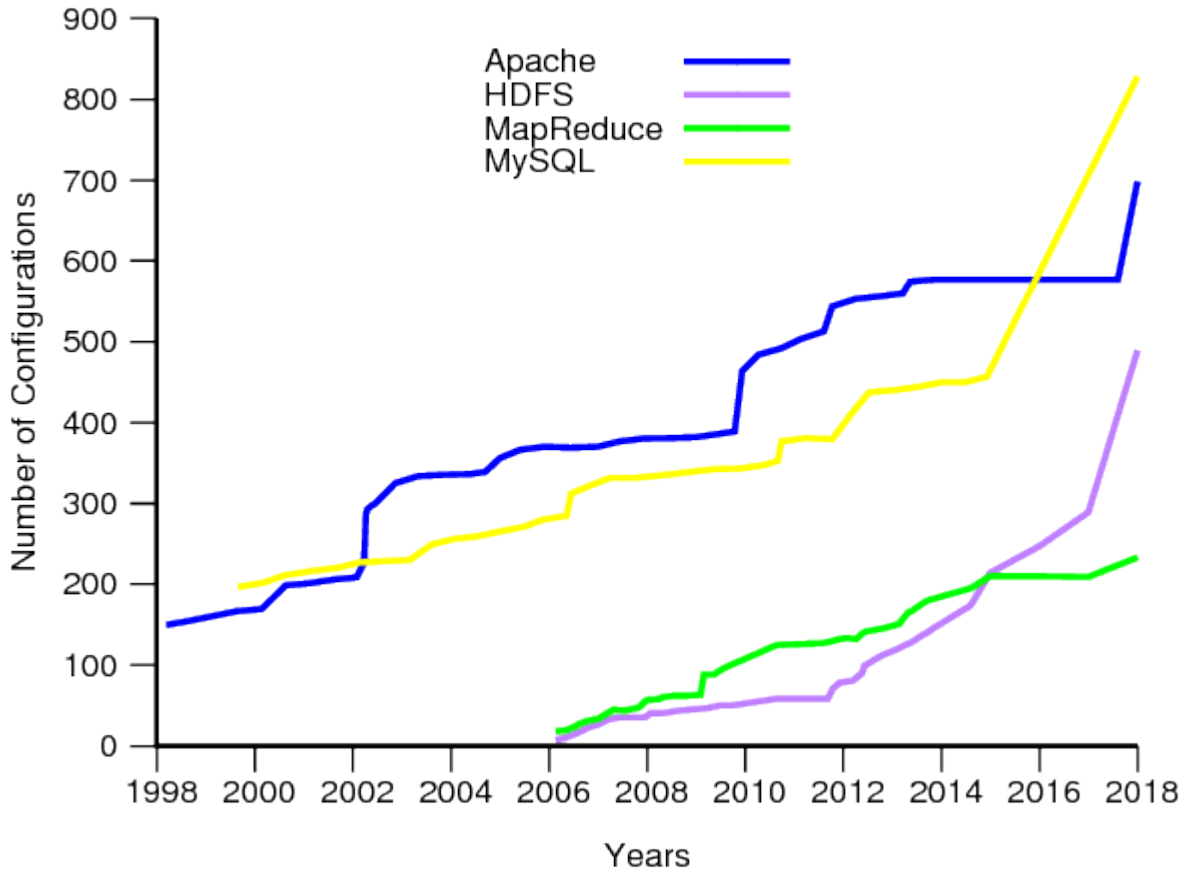
Statically Inferring Performance Properties of Software Configurations

Chi Li, Shu Wang, Henry Hoffmann, Shan Lu

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 petameters 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 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)

Shu Wang, Chi Li, Henry Hoffmann, Shan Lu, William Sentosa, and Achmad Imam Kistijantoro. Understanding and auto-adjusting performance-sensitive configurations. In ACM SIGPLAN Notices, volume 53, pages 154–168. ACM, 2018.
Zuoning Yin, Xiao Ma, Jing Zheng, Yuanyuan Zhou, Lakshmi N Bairavasundaram, and Shankar Pasupathy. An empirical study on configuration errors in commercial and open source systems. In SOSP, 2011

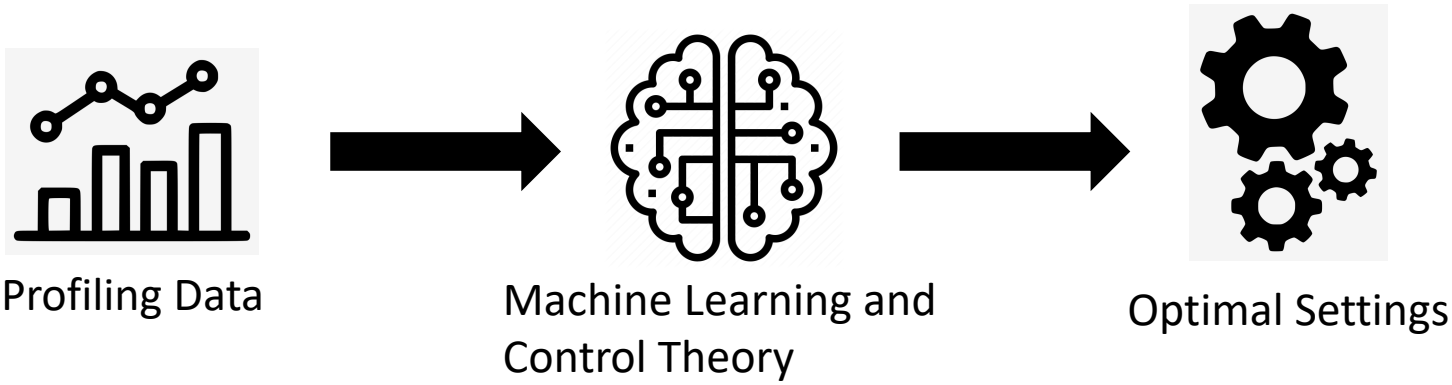
Can we help?

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

此外，如果user想改一个模型之外的配置项，就没办法了

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?

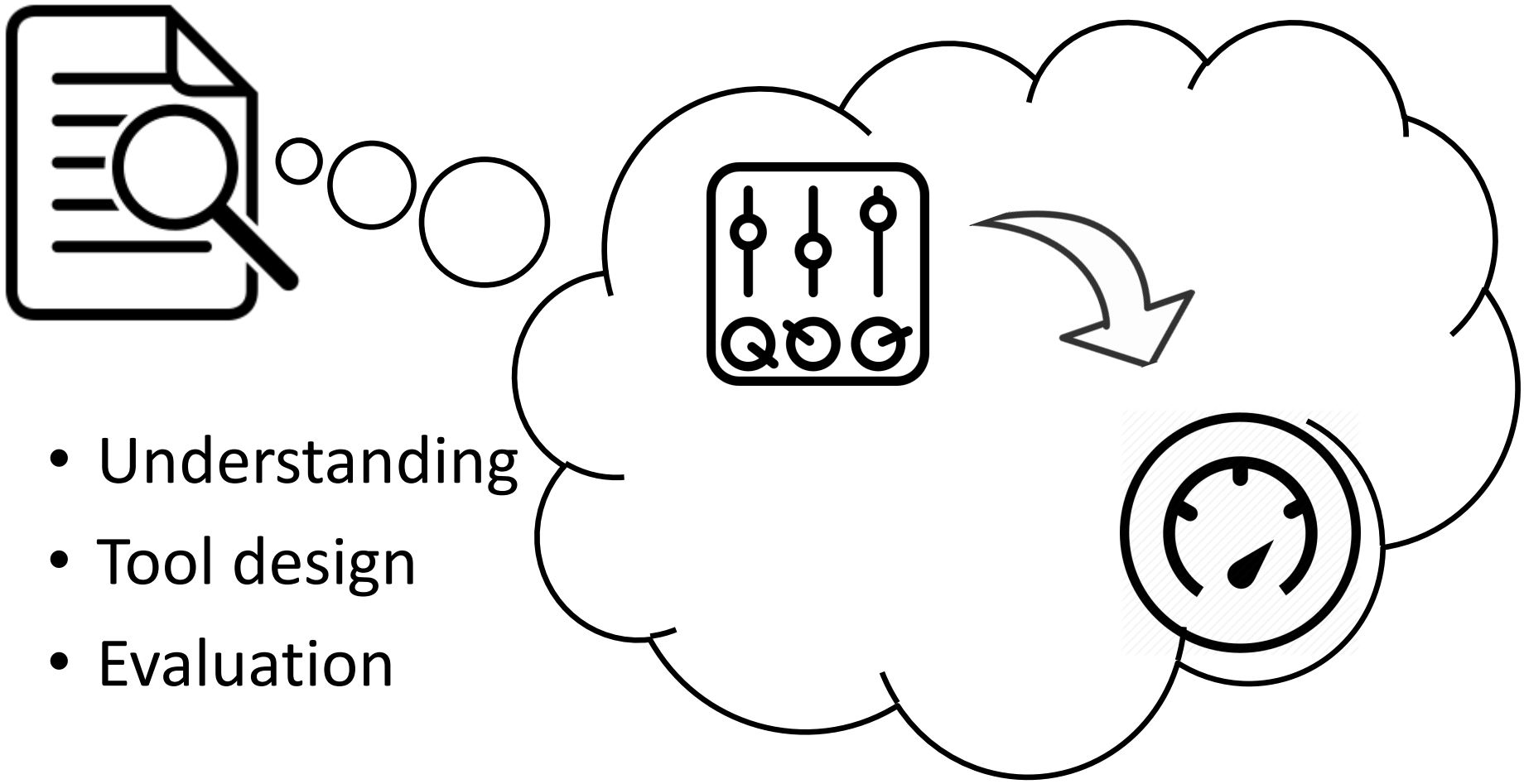
How does the PerfOp depend on the PerfConf?

