Performance Annotations for Cloud Computing

Daniele Rogora[*] Steffen Smolka[%] Antonio Carzaniga[*] Amer Diwan[$]
Robert Soulé[*~]

presented by
Daniele Rogora

[*]Università della Svizzera italiana [%]Cornell University [$]Google [~]Barefoot Networks

HotCloud 2017

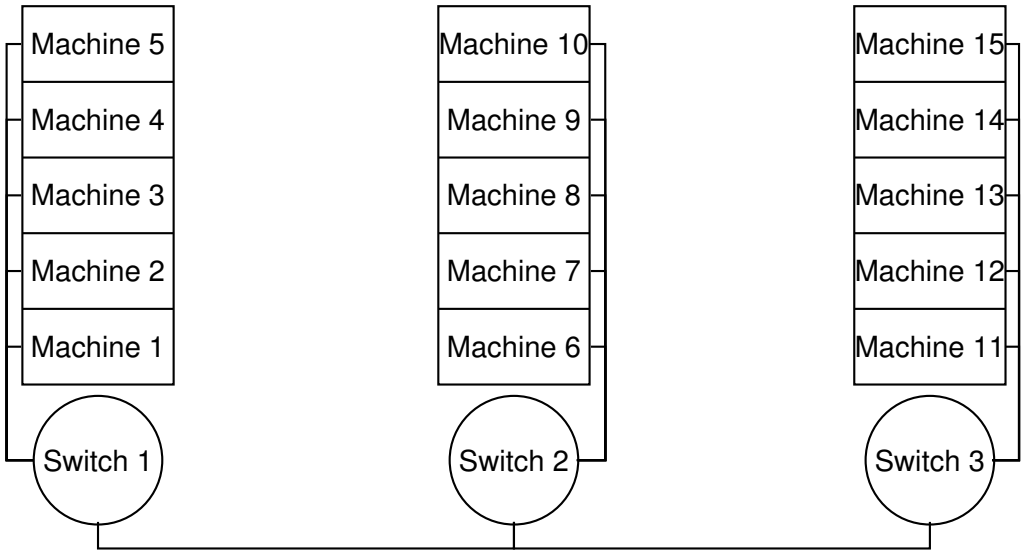
Data Centers Are Everywhere...

- Data centers provide
  - Services for end users
    - Google, Facebook, Dropbox
  - Services for companies and universities
    - AWS for SAP, cloud mail services
  - Raw processing power (IaaS, PaaS)
    - Amazon EC2, Microsoft Azure

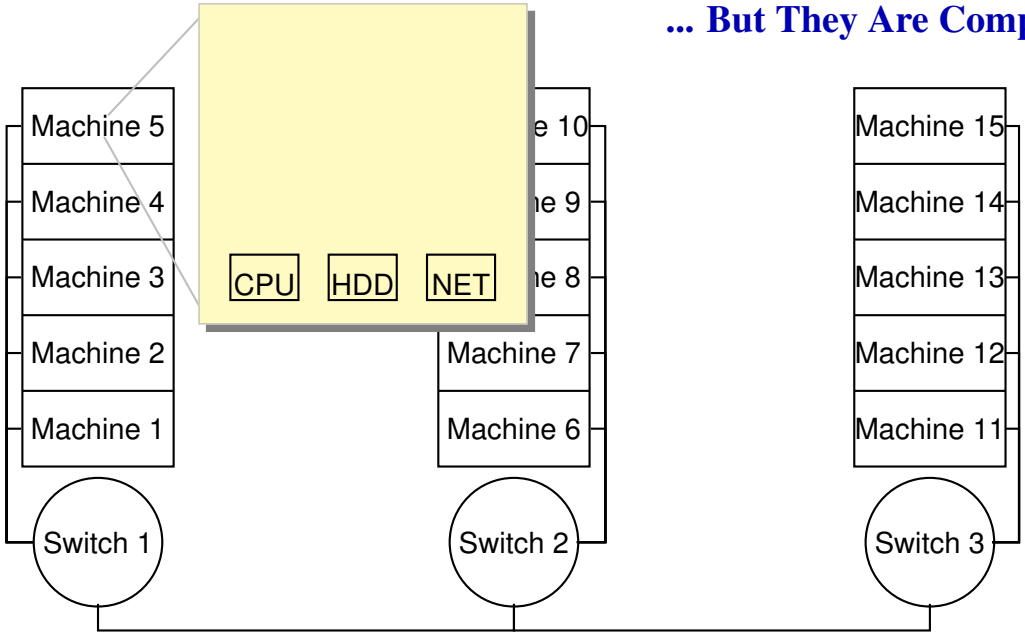
Data Centers Are Everywhere...

- Data centers provide
  - Services for end users
    - Google, Facebook, Dropbox
  - Services for companies and universities
    - AWS for SAP, cloud mail services
  - Raw processing power (IaaS, PaaS)
    - Amazon EC2, Microsoft Azure
- Data centers are widespread
  - Microsoft has more than 100 data centers
    - they account for more than 1M servers
  - Amazon has more than 30 data centers
    - they account for more than 1.5M servers
  - Google has 15 data centers scattered around the world
    - in 2013 they accounted for around 900k servers

