.add(put);
If (currentSize > bufferSize) {
writeBuffer clear();
currentSize = 0;
}
    
```



# How can a Conf affect performance?



# How can a Conf affect performance?

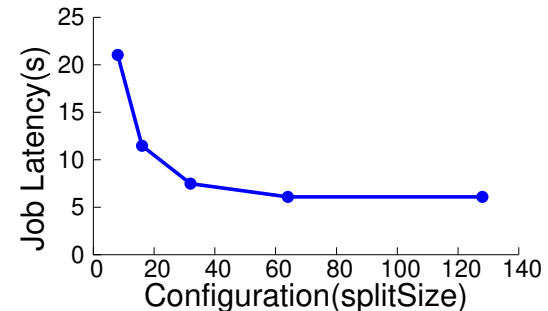


# Affect Loop Stride

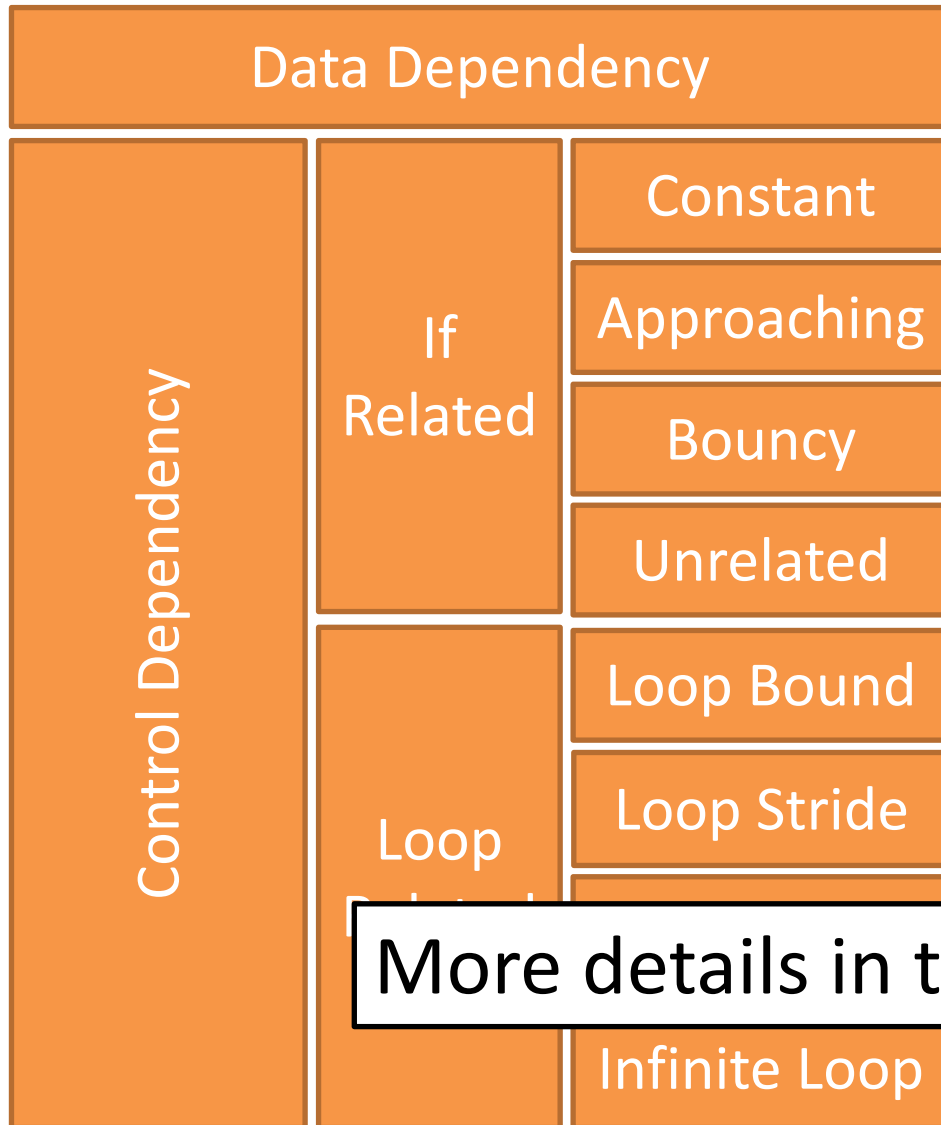