PerfConf

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

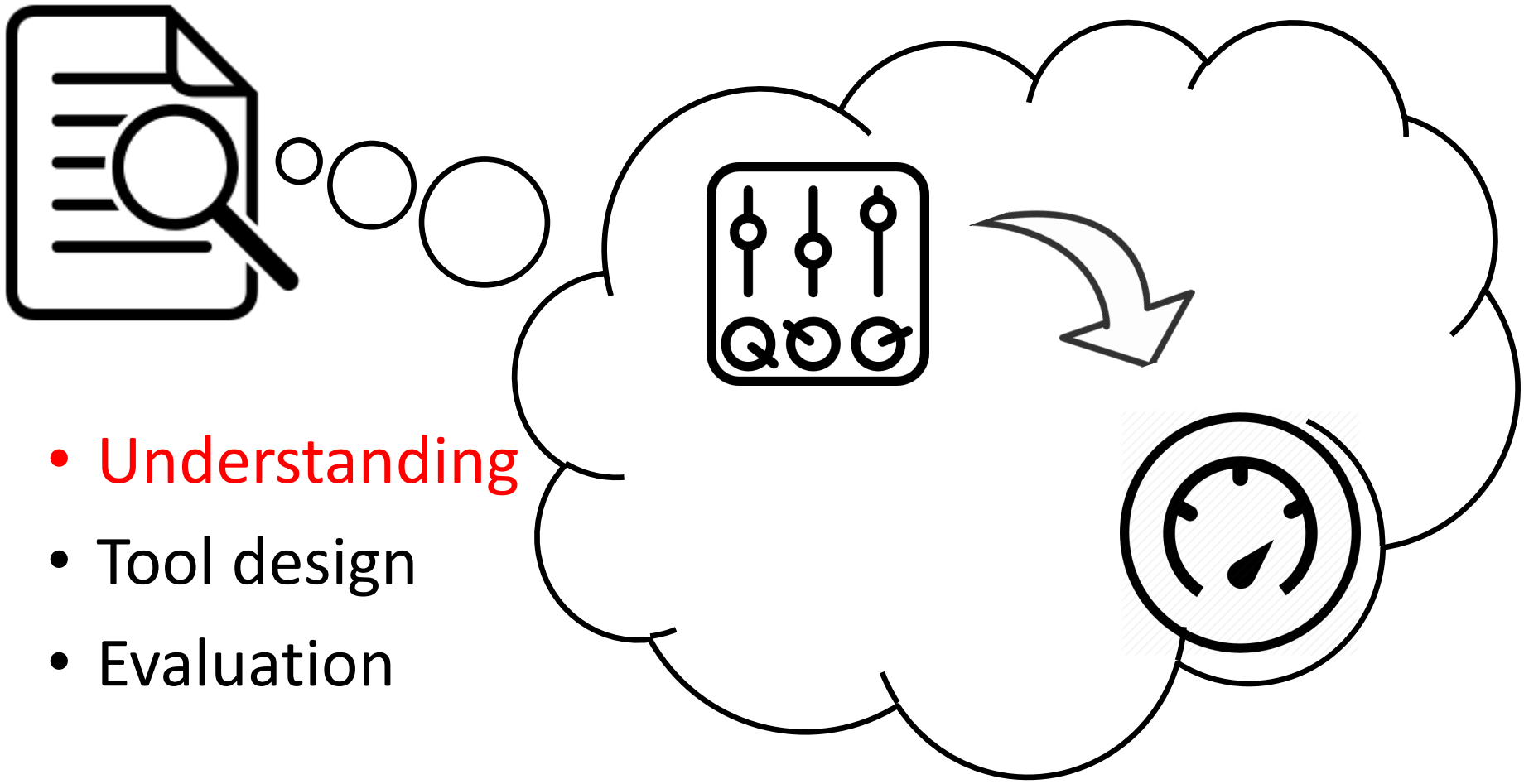
PerfOp

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



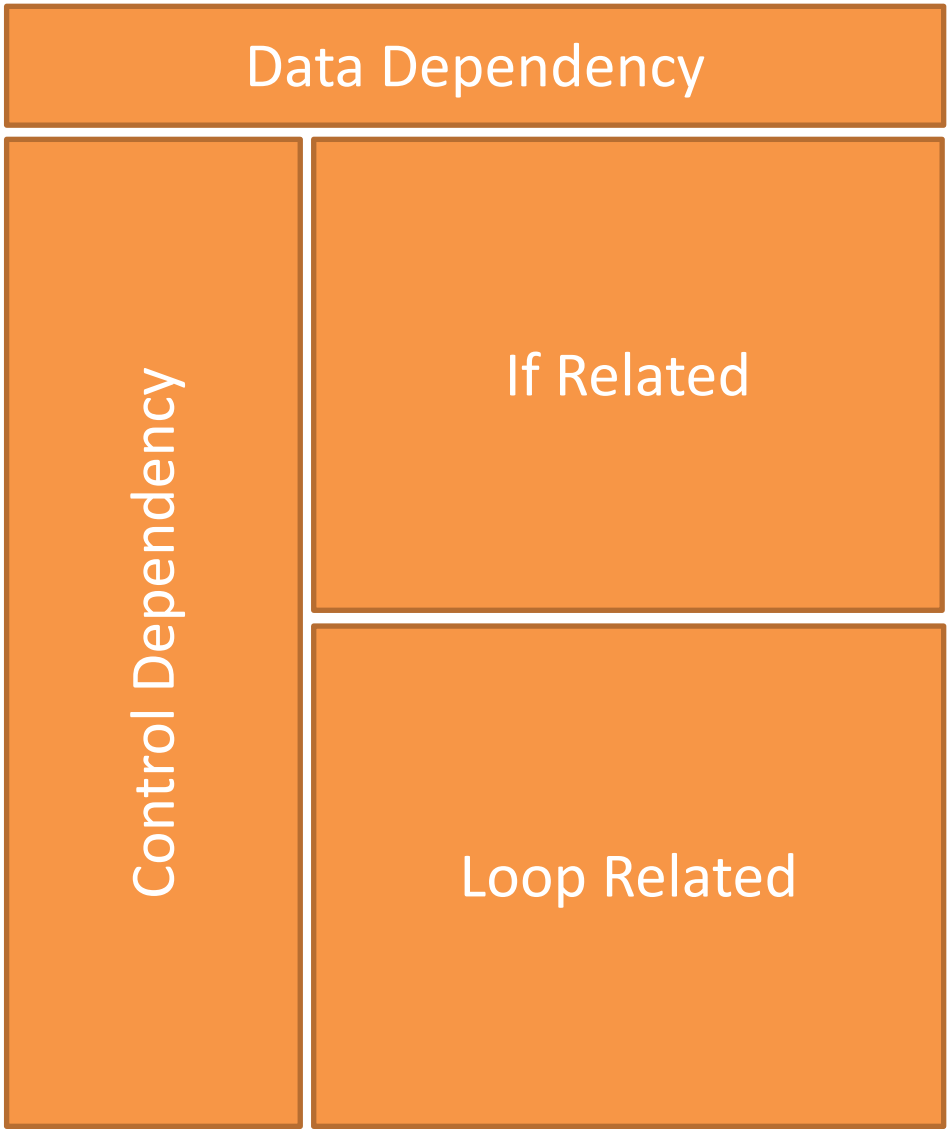
- Understanding
- Tool design
- Evaluation

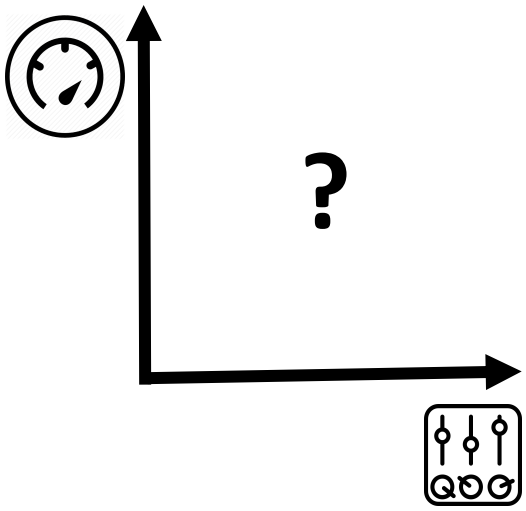
How to use program analysis to infer configurations' performance impact?



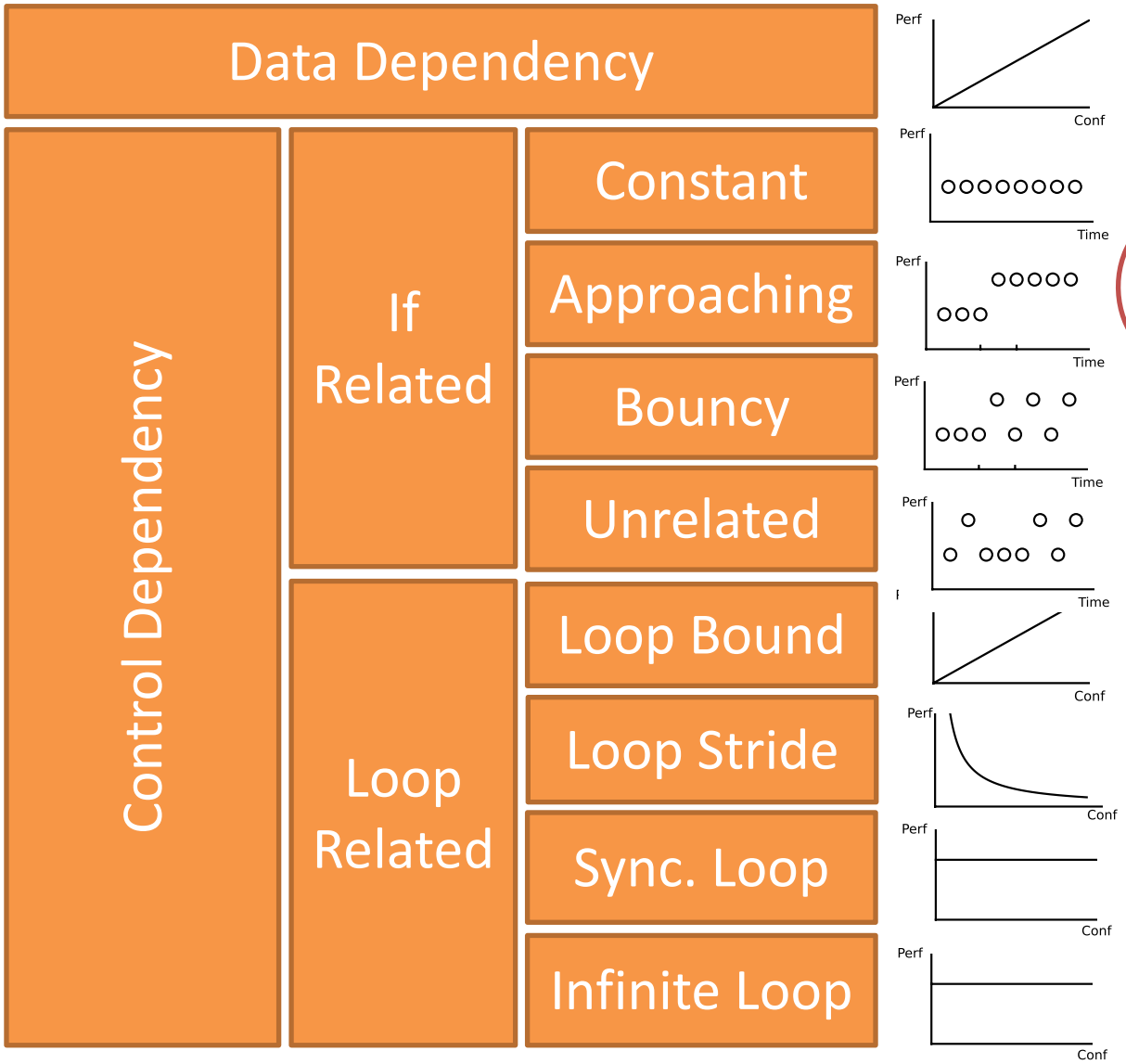
- Understanding
- Tool design
- Evaluation