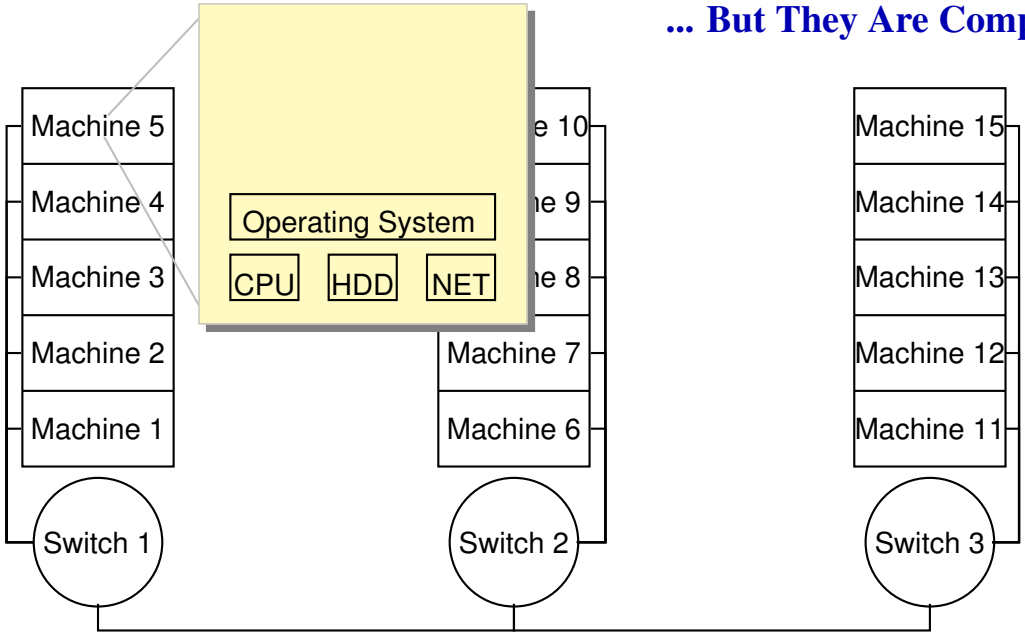
... But They Are Complex



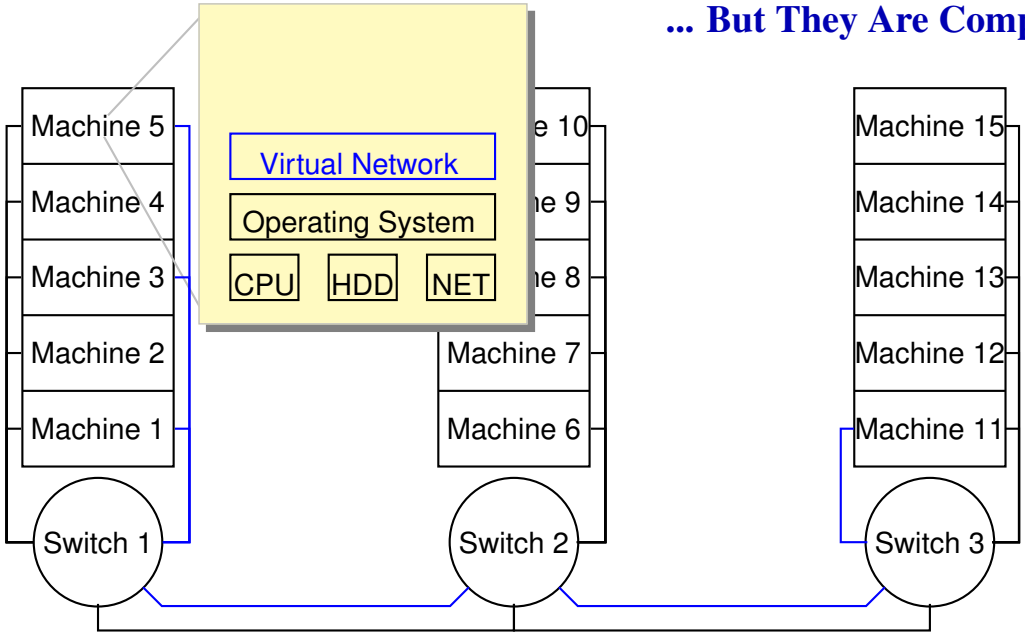
... But They Are Complex



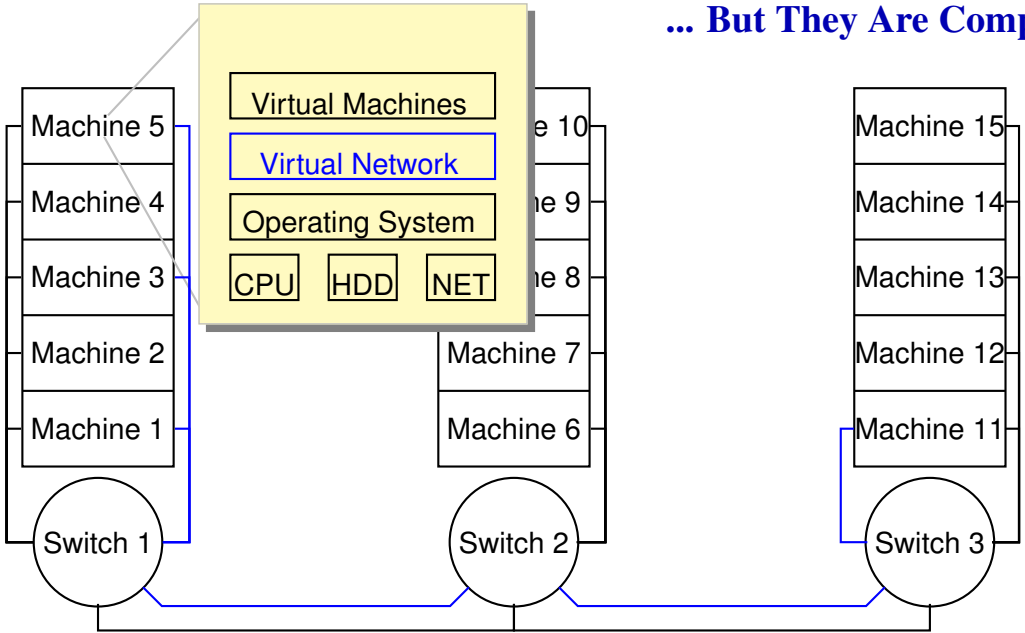
... But They Are Complex



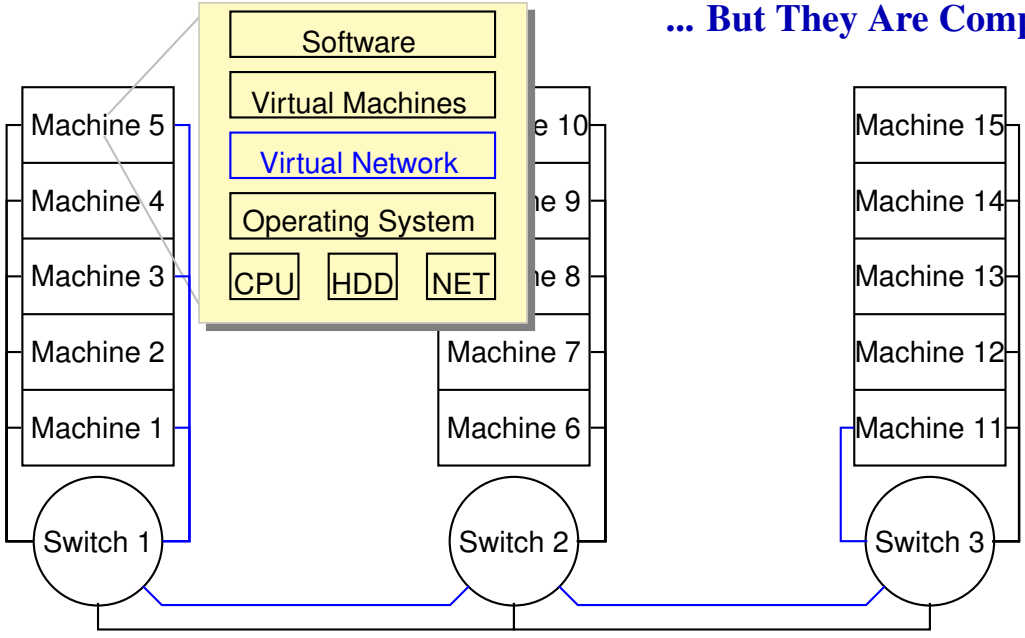
... But They Are Complex



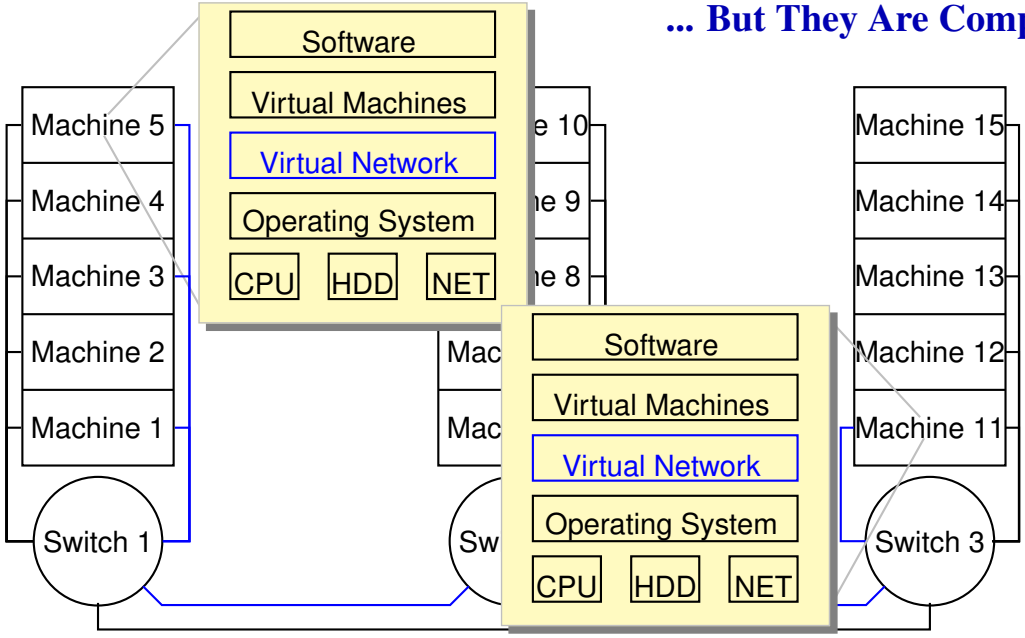
... But They Are Complex



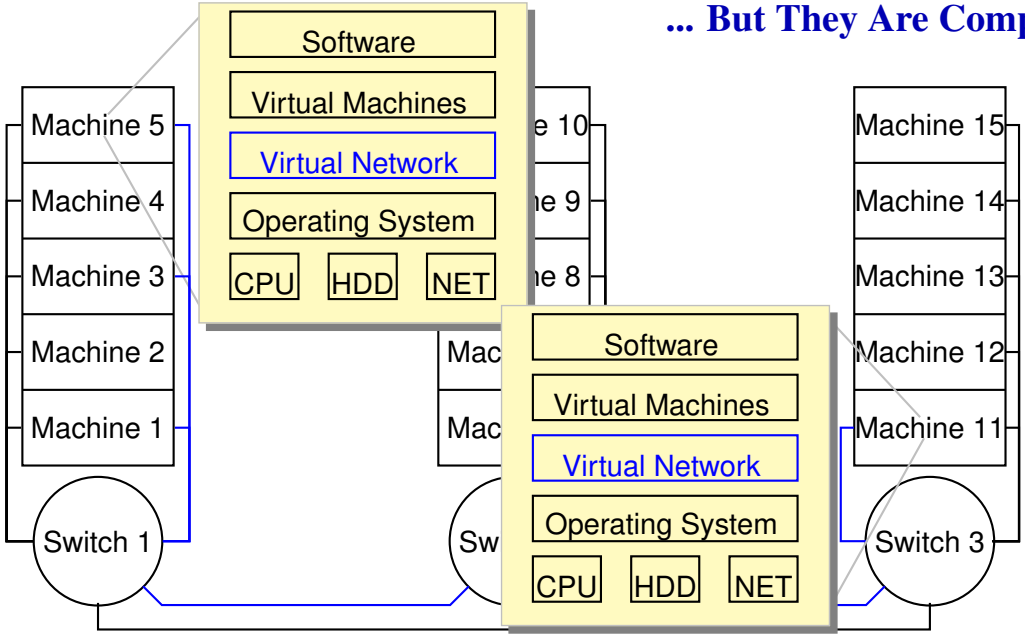
... But They Are Complex



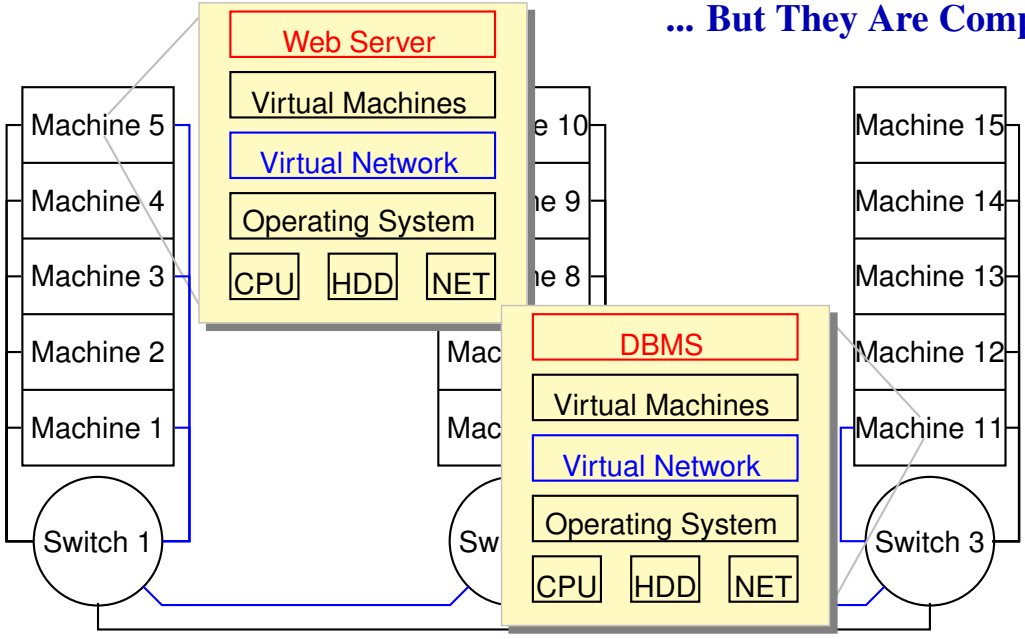
... But They Are Complex



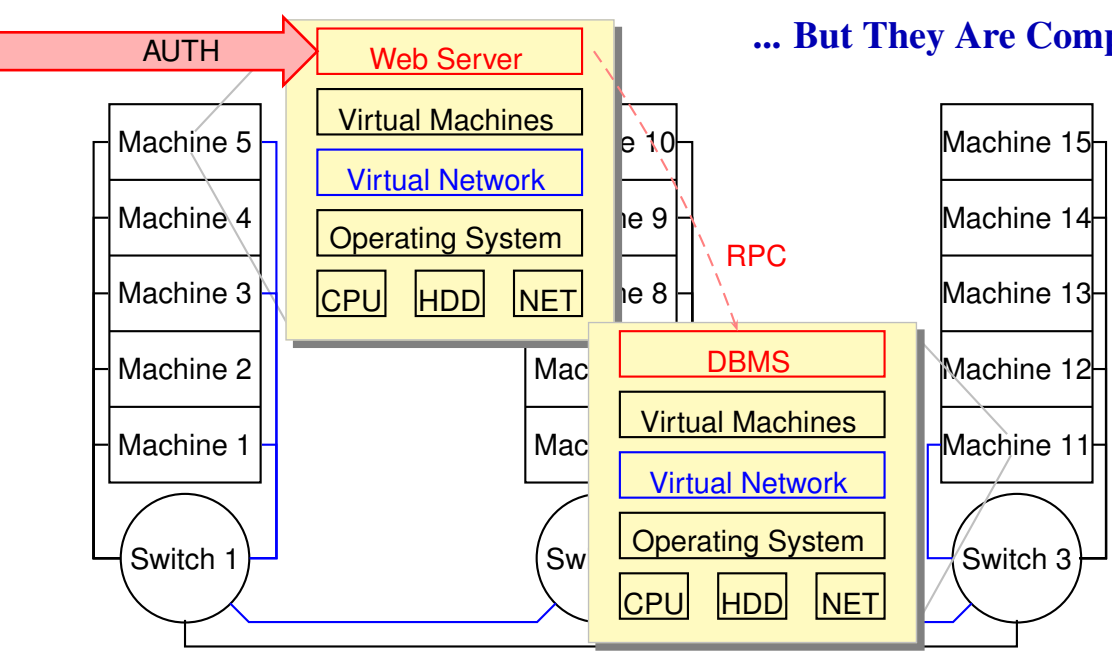
... But They Are Complex



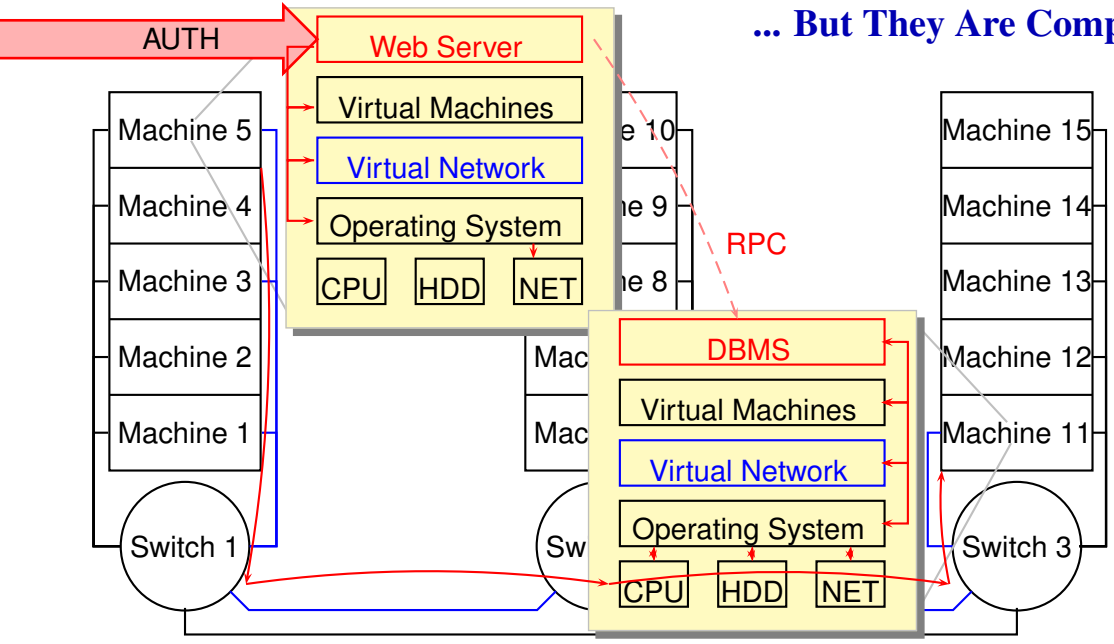
... But They Are Complex



... But They Are Complex



... But They Are Complex



Understanding the performance of a data center is difficult

- many layers
- limited scope of tools and people's knowledge

Understanding the performance of a data center is difficult

- many layers
- limited scope of tools and people's knowledge

3 real-world questions by a data center operator

How much load increase can we support with the current setup? Where will the bottleneck be? What would break first?

How much would it help to move the database server to faster hardware, or directly on the metal?

Can we understand and explain unexpected behaviors?