- Conf used as a loop stride in the loop-exit condition

Code Example	Formula	Performance Graph
<pre>for (; i &lt; N; i+=Conf) {     PerfOp(); }</pre>	$Performance = aN/Conf$	

```
while (bytesRemaining > 0) {
    splits.add(makeSplit());
    bytesRemaining -= splitSize;
}
```

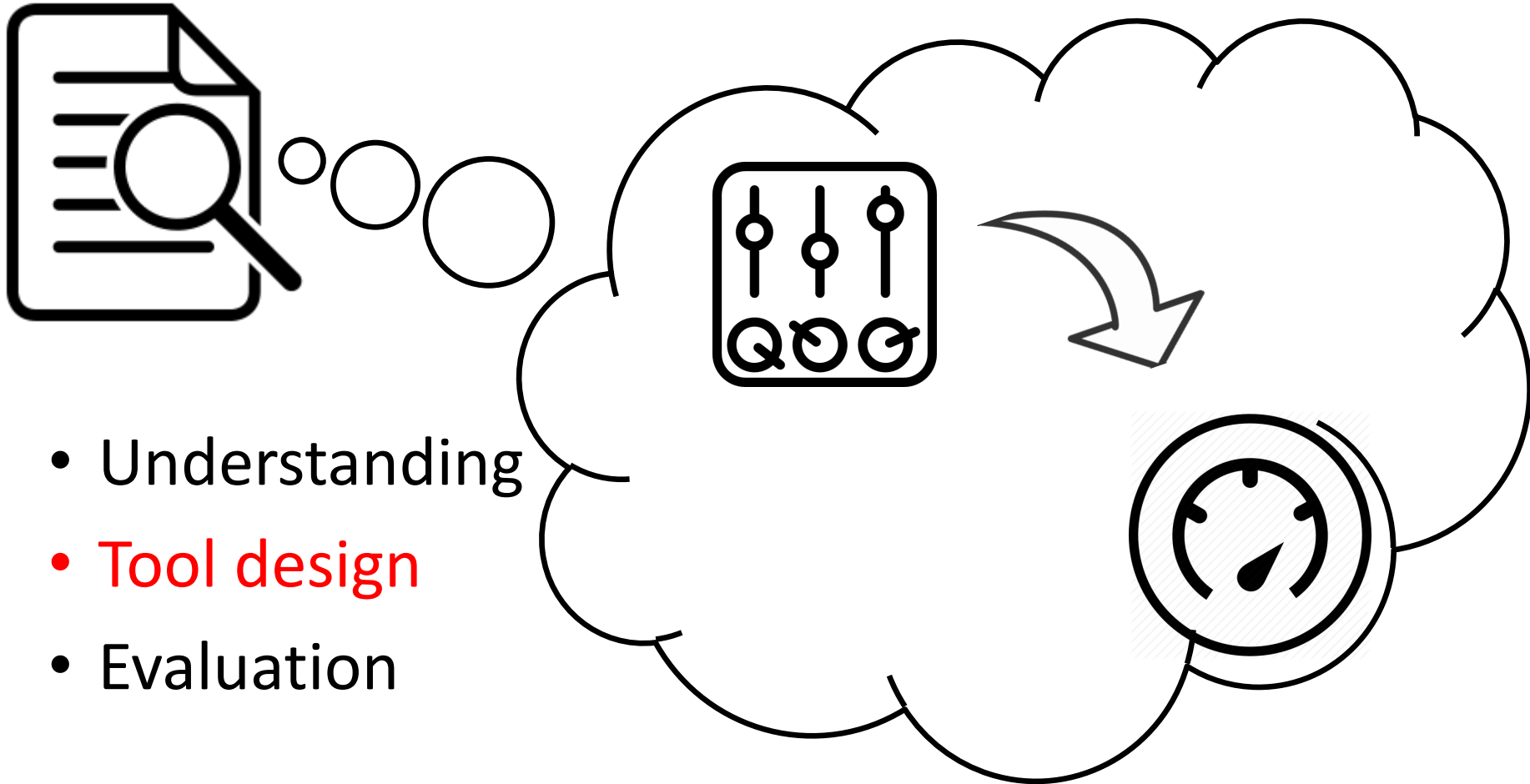