How can a Conf affect a Perf-Op?





How can a Conf affect performance?

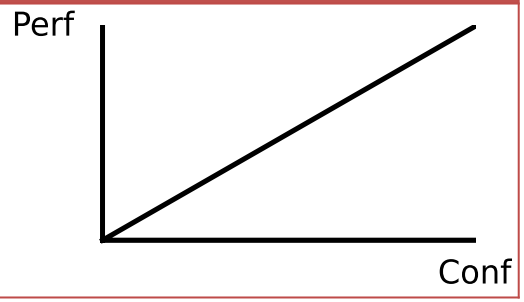


What type of variables are compared in IF predicate?

How does Conf affect loop?

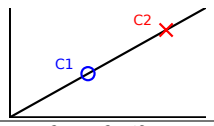
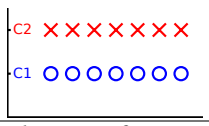
Data Dependency

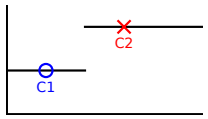
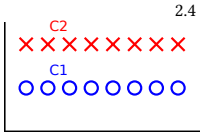
- Configuration affects the **impact of every instance** of PerfOp through parameters

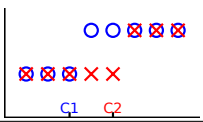
Code Example	Formula	Performance Graph
PerfOp(Conf)	$Performanc = Conf$	

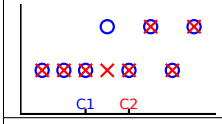

```
int sortmb = job.getInt("io.sort.mb");
int maxUsage = sortmb * 1024 * 1024;
buffer = new Byte[maxUsage];
```

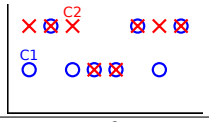
Table 1. Configurations' Performance-Impact Taxonomy. To ease the discussion, all the formulas, figures, and explanations in the table focus on the Toy Examples, where *C* is the value of a configuration and Perf refers to the memory-consumption contribution of one instance of the toy-example code snippet. The *Perf—Conf* figure in the 4th column depicts how the Perf (the y-axis) might change with the configuration setting (the x-axis); the *Perf–Time* figure in the 5th column depicts how the Perf (the y-axis) might change over time (the x-axis) under different configuration settings. In both columns' figures, we use C1 (blue dots) and C2 (red crosses) to represent two different configuration settings, and C1 < C2.

Pattern	Toy Example	Formula	Perf—Conf	Perf—Time	Explanation
Data Dependency — PerfConf affects the amount of performance contribution from one PerfOp					
1 Data	1.1 `new byte[C];`	1.2 Perf = C	1.3 	1.4 	*C* affects the impact of one `new byte[]` linearly.
Control Dependency (IF) — PerfConf affects whether a PerfOp is executed					

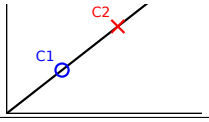
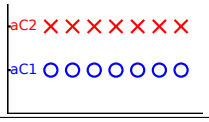
2 w/ Constant	2.1
	`if(V<=C)`
	`  new byte[a];`
	`else new byte[b];`
		2.2
		$\text{Perf} = \begin{cases} a & V \le C \\ b & V > C \end{cases}$
			2.3
			 
				2.4
				 
					C affects whether execute `new byte[a]` or `new byte[b]` statically.

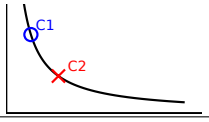
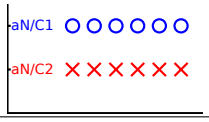
3 w/ getting-closer var.	3.1
	`if (V<=C) {`
	`  V++; new byte[a];`
	`} else`
	`  new byte[b];`
				3.4
				 
					C affects when to switch from `new byte[a]` to `new byte[b]`.

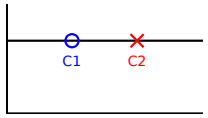
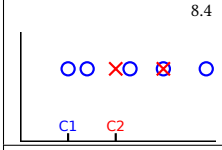
4 w/ back& forth var.	4.1
	`if (V<=C) {`
	`  V++; new byte[a];`
	`} else {`
	`  V--; new byte[b];`
	`}`
				4.4
				 
					C affect how quickly switch from `new byte[a]` to `new byte[b]`.

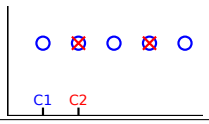
5 w/ unrelated var.	5.1
	`if(V<=C)`
	`  new byte[a];`
	`else new byte[b];`
				5.4
				 
					C affects the probability of `new byte[a]` executes over `new byte[b]`.

Control Dependency (LOOP) — PerfConf affects the frequency/#-of-times a PerfOp executes					

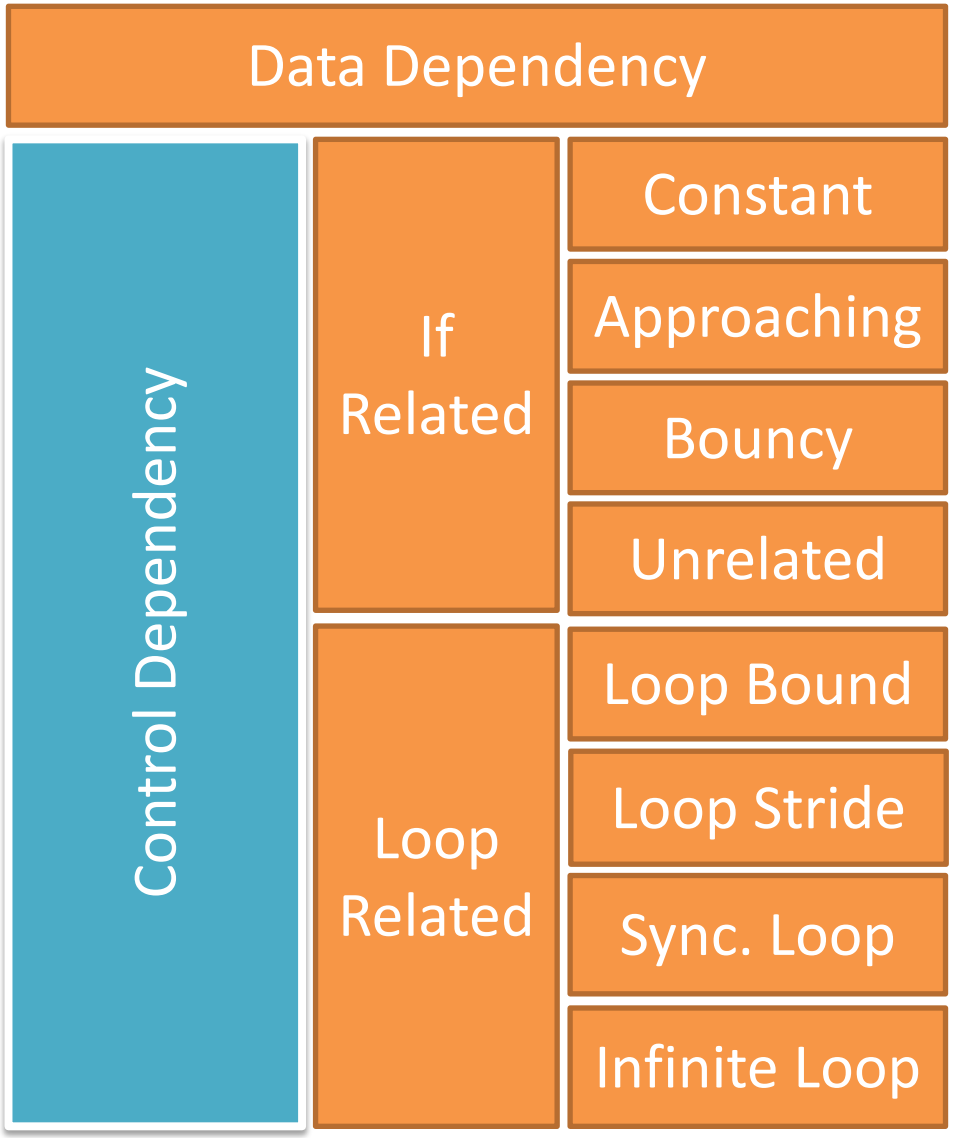
6 Regular loop bound	6.1
	`for(;i<C;i++) {`
	`  new byte[a];`
	`}`
		6.2
		Perf = a * C
			6.3
			 
				6.4
				 
					C affects the number of `new byte[a]` via loop bound.

7 Regular loop stride	7.1
	`for(;i<N;i+=C) {`
	`  new byte[a];`
	`}`
		7.2
		Perf = a * N /C
			7.3
			 
				7.4
				 
					C affects the number of `new byte[a]` via loop stride.