Goal: Creating a New Model

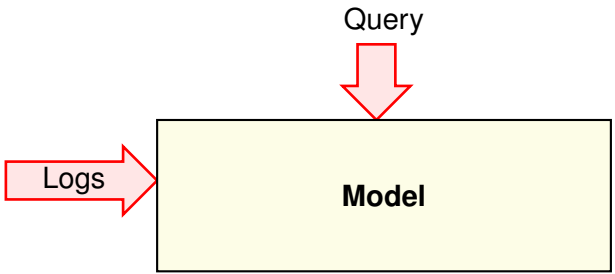
We want to build a ***dynamic performance model for data centers***

- comprehensive
- live
- interactive

Goal: Creating a New Model

We want to build a ***dynamic performance model for data centers***

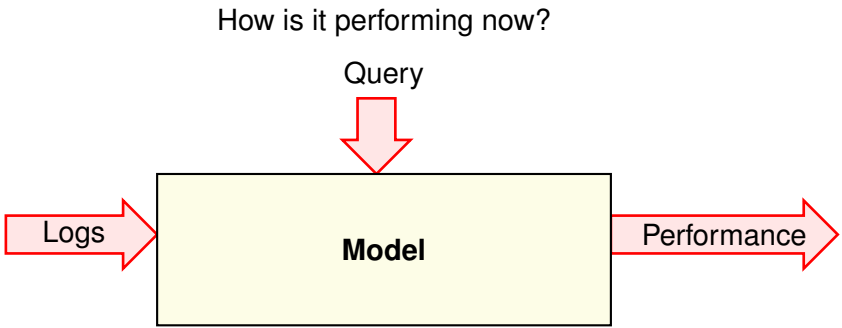
- comprehensive
- live
- interactive



Goal: Creating a New Model

We want to build a ***dynamic performance model for data centers***

- comprehensive
- live
- interactive

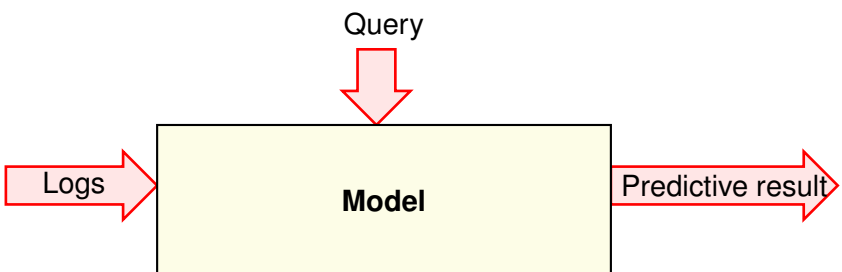


Goal: Creating a New Model

We want to build a ***dynamic performance model for data centers***

- comprehensive
- live
- interactive

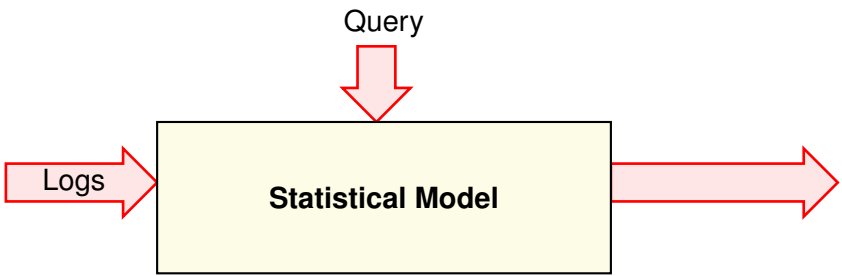
What if the we improve the hardware?



Goal: Creating a New Model

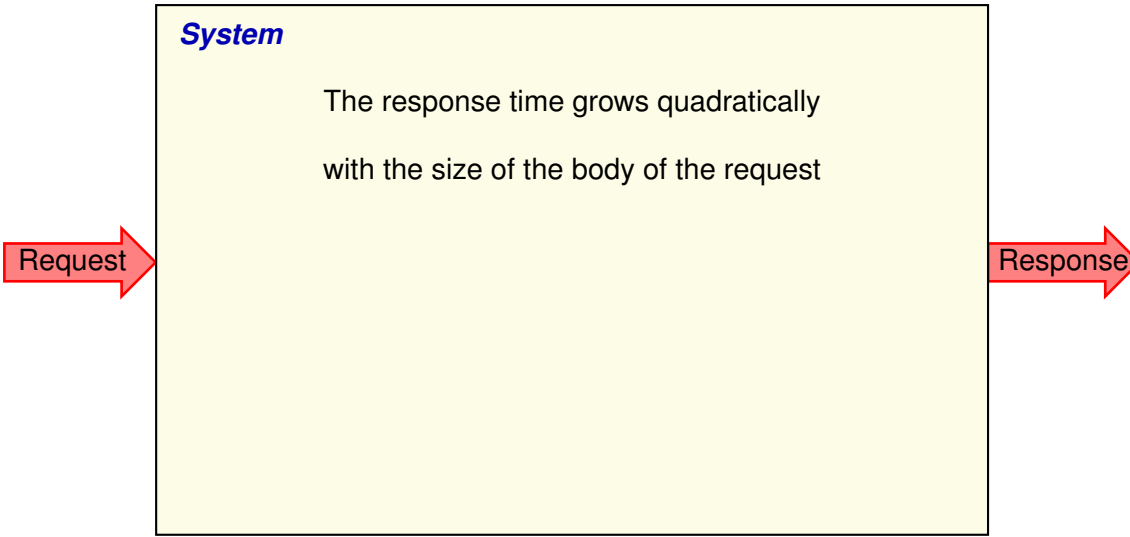
We want to build a ***dynamic performance model for data centers***

- comprehensive
- live
- interactive



Performance Annotations

Example: a Web Service



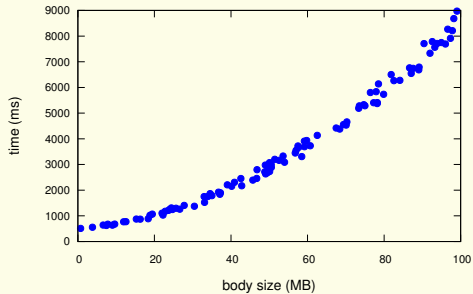
Example: a Web Service

Request

System

The response time grows quadratically

with the size of the body of the request



Response

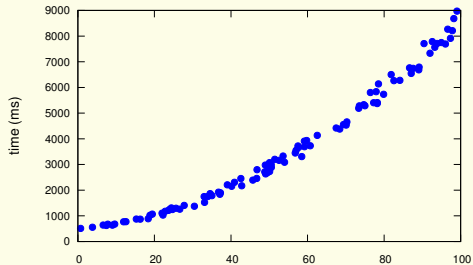
Example: a Web Service

Request

System

The response time grows quadratically

with the size of the body of the request



$$@Time \sim a * |in|^2 + b * |in| + c$$

Response

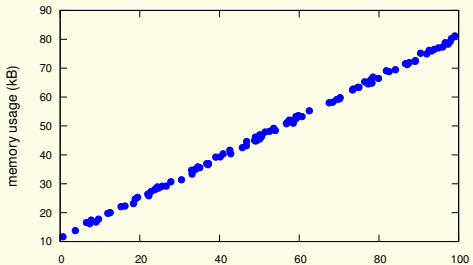
Example: a Web Service

Request

System

The memory used grows linearly

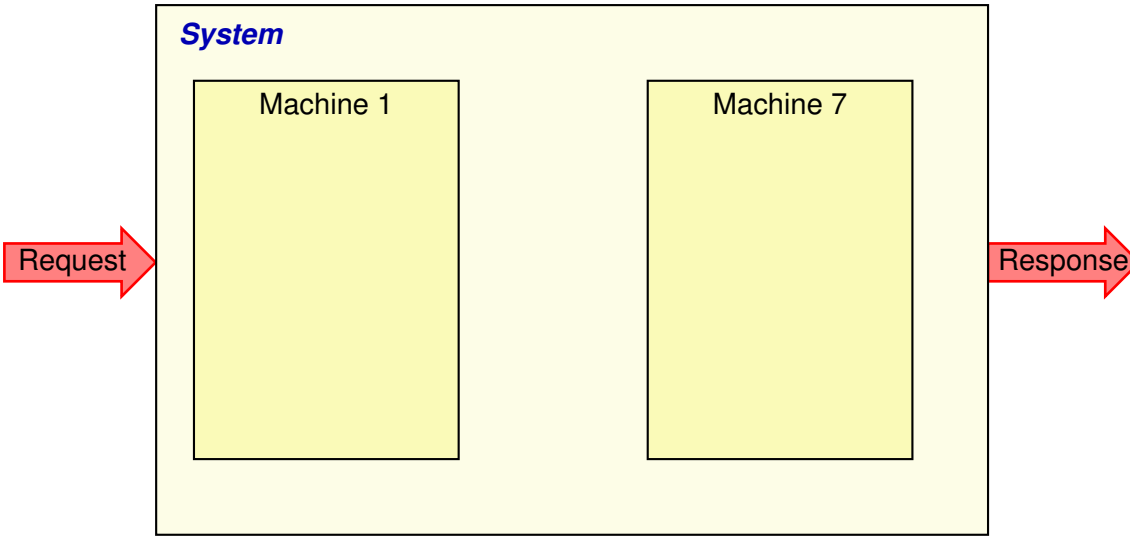
with the size of the body of the request