# How can a Conf affect performance?



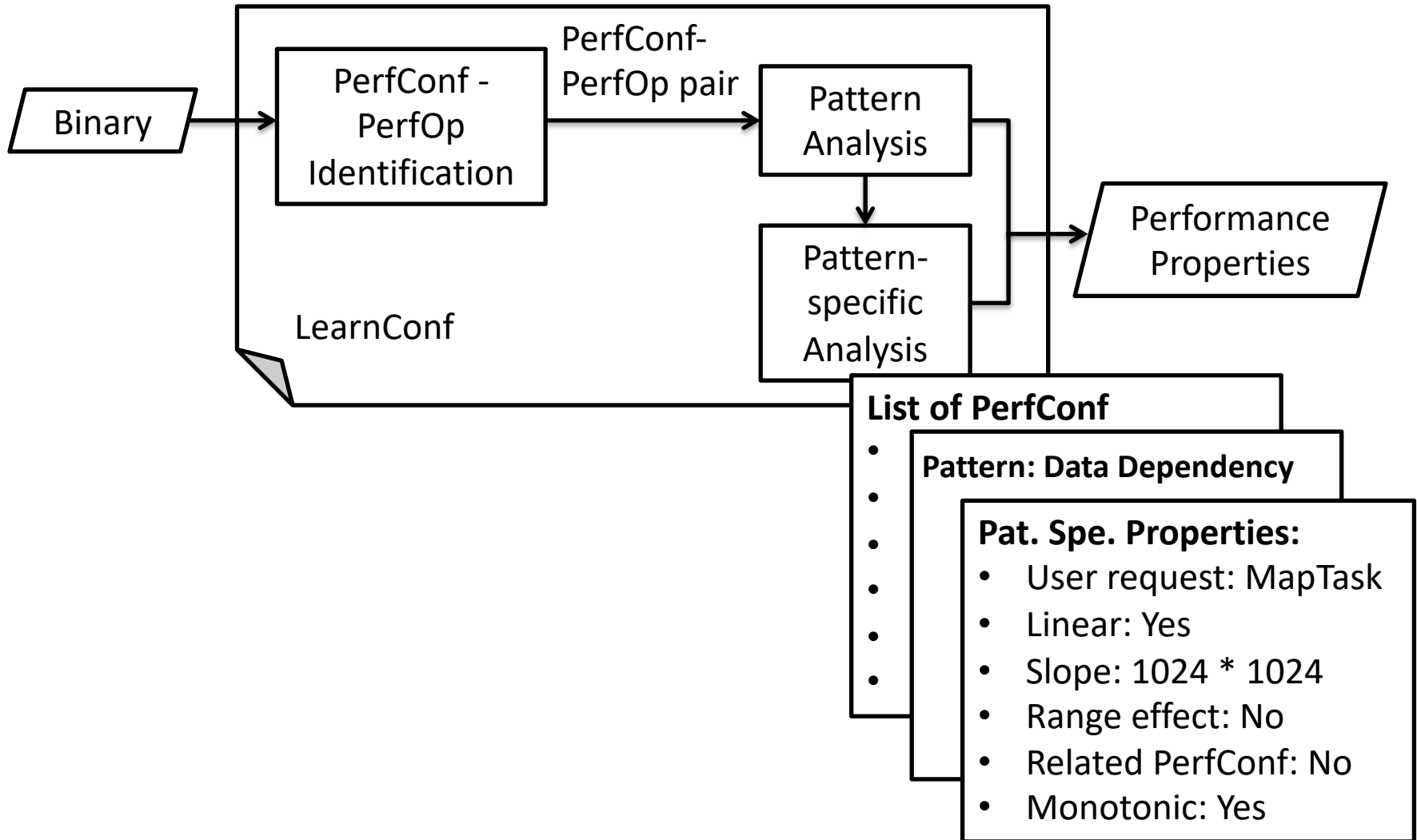
More details in the paper





- Understanding
- **Tool design**
- Evaluation

# LearnConf Overview



# Identify Configuration Variable

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

Configuration variable

configuration-loading API

```
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.