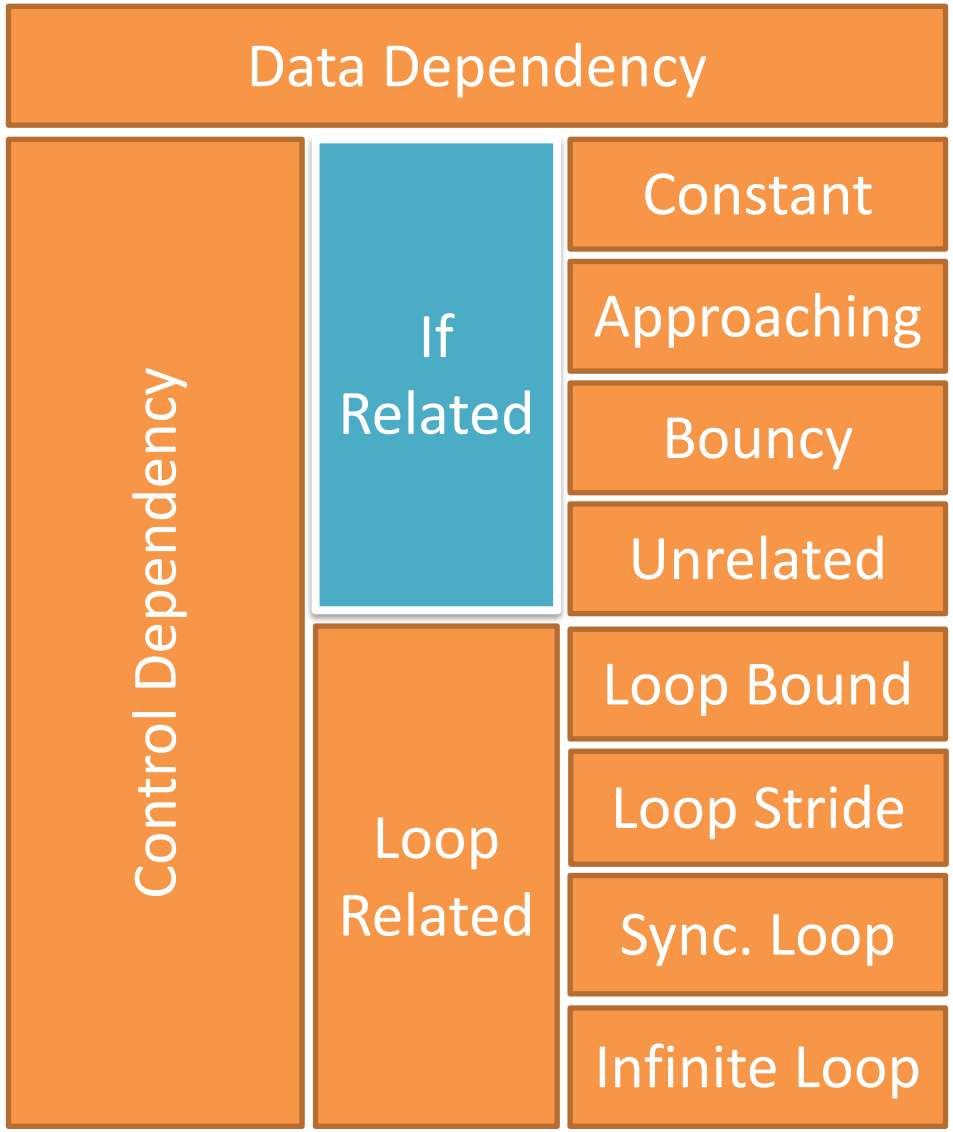
8 Sync loop	8.1
	`while (i<C) {`
	`  wait();`
	`}`
	`new byte[a];`
		8.2
		Perf = a
			8.3
			 
				8.4
				 
					C affects the frequency of `new byte[a]` via synchronization loop.

9 Infinite loop	9.1
	`new byte[a];`
	`sleep(C);`
	`//in infinite loop`
				9.4
				 
					C affects the frequency of `new byte[a]` via a fixed interval.

How can a Conf affect performance?



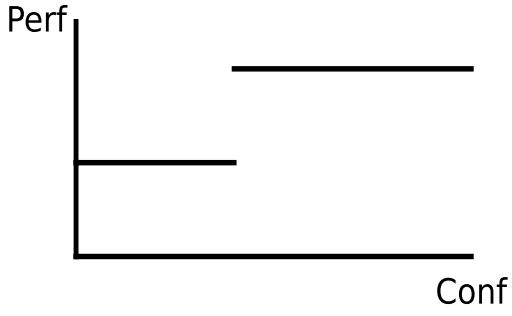
How can a Conf affect performance?



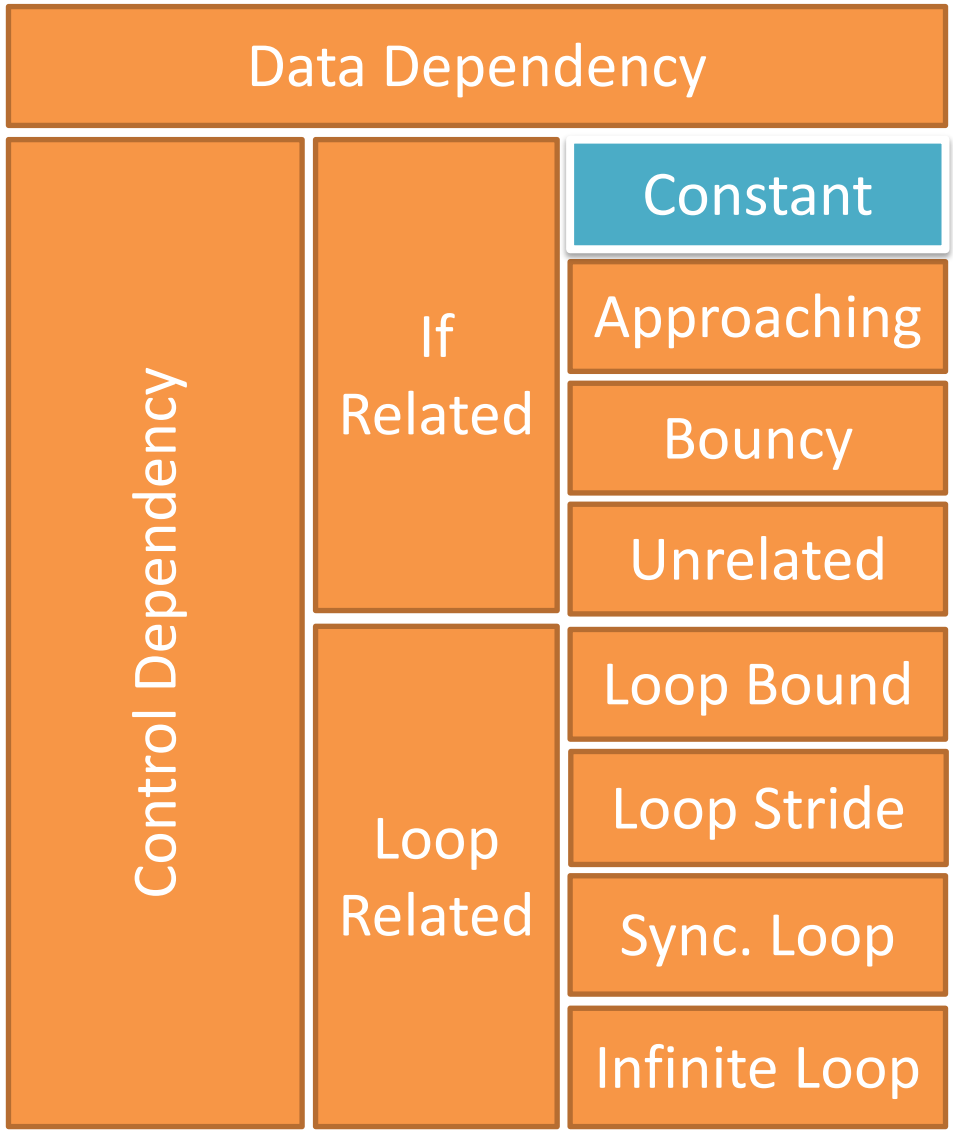
If Related Patterns

- Conf affects whether the PerfOp is executed

Code Example	Formula	Performance Graph

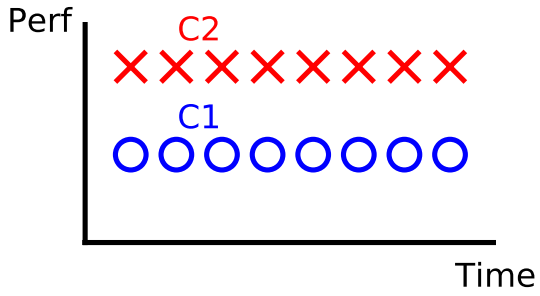
`If (V <= C) {`
`  PerfOpA`
`} else {`
`  PerfOpB`
`}`
	Performance $= \begin{cases} a, & V \leq C \\ b, & V > C \end{cases}$	

How can a Conf affect performance?