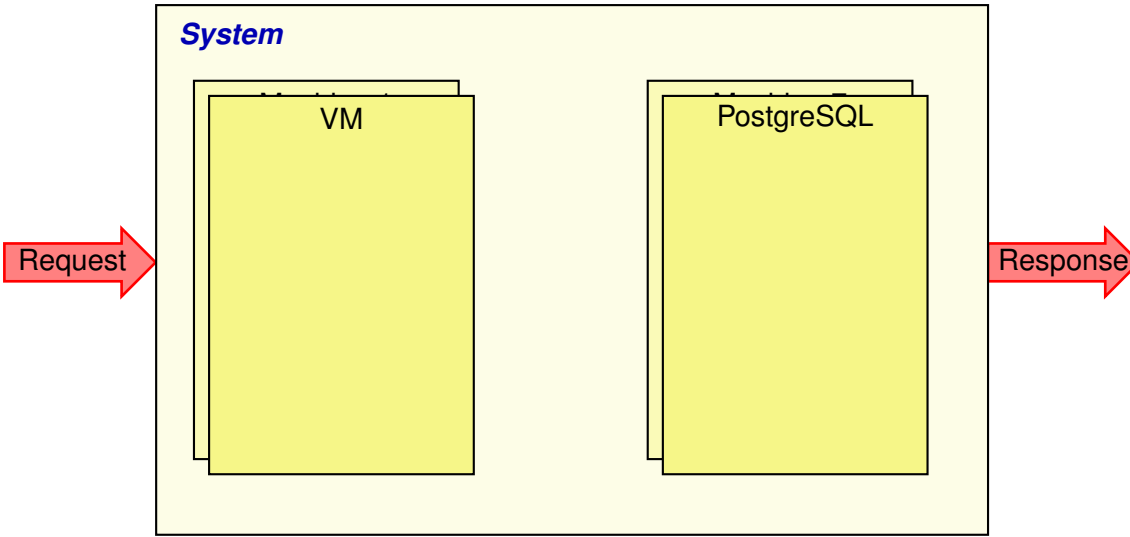
$$@Mem \sim a * |in| + b$$

Response

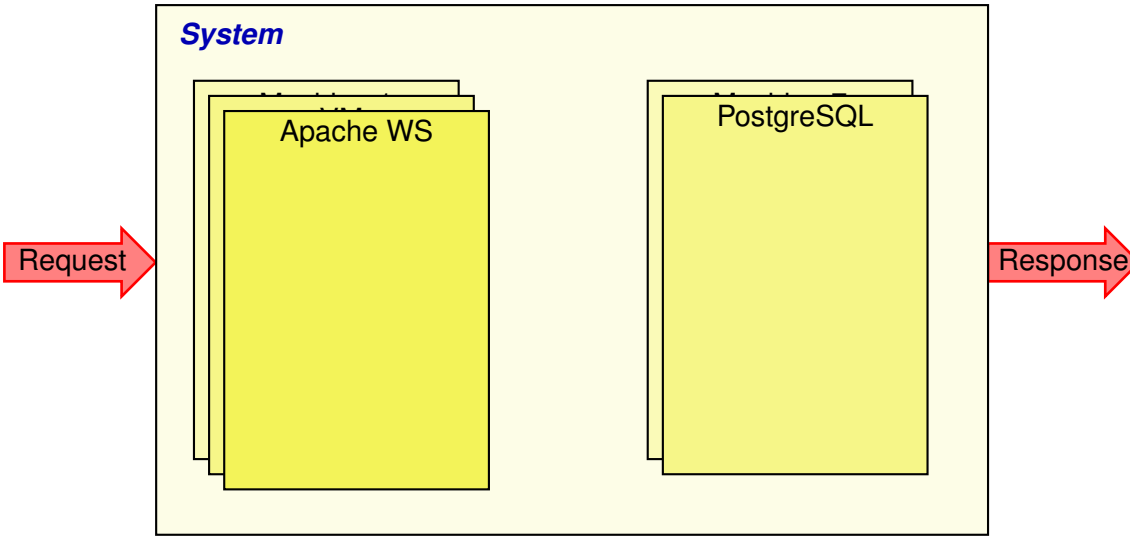
Example: a Web Service



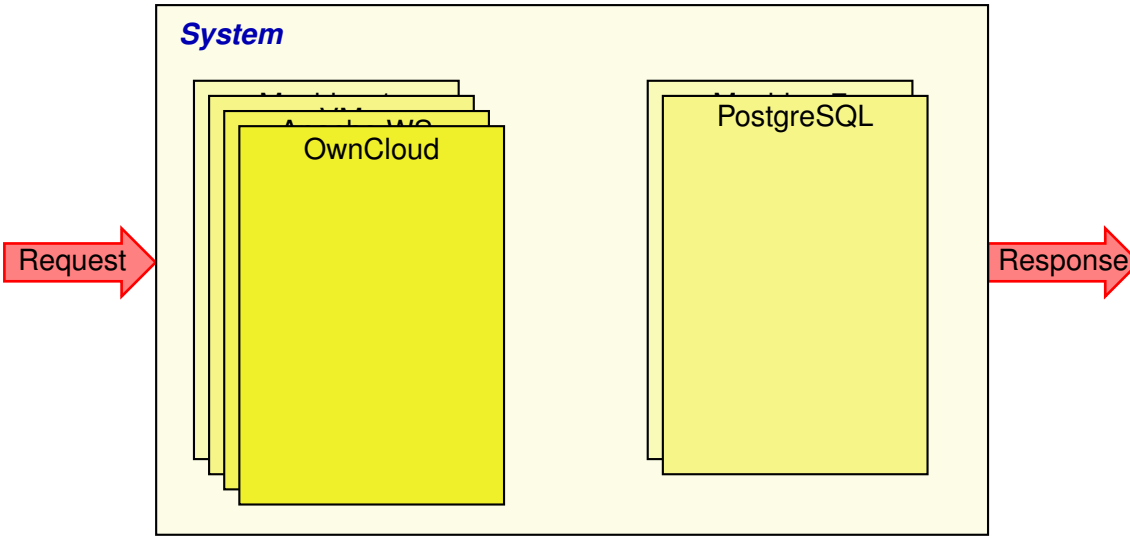
Example: a Web Service



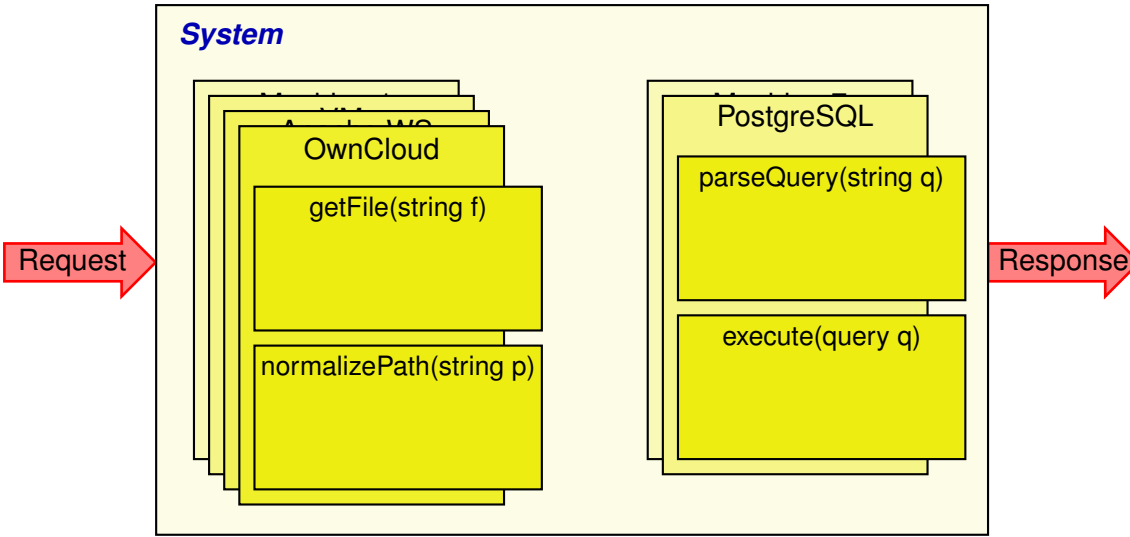
Example: a Web Service



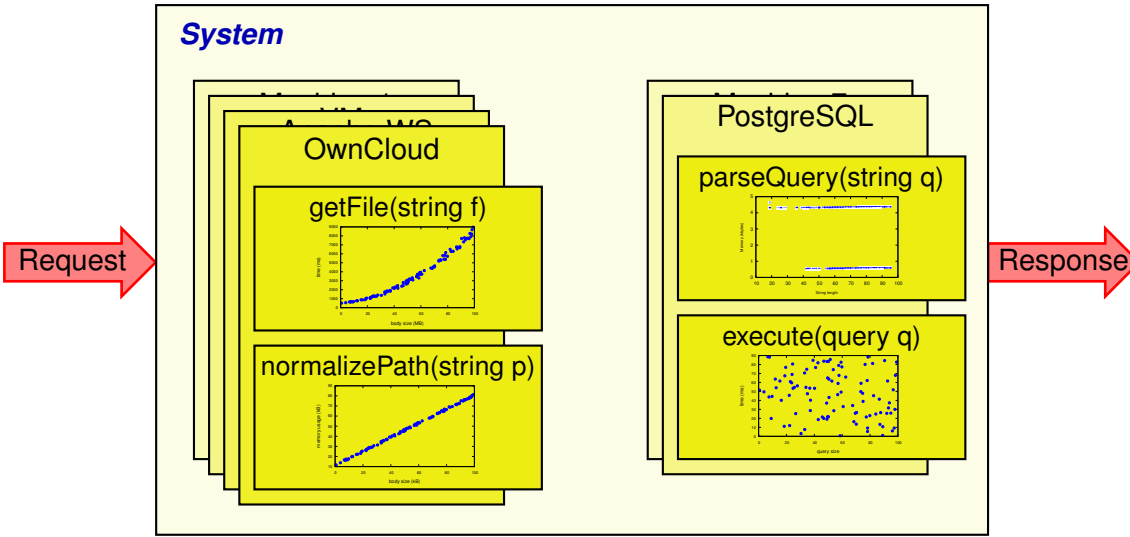
Example: a Web Service

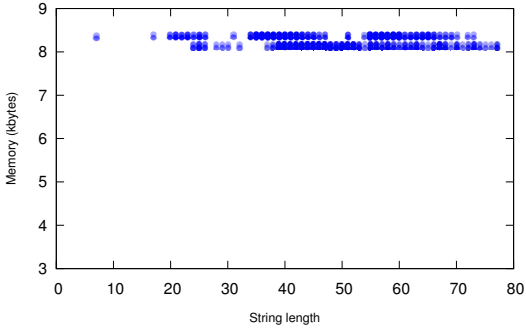


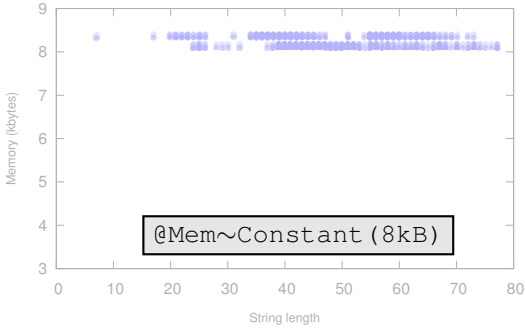
Example: a Web Service

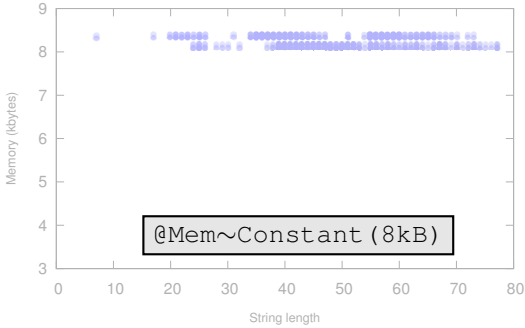


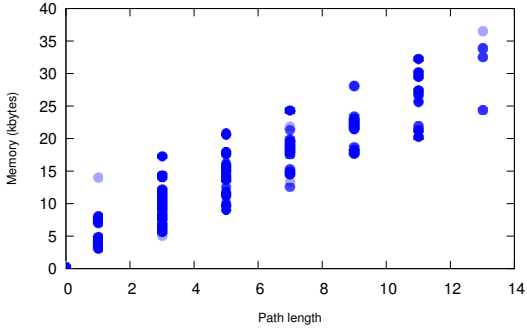
Example: a Web Service

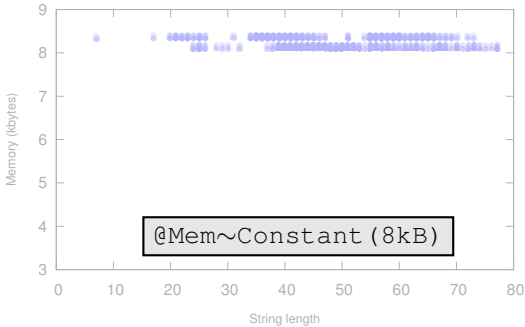


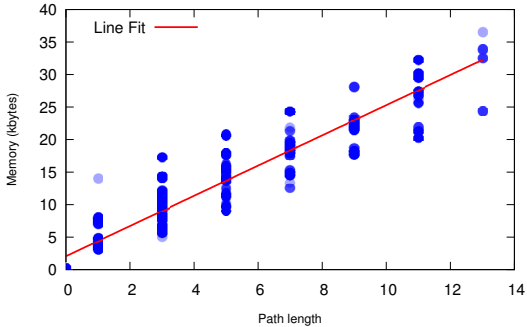

Memory (kbytes)
String length
9
8
7
6
5
4
3
0
10
20
30
40
50
60
70
80

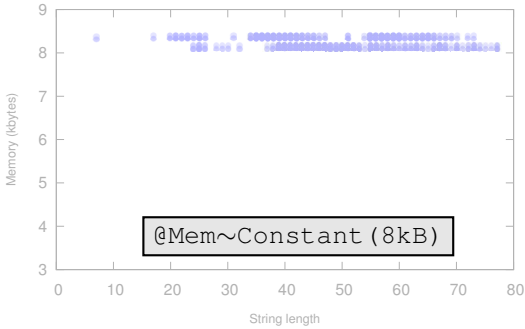

Memory (kbytes)
9
8
7
6
5
4
3
0
10
20
30
40
50
60
70
80
String length
@Mem~Constant(8kB)

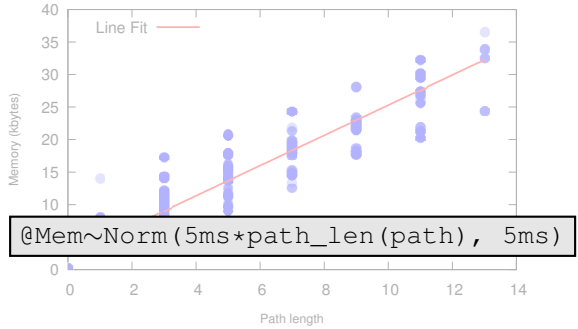

Memory (kbytes)
String length
@Mem~Constant(8kB)

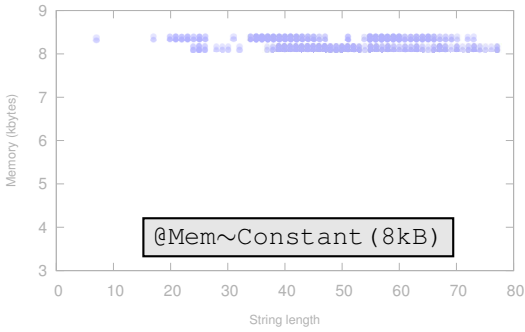

Memory (kbytes)
Path length

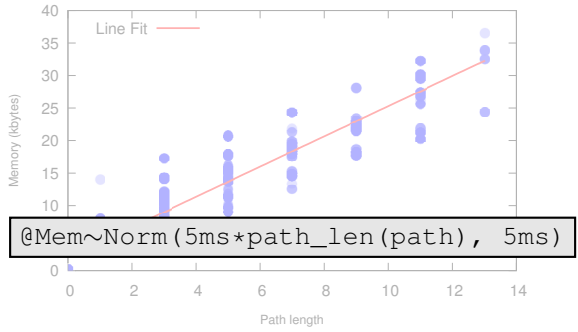

Memory (kbytes)
String length
@Mem~Constant(8kB)