```
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

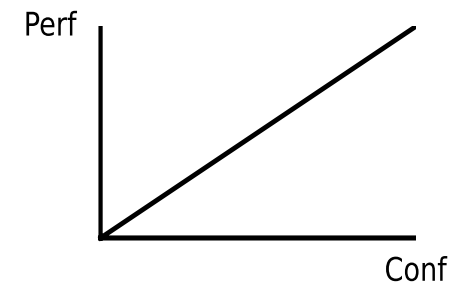
- io.sort.mb
- ...

# Categorize PerfConf-PerfOp dependency

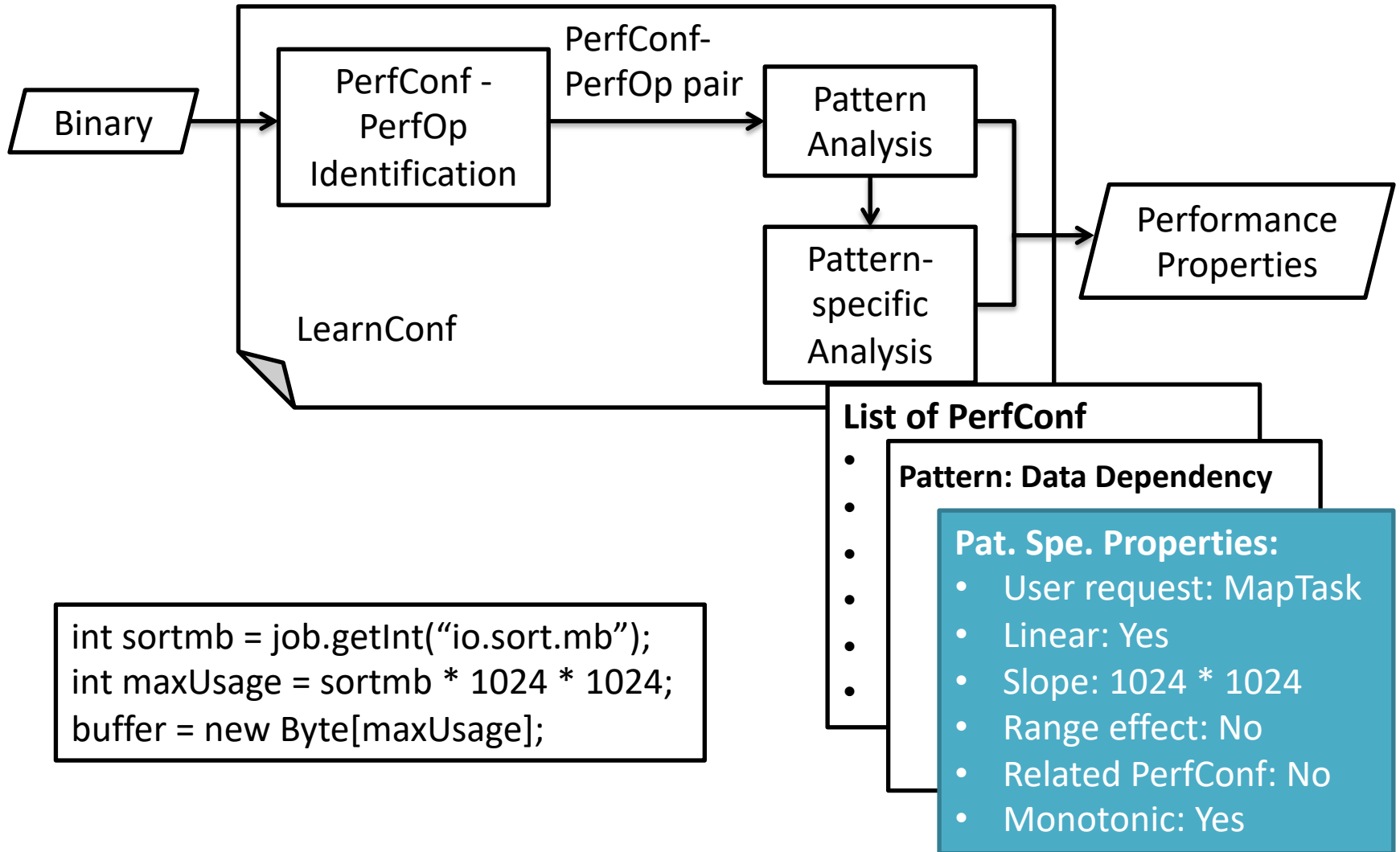
- Data Dependency Pattern
  - Conf used in the parameter of the PerfOp
- If Pattern
  - Conf used in an if-predicate