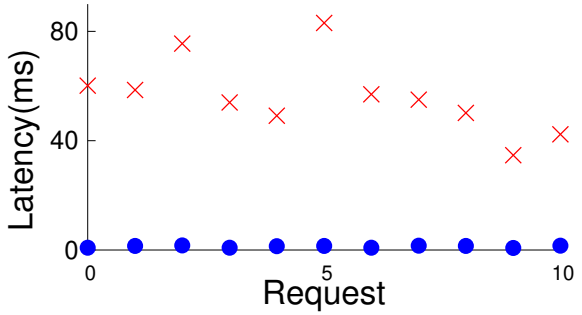


Compared with Constant

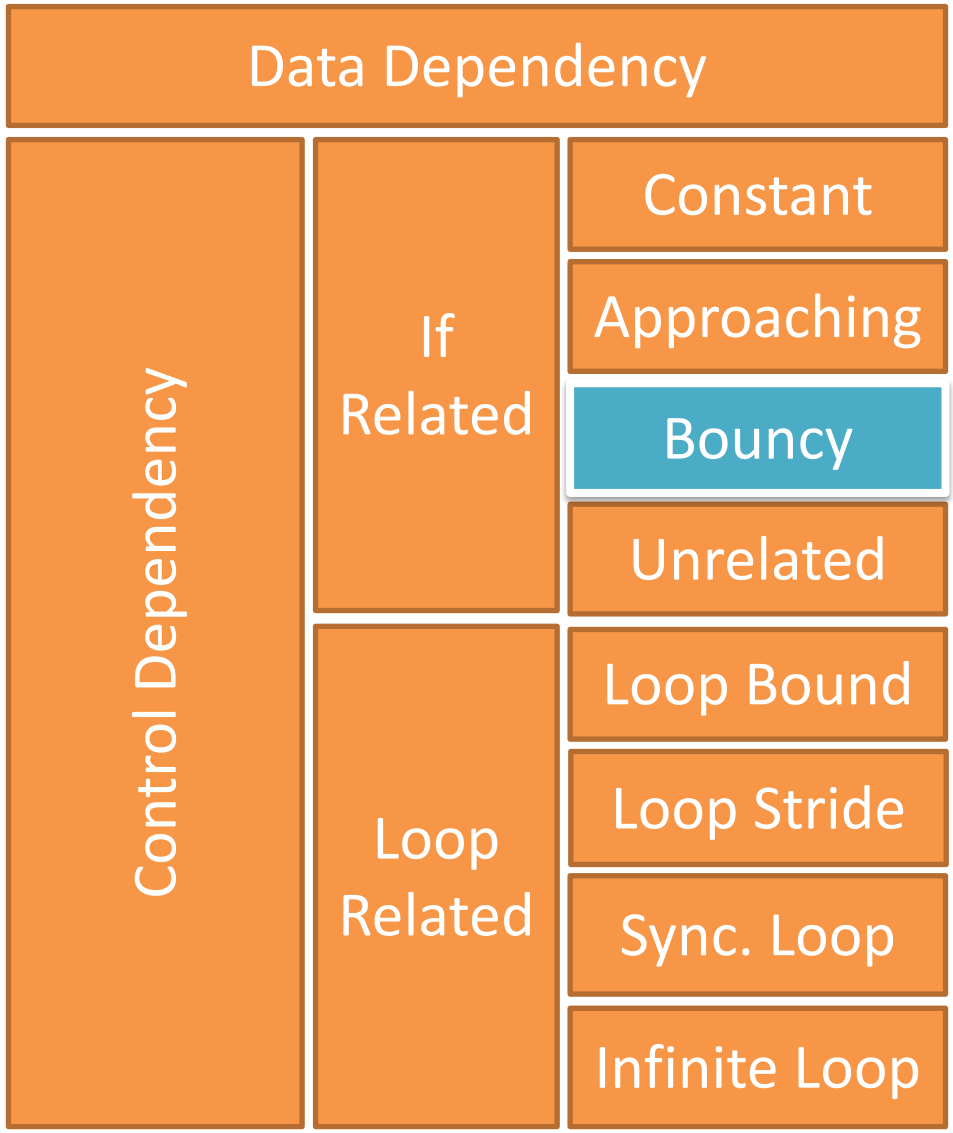
- The if-else decision does not change over time




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

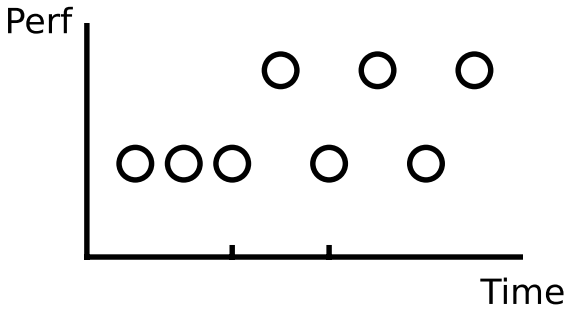


How can a Conf affect performance?

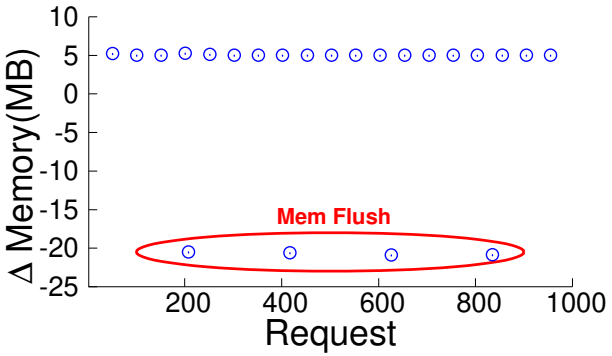


Compared with Bouncy Variable

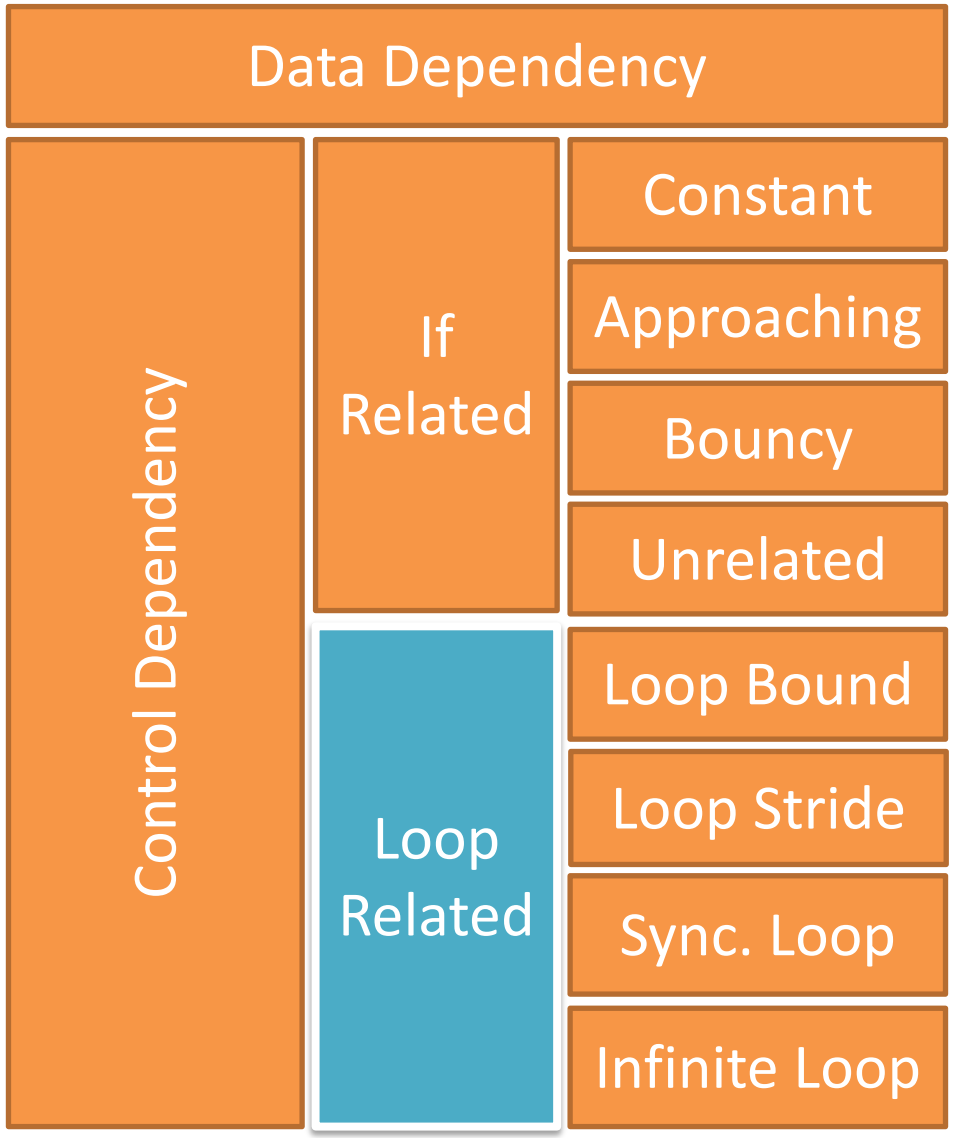
- The if-else decision keeps changing over time



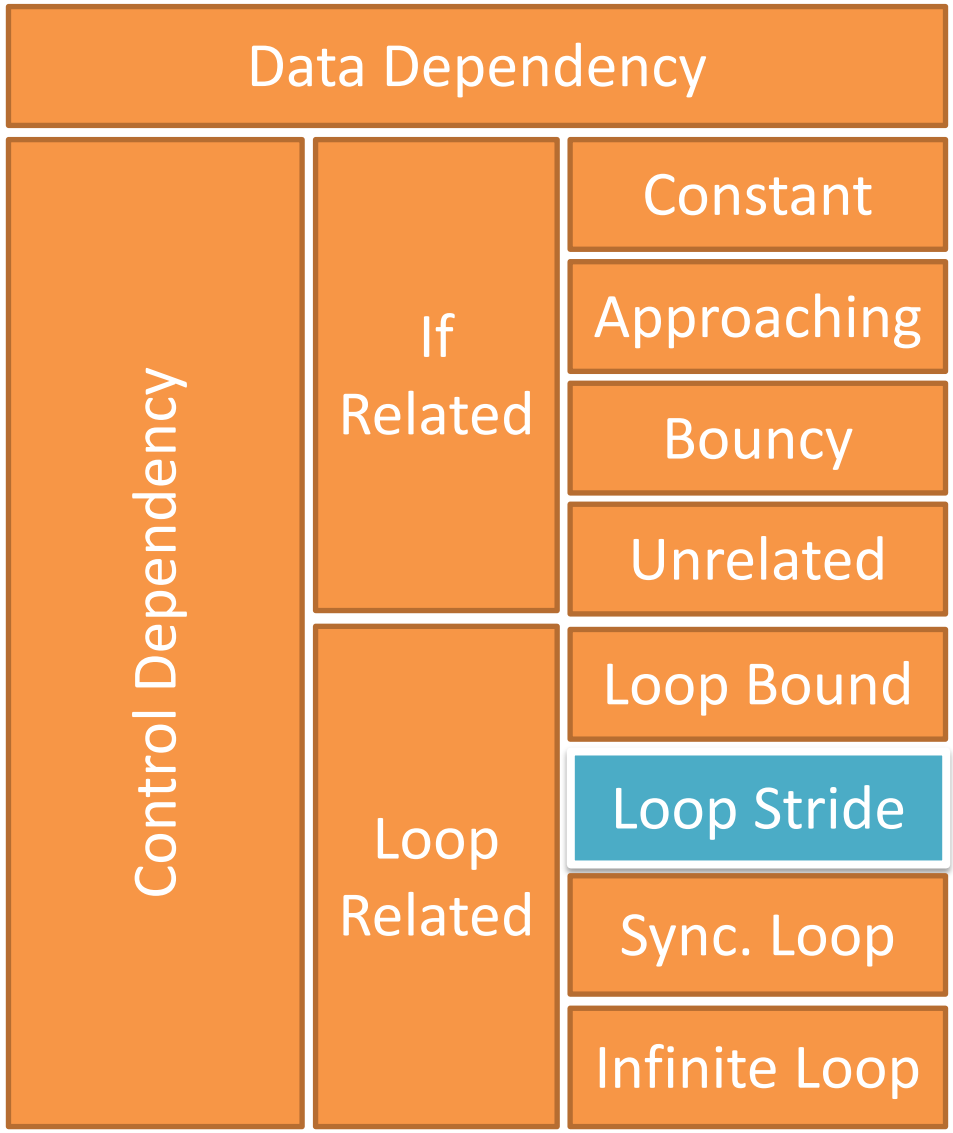

```
currentSize += put.heapSize();
writeBuffer.add(put);
If (currentSize > bufferSize) {
  writeBuffer.clear();
  currentSize = 0;
}
```



How can a Conf affect performance?