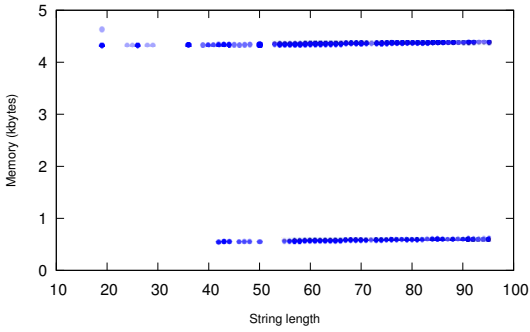

Line Fit
Memory (kbytes)
Path length

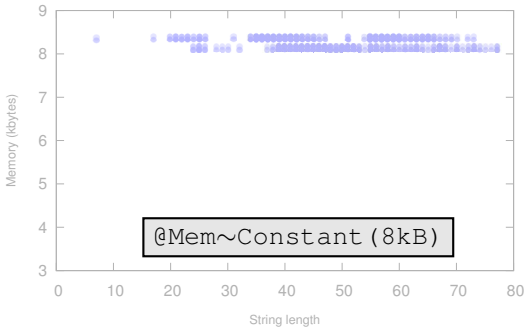

Memory (kbytes)
String length
@Mem~Constant(8kB)

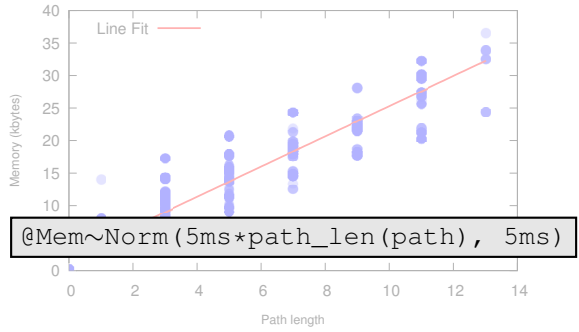

Line Fit
Memory (kbytes)
Path length
@Mem~Norm(5ms*path_len(path), 5ms)

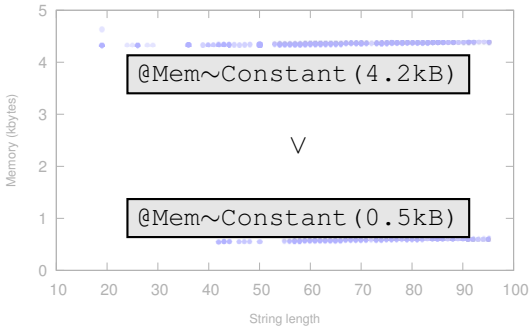

Memory (kbytes)
String length
@Mem~Constant(8kB)

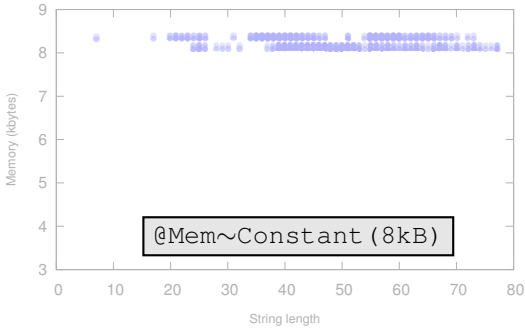

Line Fit
Memory (kbytes)
Path length
@Mem~Norm(5ms*path_len(path), 5ms)

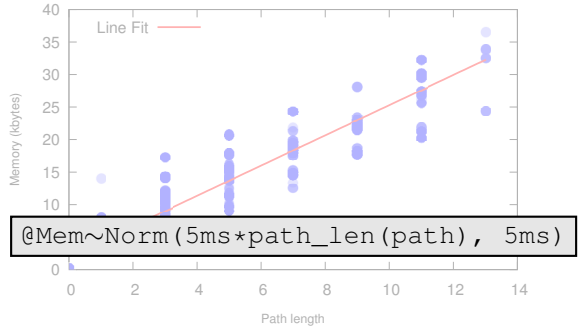

Memory (kbytes)
String length

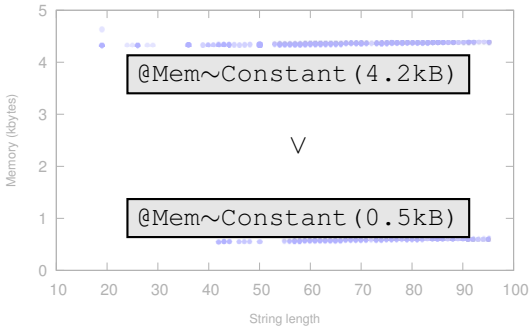

Memory (kbytes)
String length
@Mem~Constant(8kB)

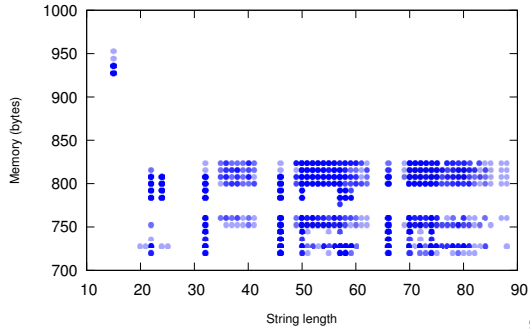

Line Fit
Memory (kbytes)
Path length
@Mem~Norm(5ms*path_len(path), 5ms)

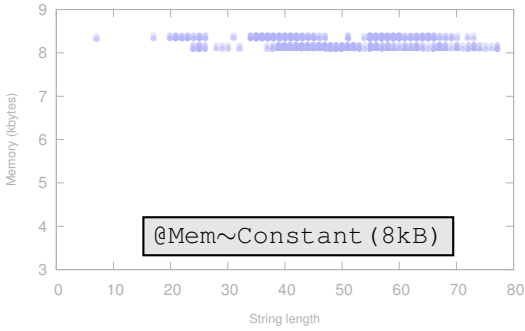

Memory (kbytes)
String length
@Mem~Constant(4.2kB)
\/
@Mem~Constant(0.5kB)

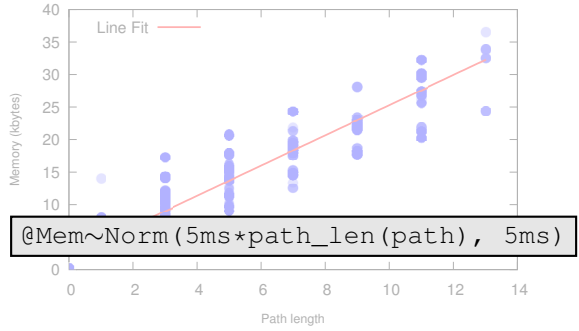

Memory (kbytes)
String length
@Mem~Constant(8kB)

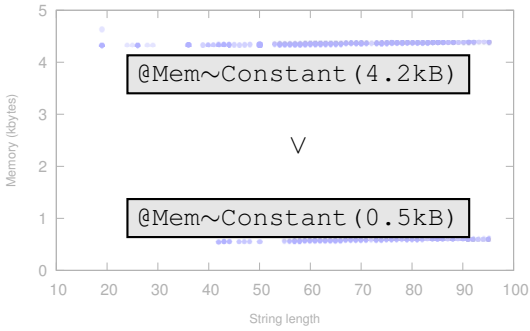

Line Fit
Memory (kbytes)
Path length
@Mem~Norm(5ms*path_len(path), 5ms)

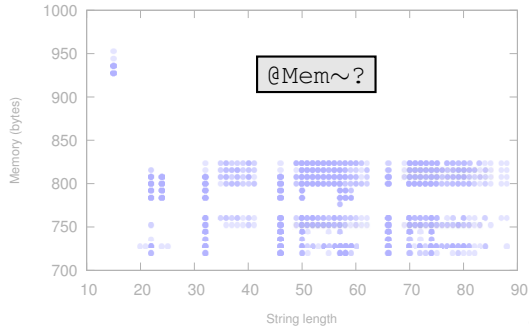

Memory (kbytes)
String length
@Mem~Constant(4.2kB)
V
@Mem~Constant(0.5kB)


Memory (bytes)
String length


Memory (kbytes)
String length
@Mem~Constant(8kB)


Line Fit
Memory (kbytes)
Path length
@Mem~Norm(5ms*path_len(path), 5ms)


Memory (kbytes)
String length
@Mem~Constant(4.2kB)
\/
@Mem~Constant(0.5kB)


Memory (bytes)
String length
@Mem~?

How To Create Annotations

Instrumentation

Instrumentation

For every call of all the functions in the system, we need:

- metrics of interest
  - execution time
  - dynamic memory allocation
  - locks holding time
- relevant features of the input parameters
  - string length
  - collection size

Instrumentation

For every call of all the functions in the system, we need:

- metrics of interest
  - execution time
  - dynamic memory allocation
  - locks holding time
- relevant features of the input parameters
  - string length
  - collection size

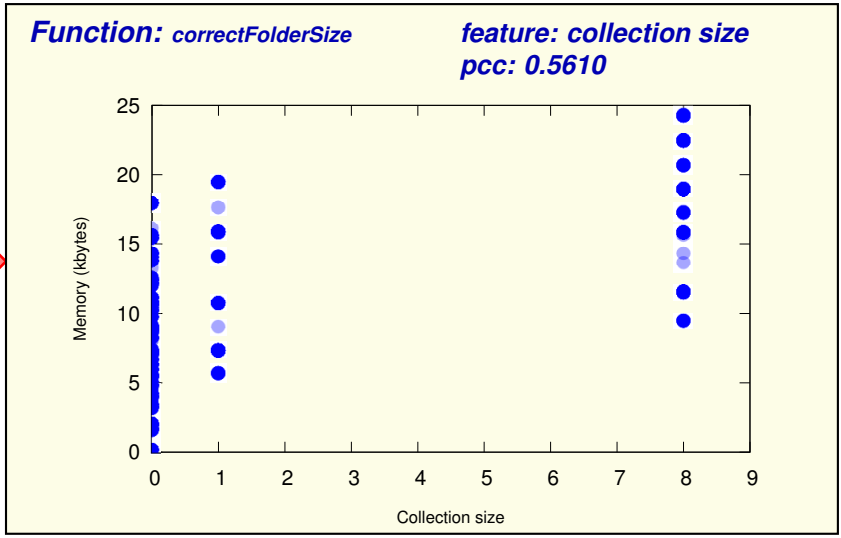
Also, the instrumentation must be:

- crosslayer
- crossplatform

Automatic Annotation Inference

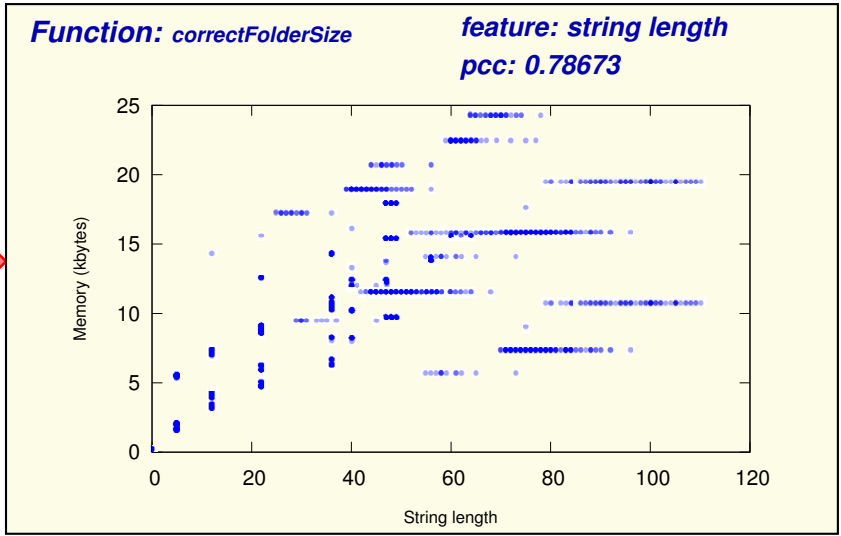
Automatic Annotation Inference

Logs



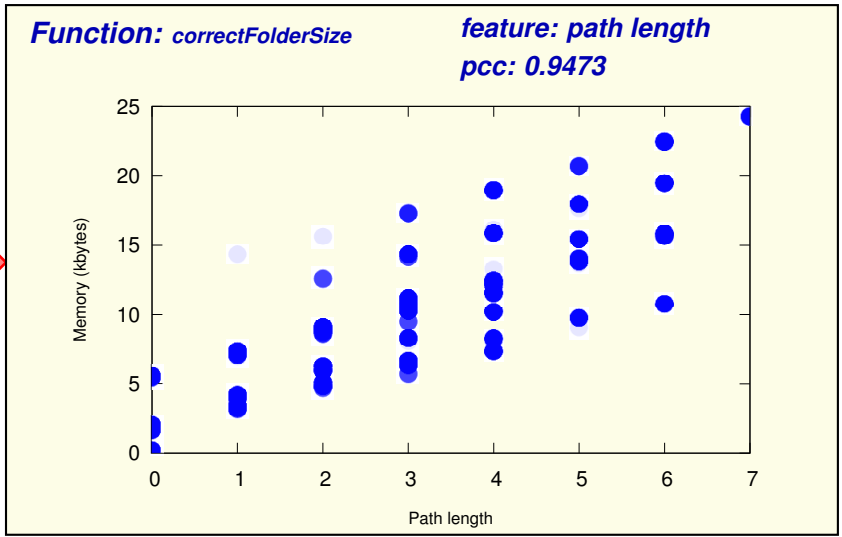
Automatic Annotation Inference

Logs



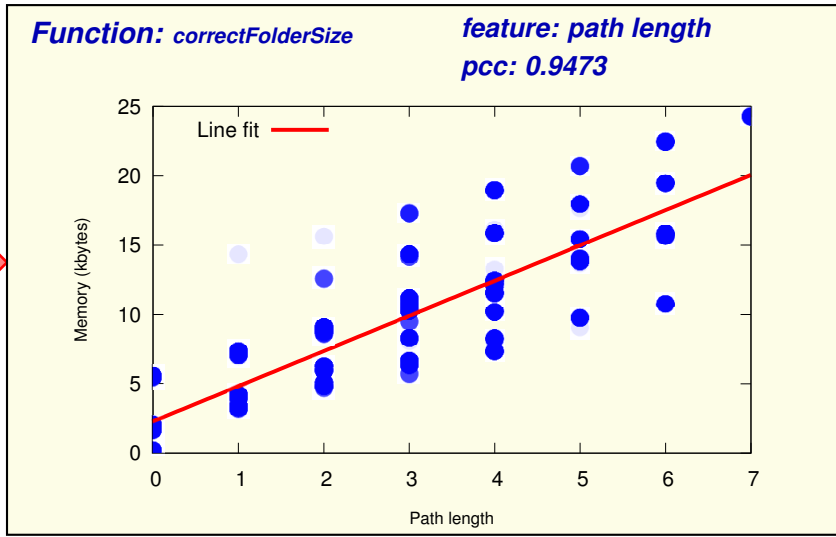
Automatic Annotation Inference

Logs



Automatic Annotation Inference

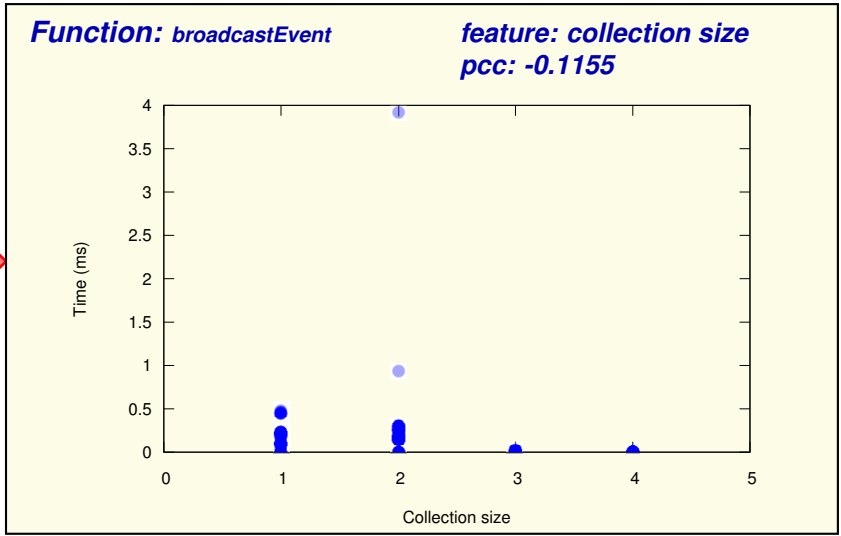
Logs



regression

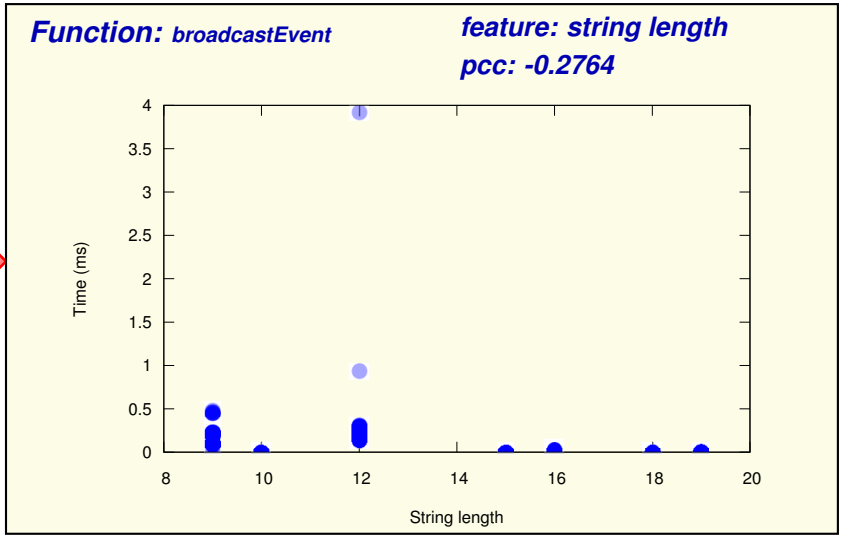
Automatic Annotation Inference

Logs



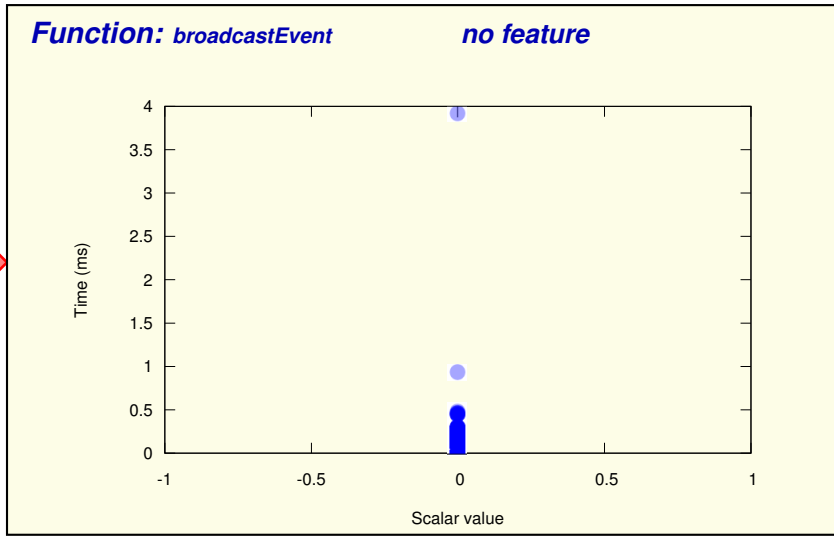
Automatic Annotation Inference

Logs



Automatic Annotation Inference

Logs



clusters

Annotations Uses

Uses

Uses

- Documentation
  - how do functions behave?

Uses

- Documentation
  - how do functions behave?
- Anomaly/failure detection
  - is the system behaving normally? Is there a performance regression?

Uses

- Documentation
  - how do functions behave?
- Anomaly/failure detection
  - is the system behaving normally? Is there a performance regression?
- Extrapolation
  - can we scale up?

Uses

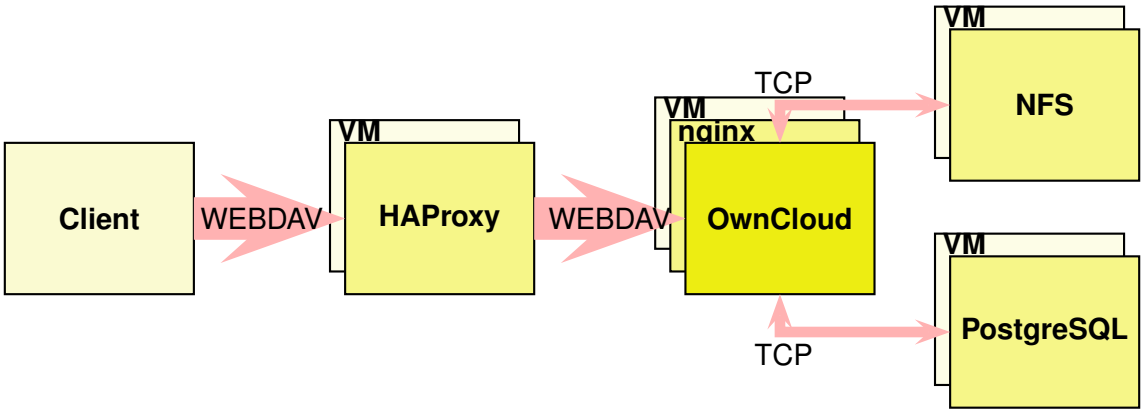
- Documentation
  - how do functions behave?
- Anomaly/failure detection
  - is the system behaving normally? Is there a performance regression?
- Extrapolation
  - can we scale up?
- Composition
  - can we infer the behavior of the caller from the annotations of the callees?