- Loop Pattern
  - Conf used in a loop-exit condition

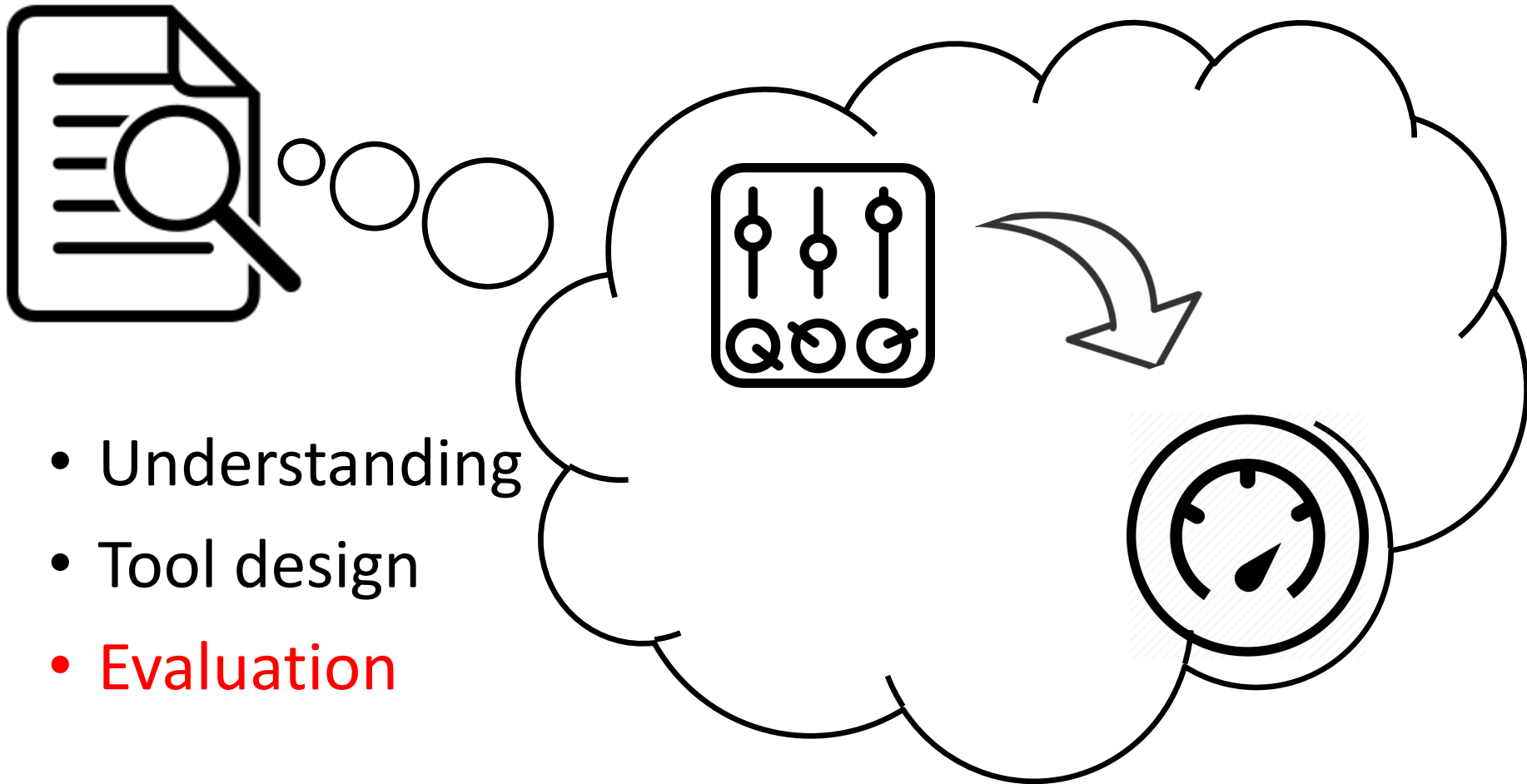
```
int sortmb = job.getInt("io.sort.mb");  
int maxUsage = sortmb * 1024 * 1024;  
buffer = new Byte[maxUsage];
```

**Pattern: Data Dependency**



# Pattern-Specific Analysis





- Understanding
- Tool design
- **Evaluation**