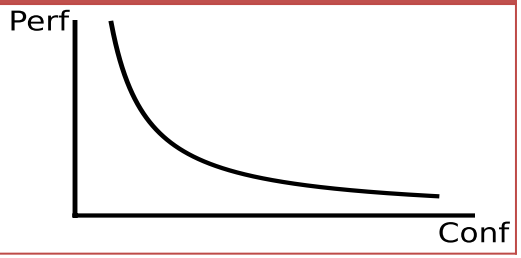
How can a Conf affect performance?



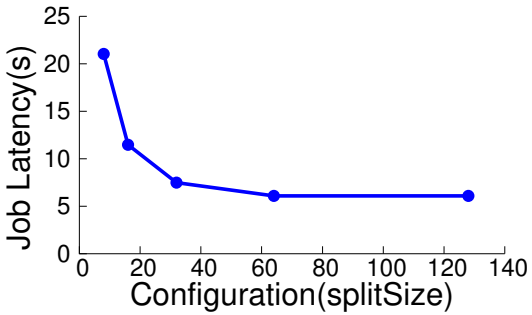
Affect Loop Stride

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

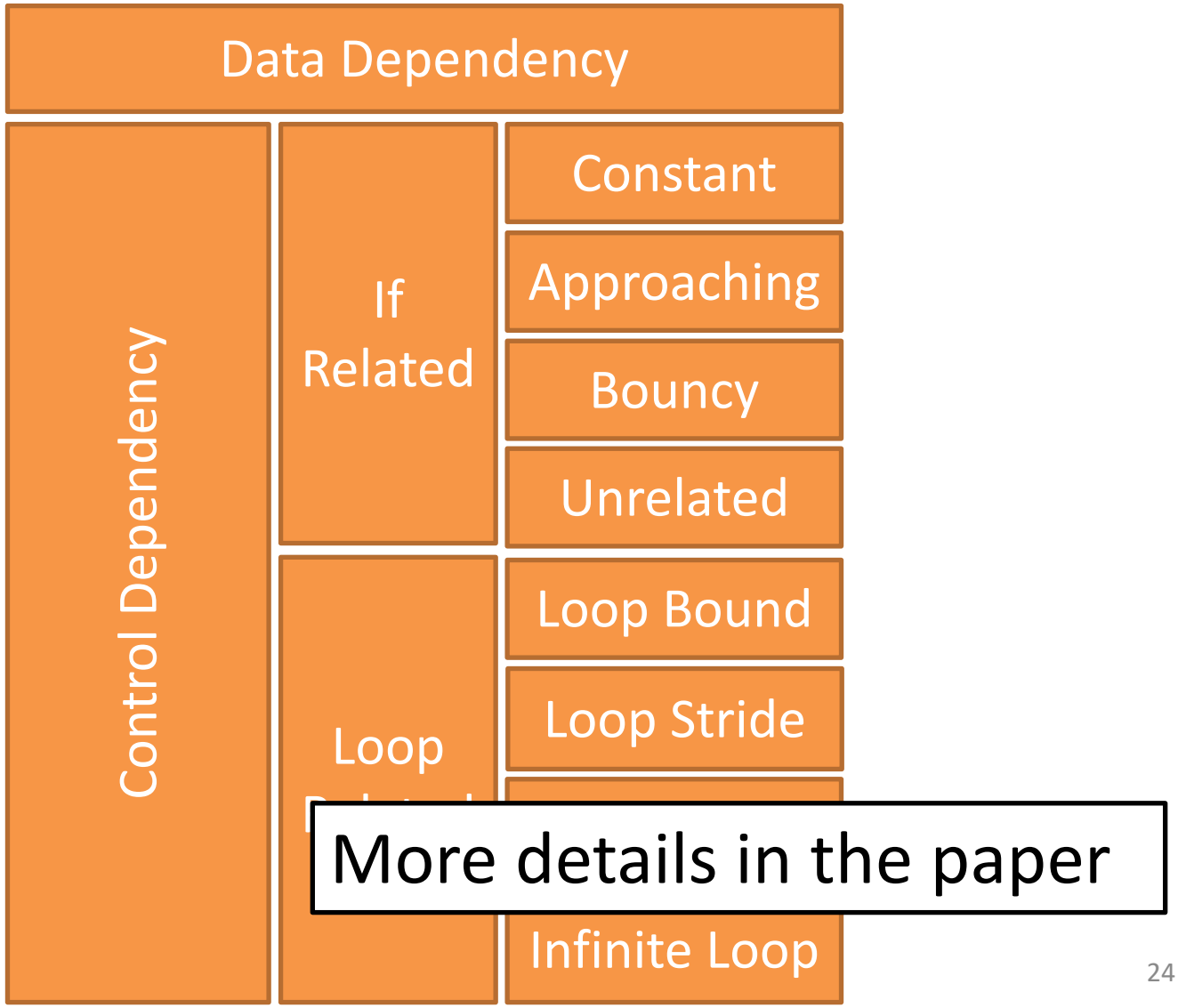
Code Example	Formula	Performance Graph

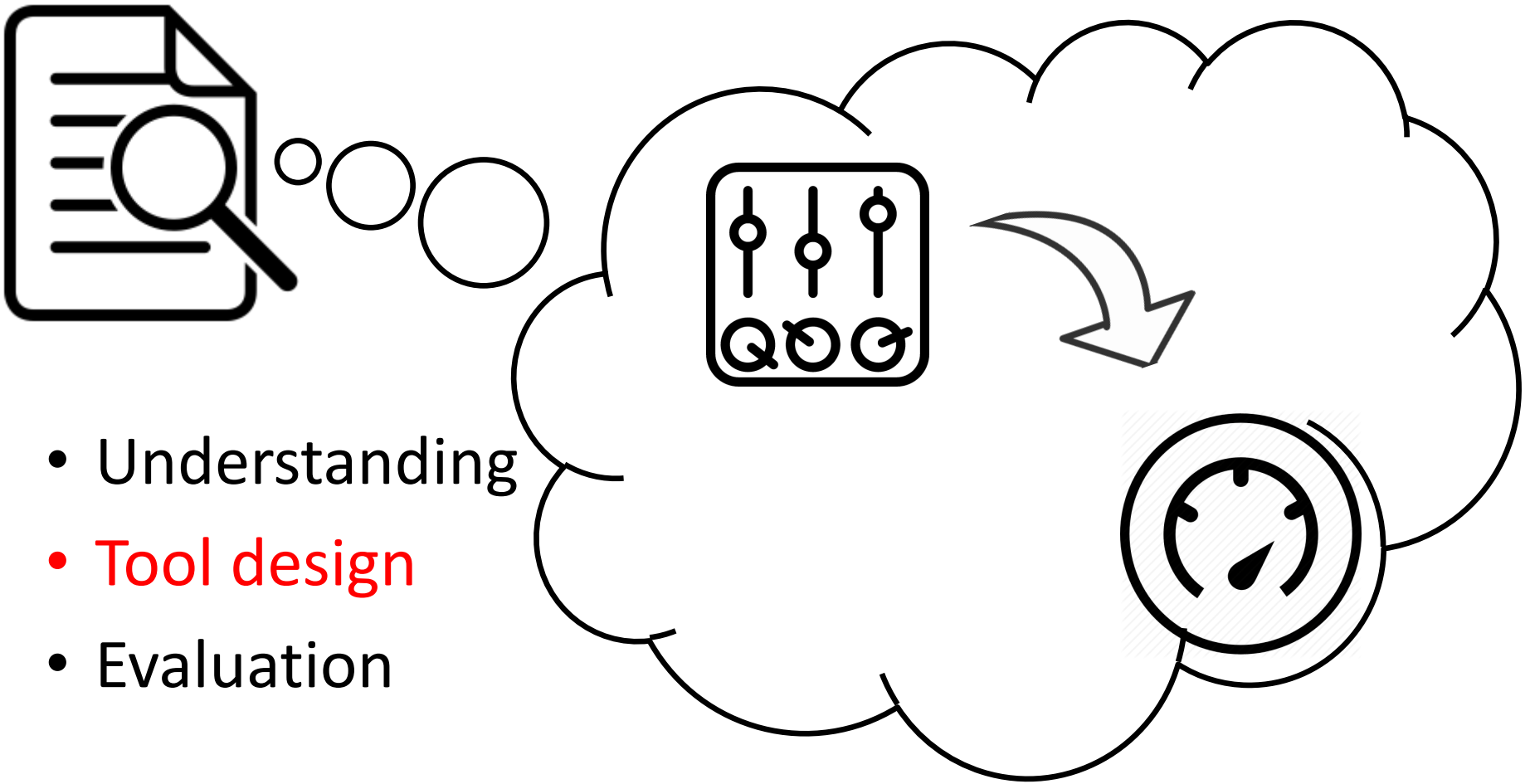
`for (; i < N; i+=Conf)`
`{`
`  PerfOp();`
`}`
	$Performance = aN/Conf$	