Case Study

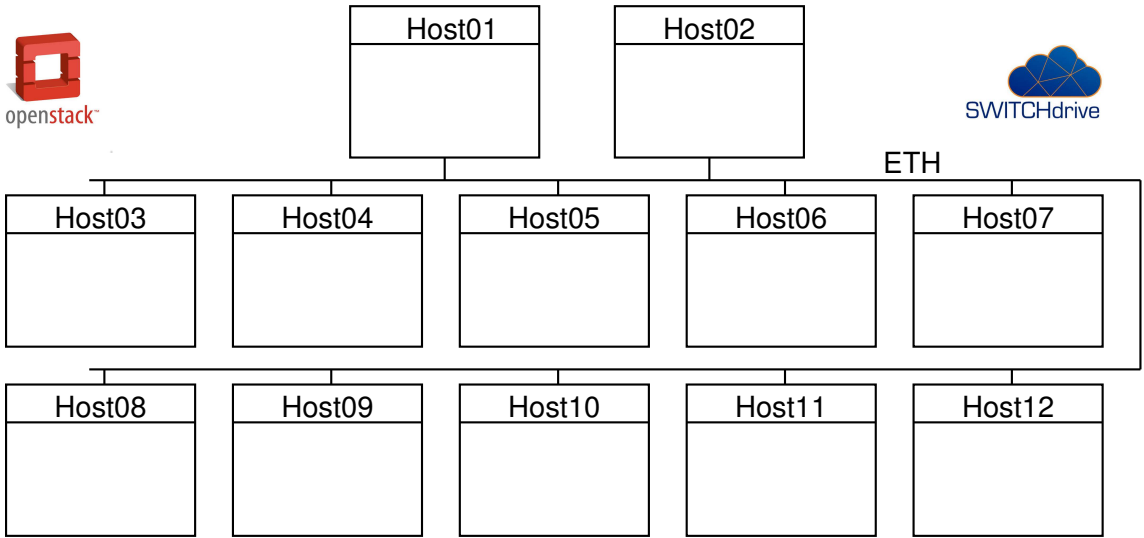
Case Study

setup

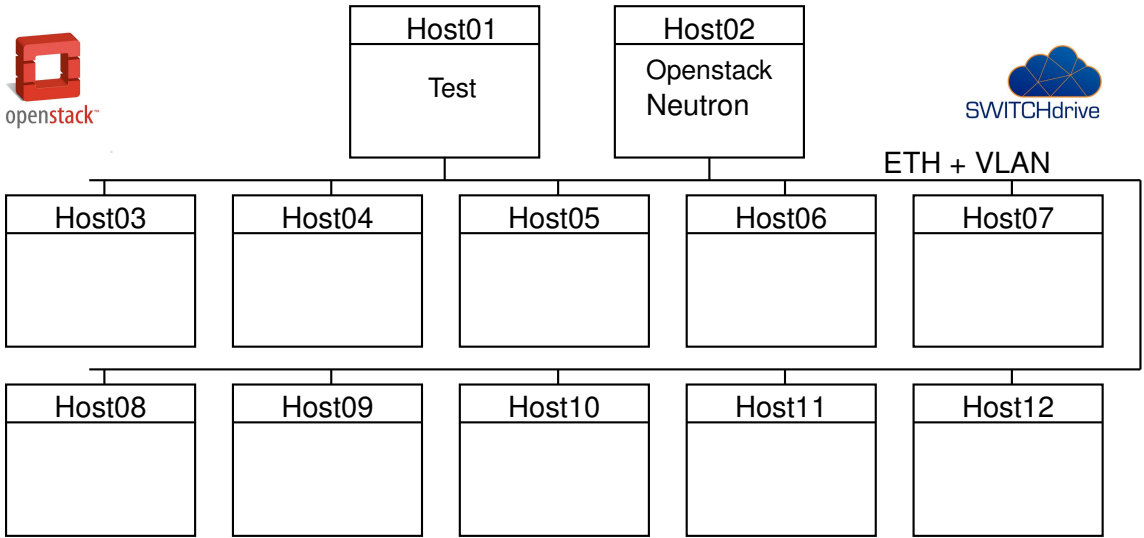
The System: Application Level



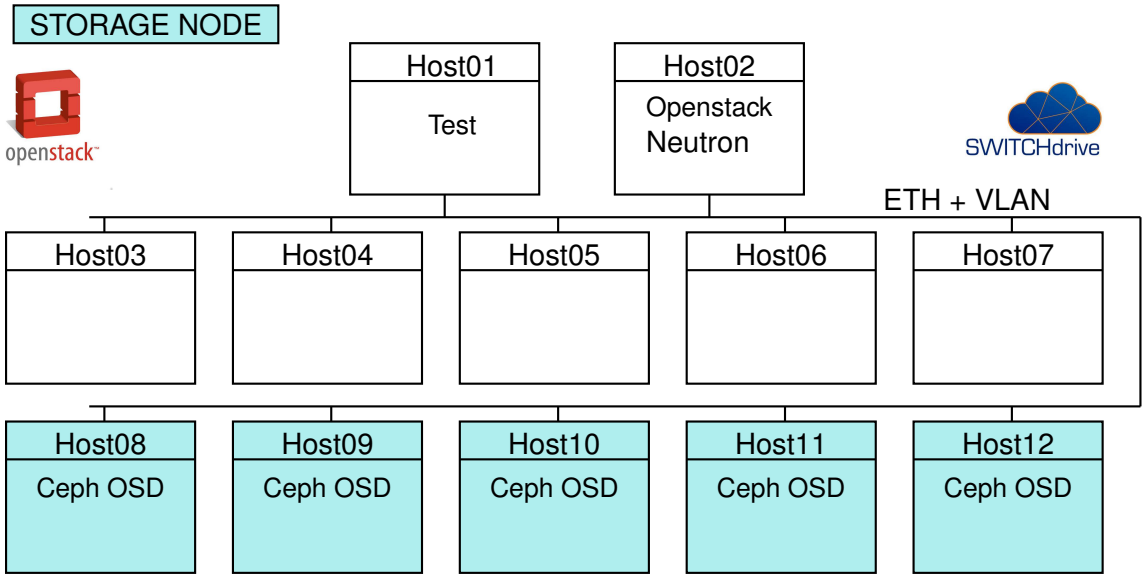
The System: Computing Resources



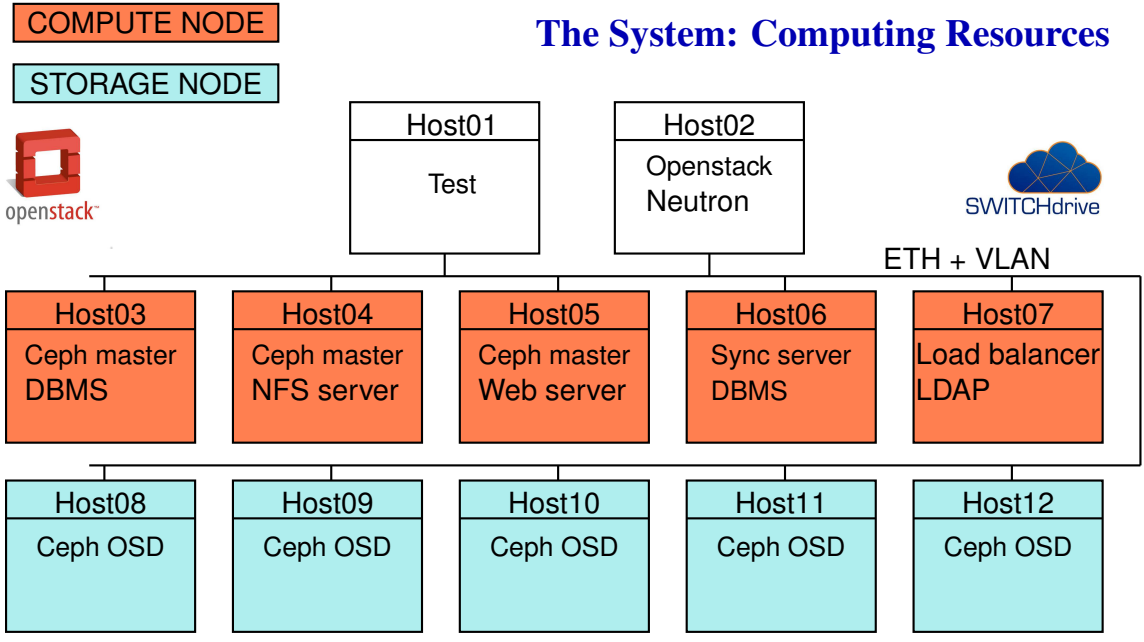
The System: Computing Resources



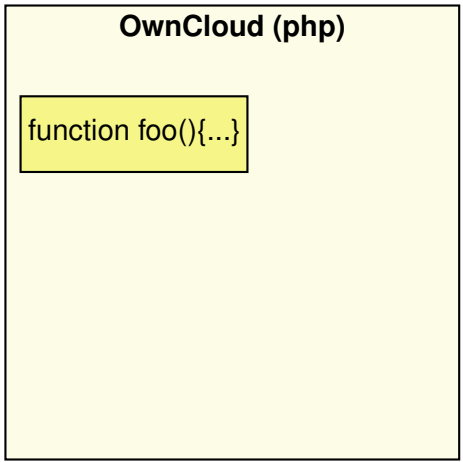
The System: Computing Resources



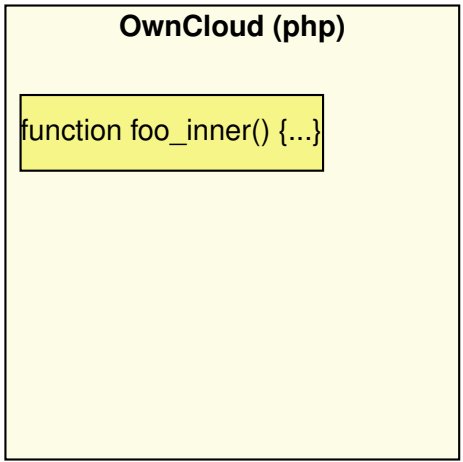
The System: Computing Resources



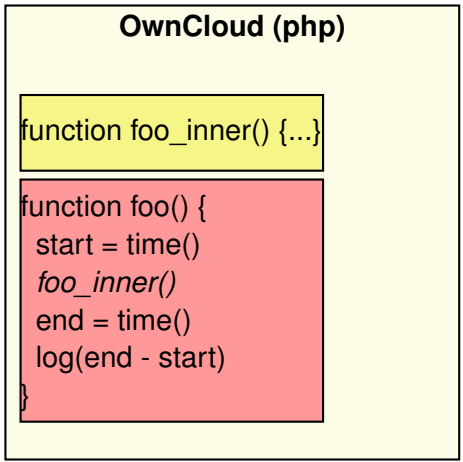
PHP Instrumentation



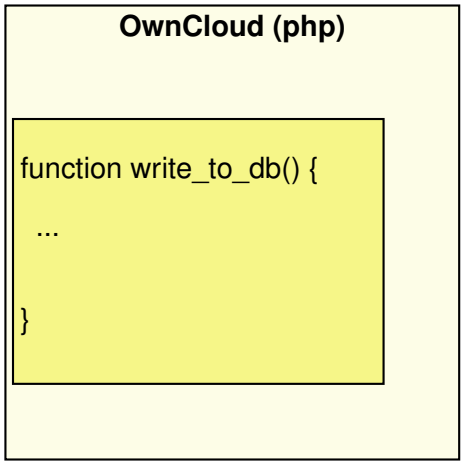
PHP Instrumentation



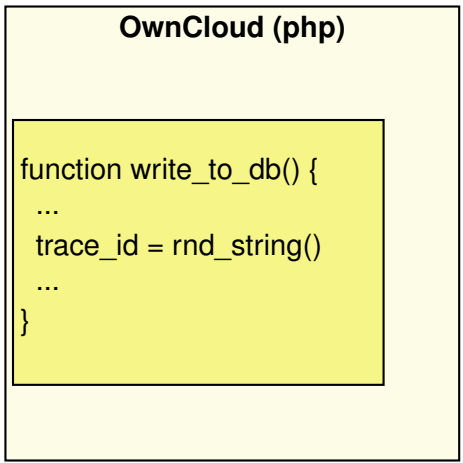
PHP Instrumentation



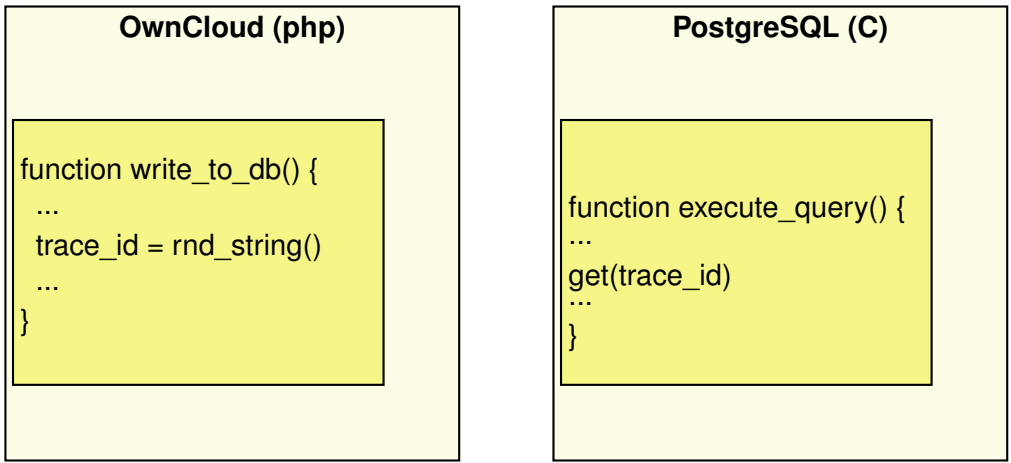
Cross-Platform Instrumentation



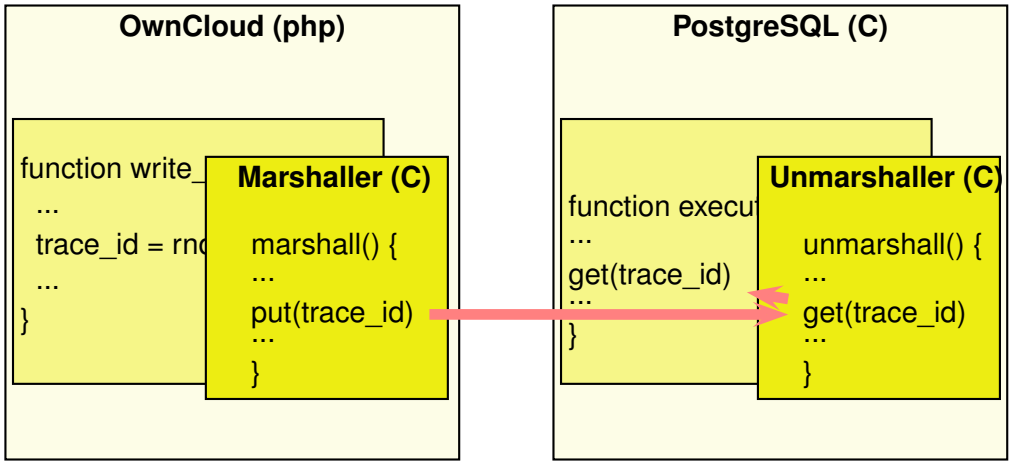
Cross-Platform Instrumentation



Cross-Platform Instrumentation



Cross-Platform Instrumentation

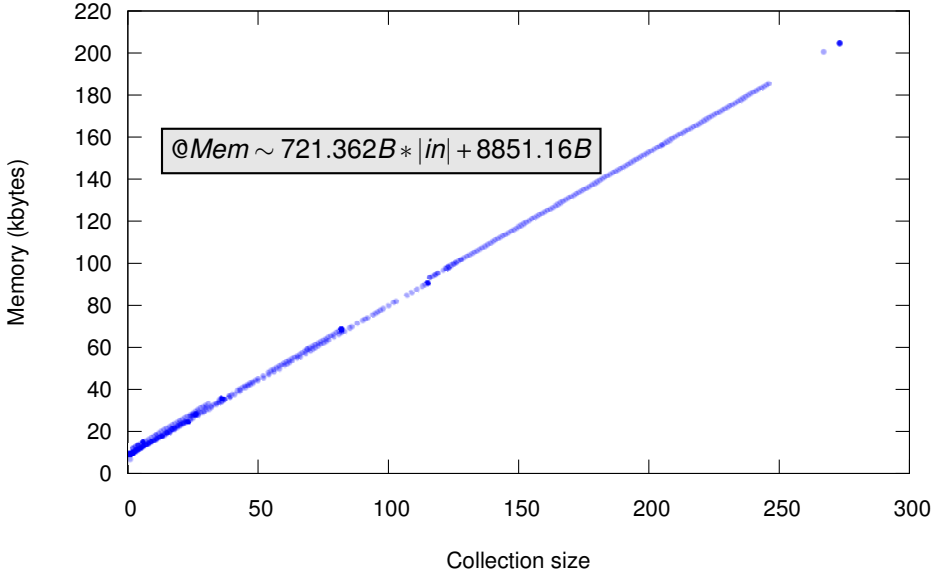


Case Study

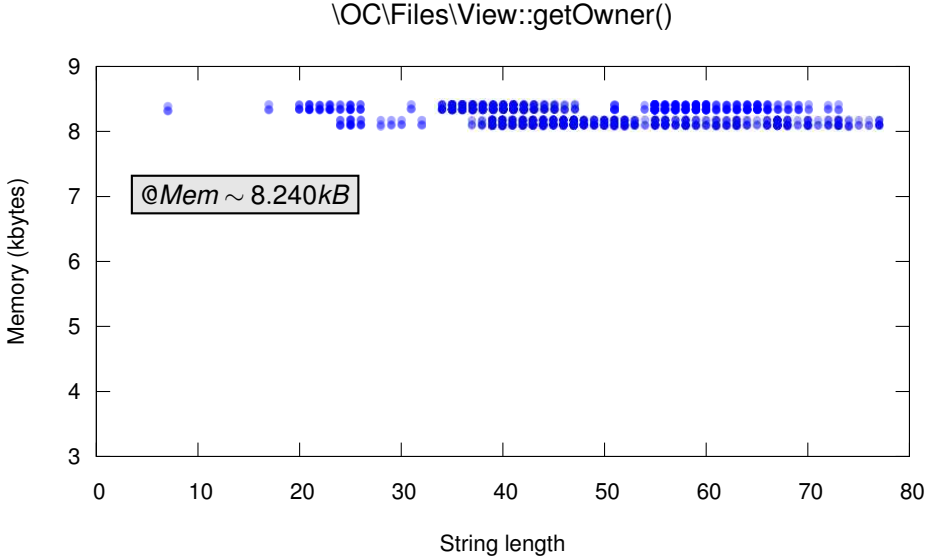
Case Study

annotations