# Methodology

- Benchmarks
  - Four widely used distributed systems
  - Each contains around 100~150 configurations



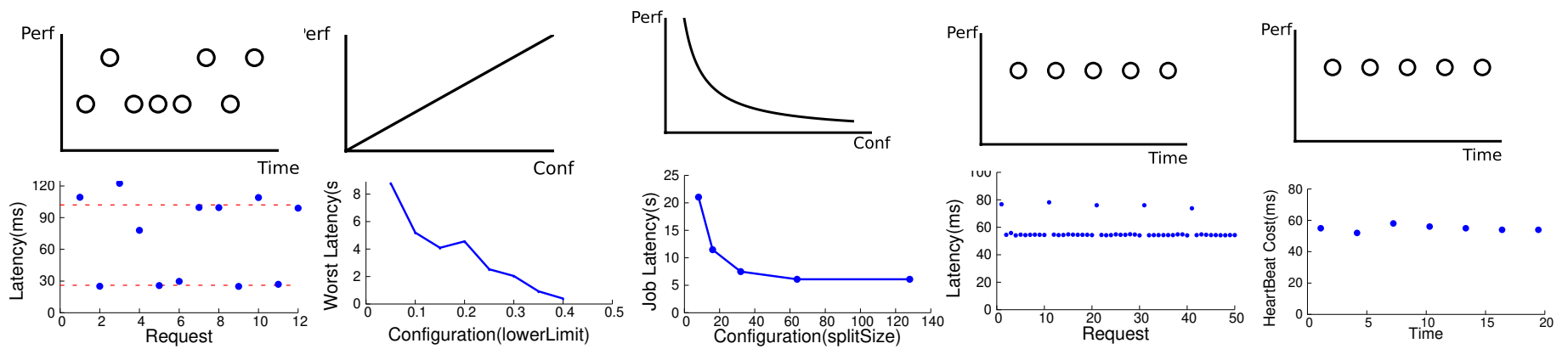
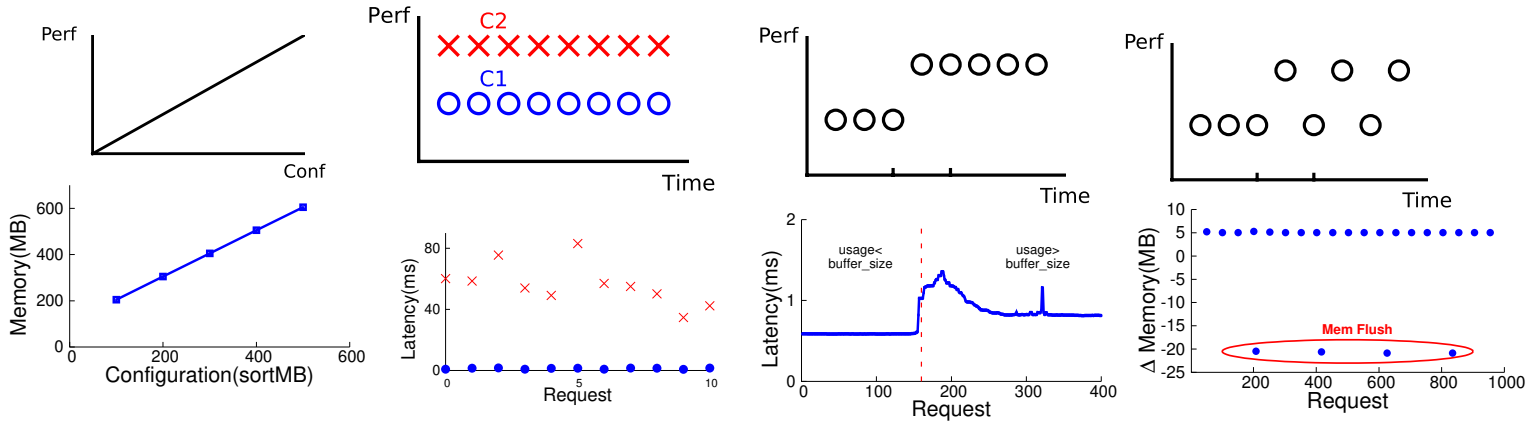
# Identify Correct PerfConf

UNION of tutorials and papers

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