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



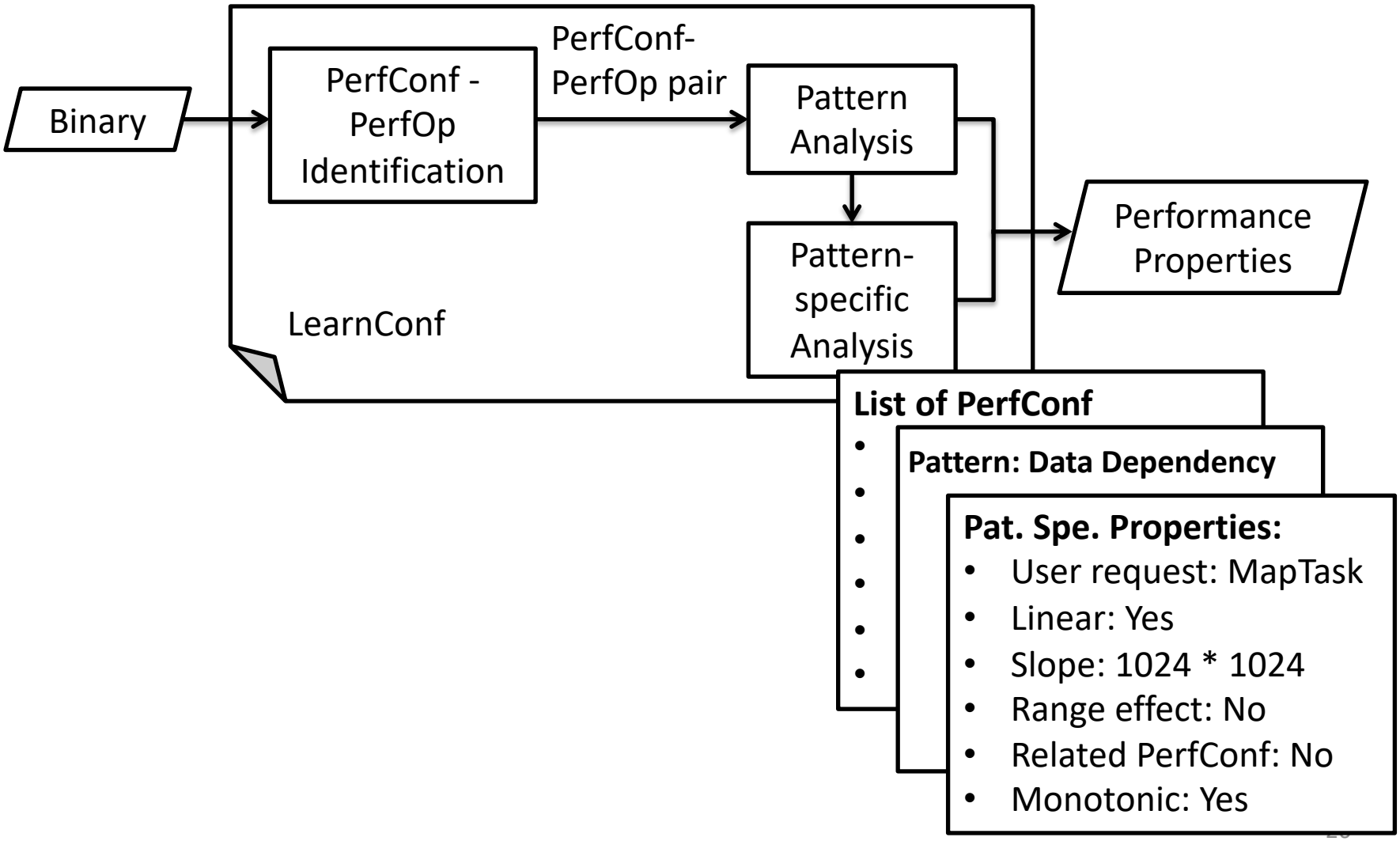
How can a Conf affect performance?





- 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];
```

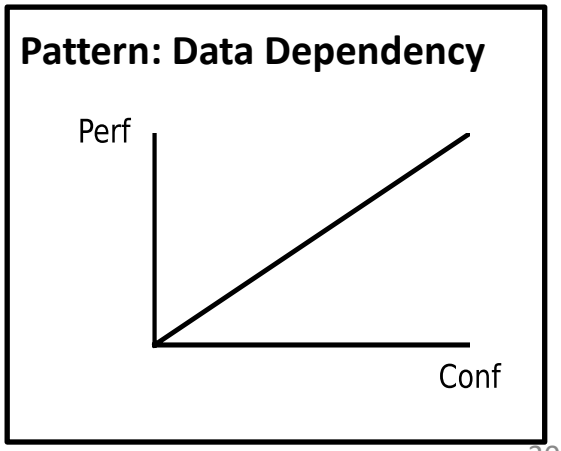
List of PerfConf

- io.sort.mb
- …

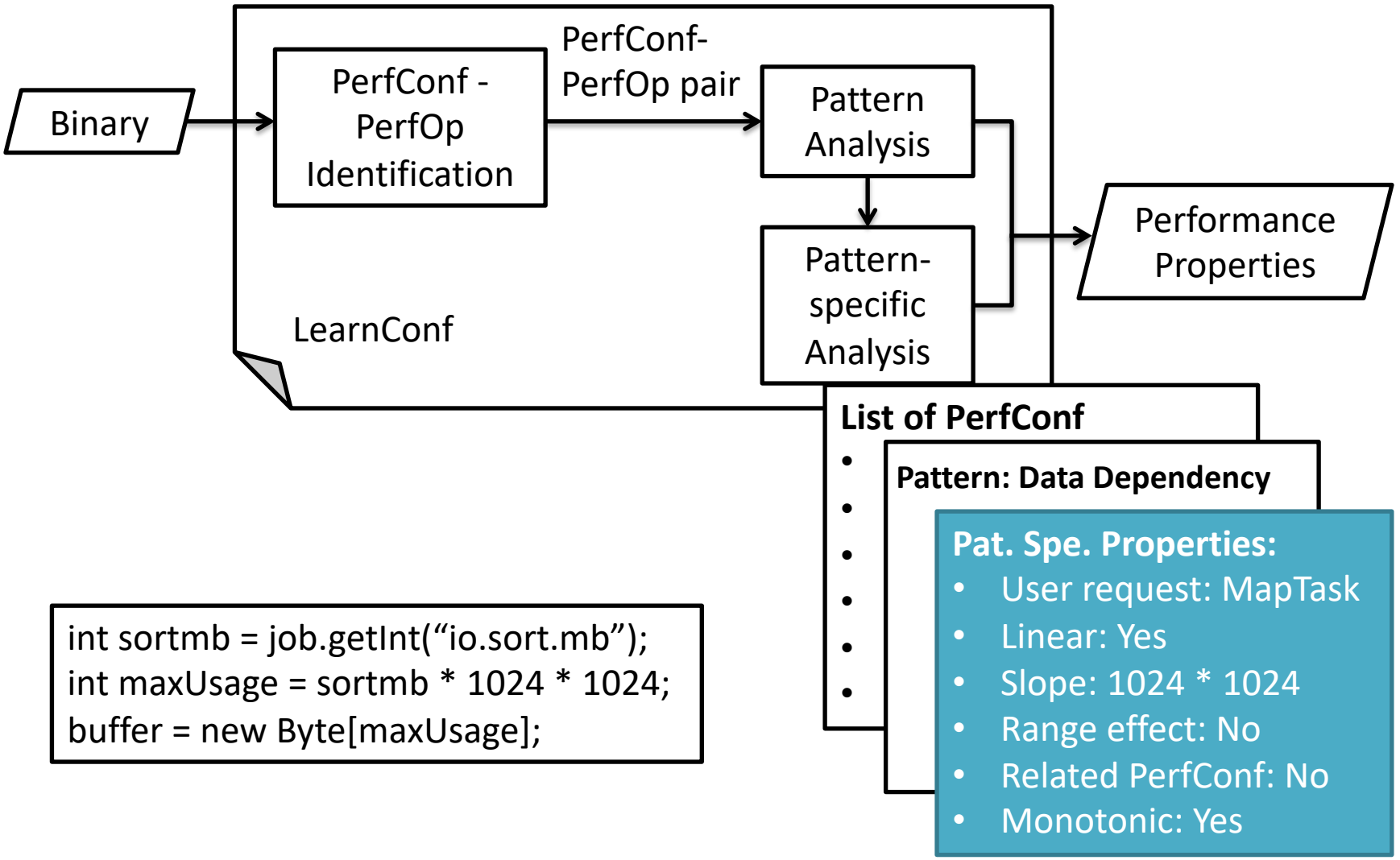
Categorize PerfConf-PerfOp dependency

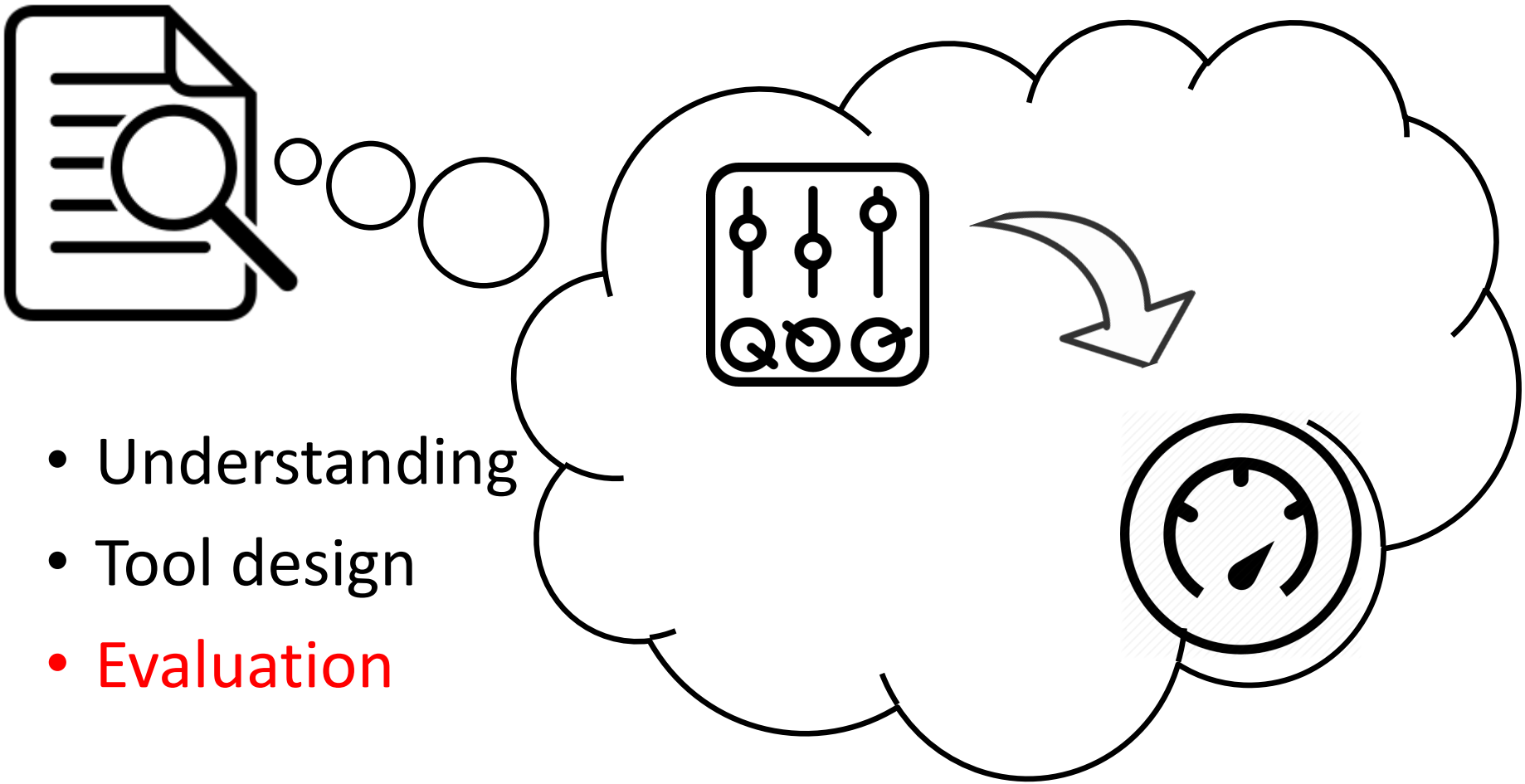
- Data Dependency Pattern
  - Conf used in the parameter of the PerfOp
- If Pattern
  - Conf used in an if-predicate
- Loop Pattern
  - Conf used in a loop-exit condition

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



Pattern-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!

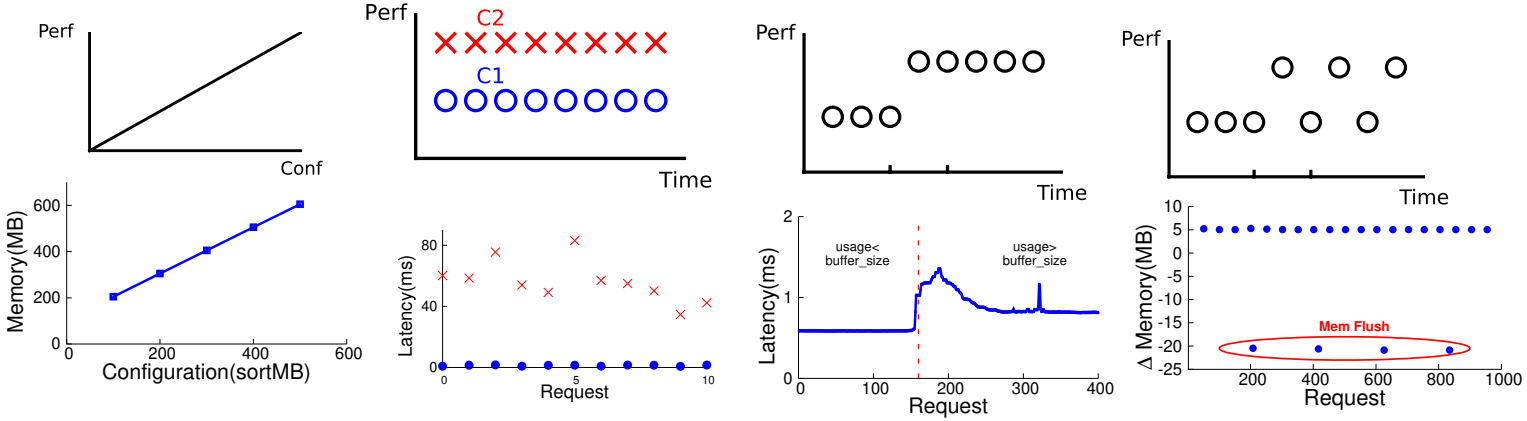
workload、memory consumption

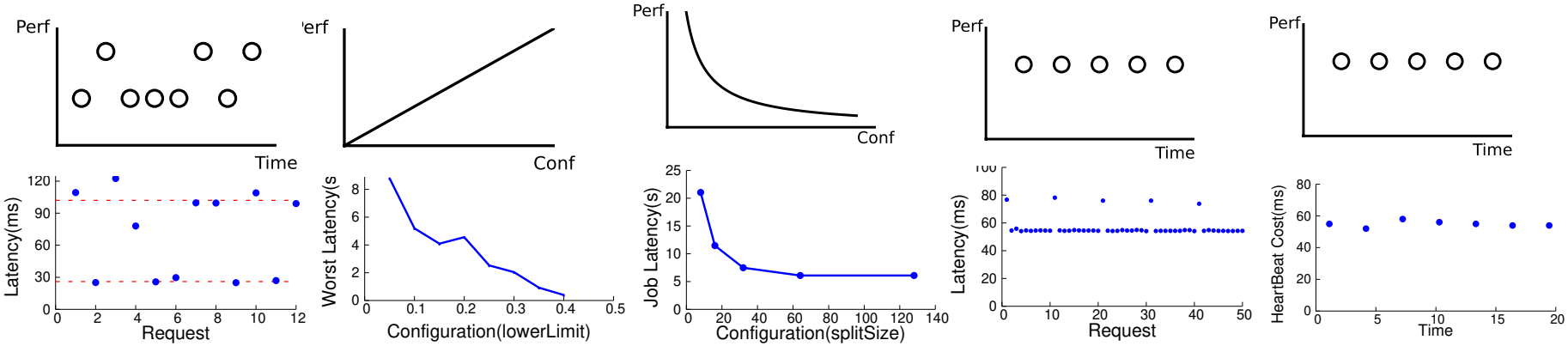
performance impact eliminated by software machinism

complicated control dependencies、IPC...

	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



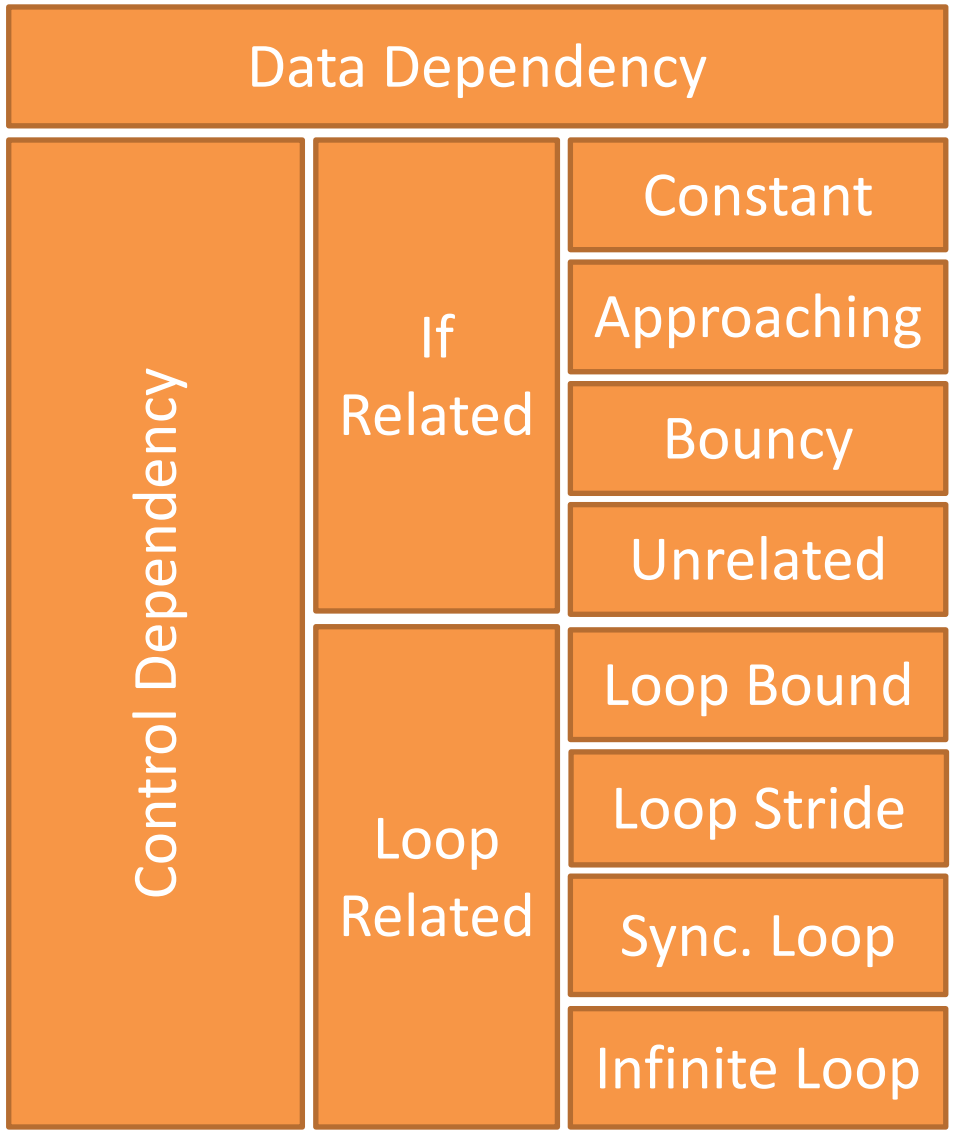


More Result

- 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
lichi@uchicago.edu