	Identified	False Positive	False Negative
MapReduce	16	1	7
HBase	19	1	2
HDFS	13	5	1
Cassandra	21	2	1
Total	69	9	11

# Identify Correct Pattern

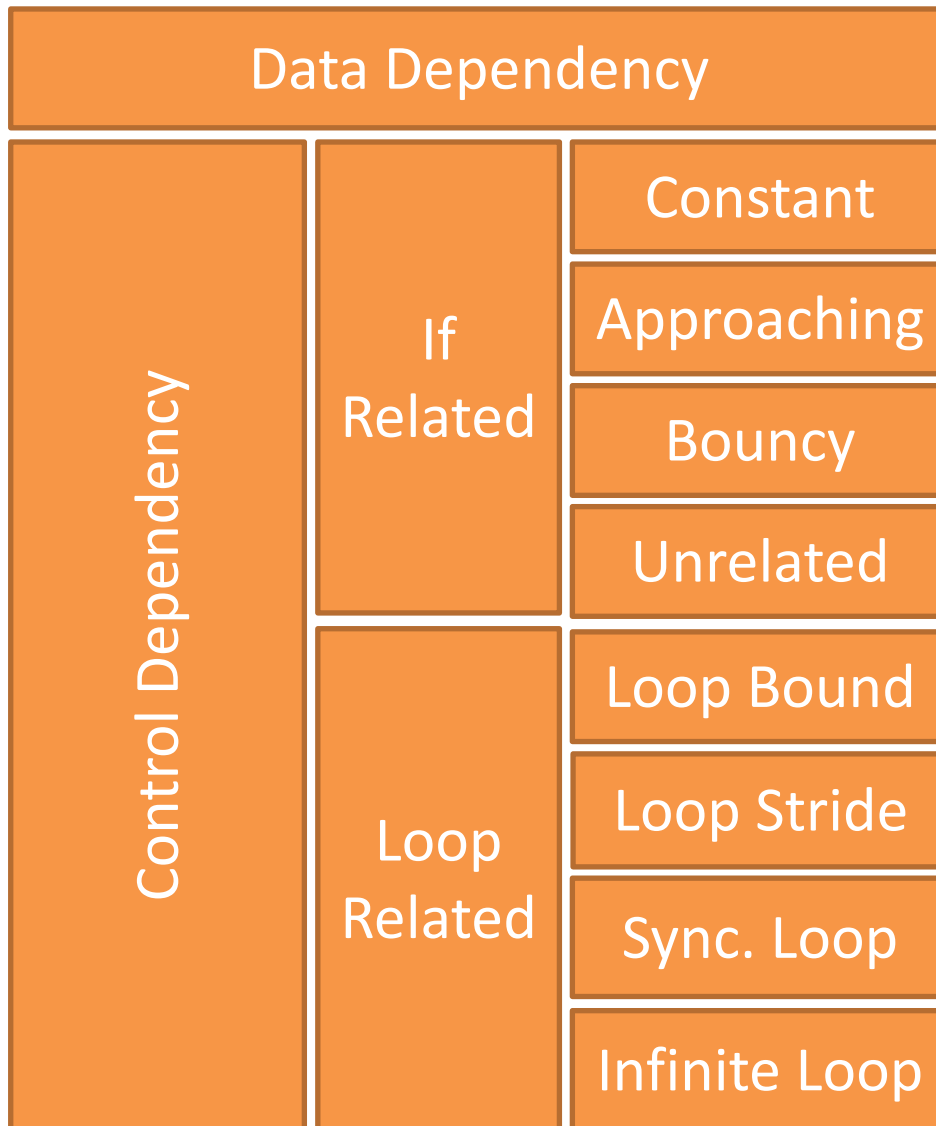


# More Result

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

More results in the paper

# Conclusion



**Thanks**

Chi Li  
 lichil@uchicago.edu