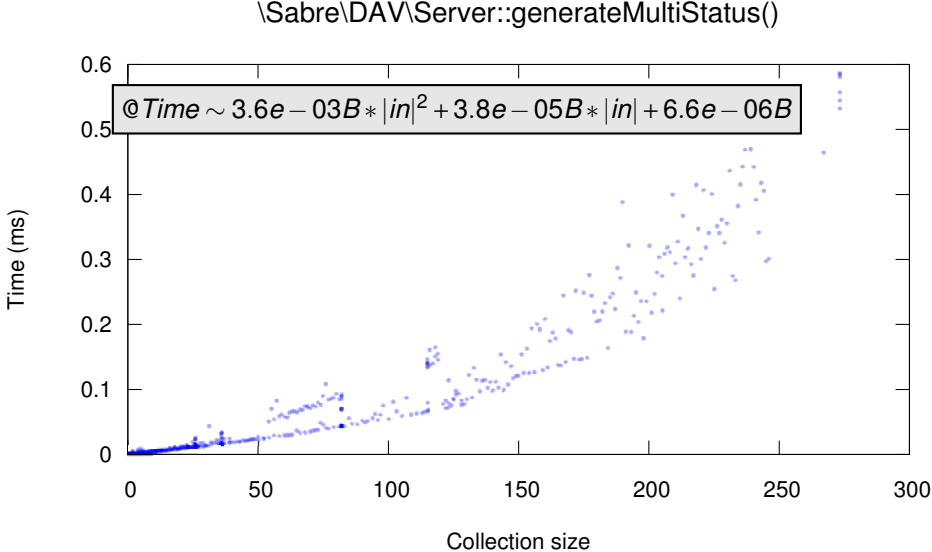
It Works!


@Mem ∼ 721.362B ∗ |in| + 8851.16B
Memory (kbytes)
Collection size
0
20
40
60
80
100
120
140
160
180
200
220
0
50
100
150
200
250
300

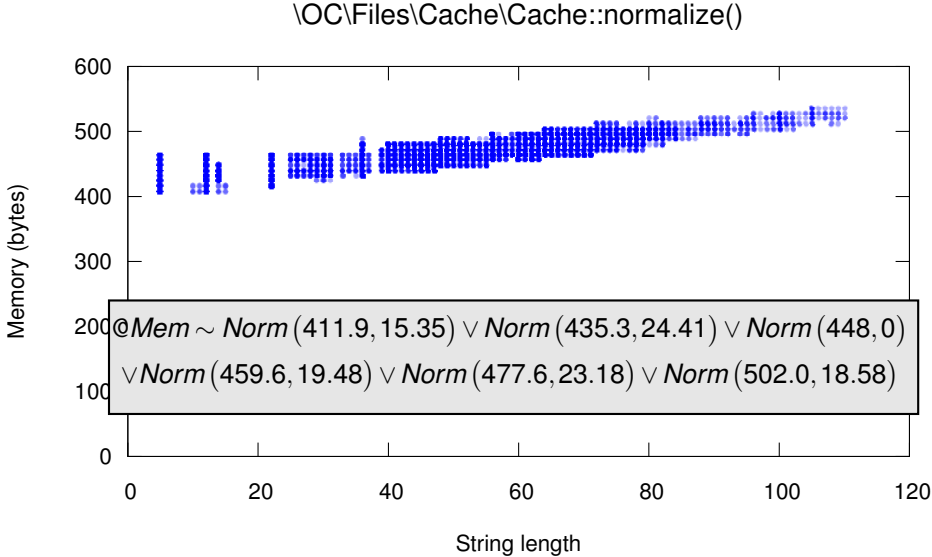
It Works!


\OC\Files\View::getOwner()
@Mem ∼ 8.240kB
Memory (kbytes)
String length
9
8
7
6
5
4
3
0
10
20
30
40
50
60
70
80

It Works!


\Sabre\DAV\Server::generateMultiStatus()
@Time ∼ 3.6e − 03B ∗ |in|^2 + 3.8e − 05B ∗ |in| + 6.6e − 06B
Time (ms)
0.6
0.5
0.4
0.3
0.2
0.1
0
0
50
100
150
200
250
300
Collection size

It Works!

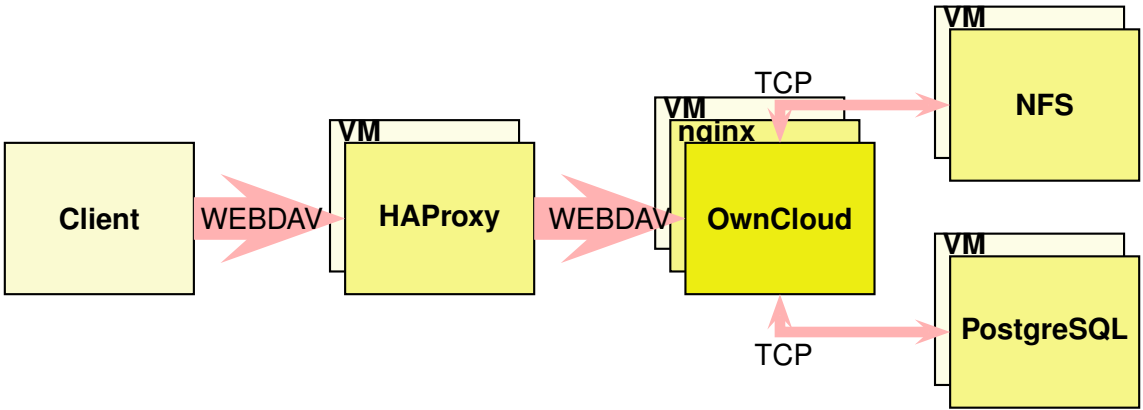

\OC\Files\Cache\Cache::normalize()
Memory (bytes)
String length
0
100
200
300
400
500
600
0
20
40
60
80
100
120
@Mem ∼ Norm (411.9, 15.35) ∨ Norm (435.3, 24.41) ∨ Norm (448, 0)
∨ Norm (459.6, 19.48) ∨ Norm (477.6, 23.18) ∨ Norm (502.0, 18.58)

Case Study

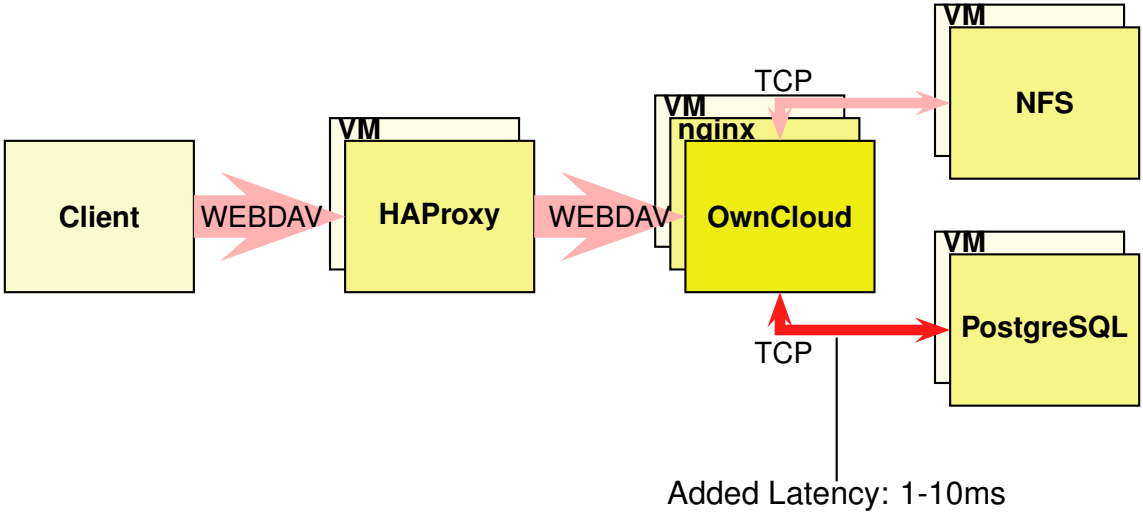
Case Study

anomaly detection

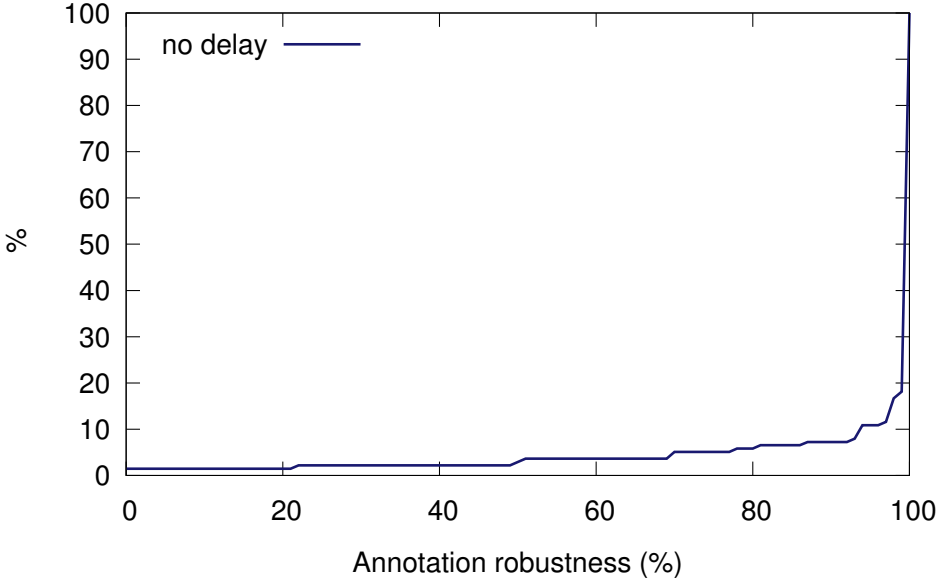
Anomaly Injection



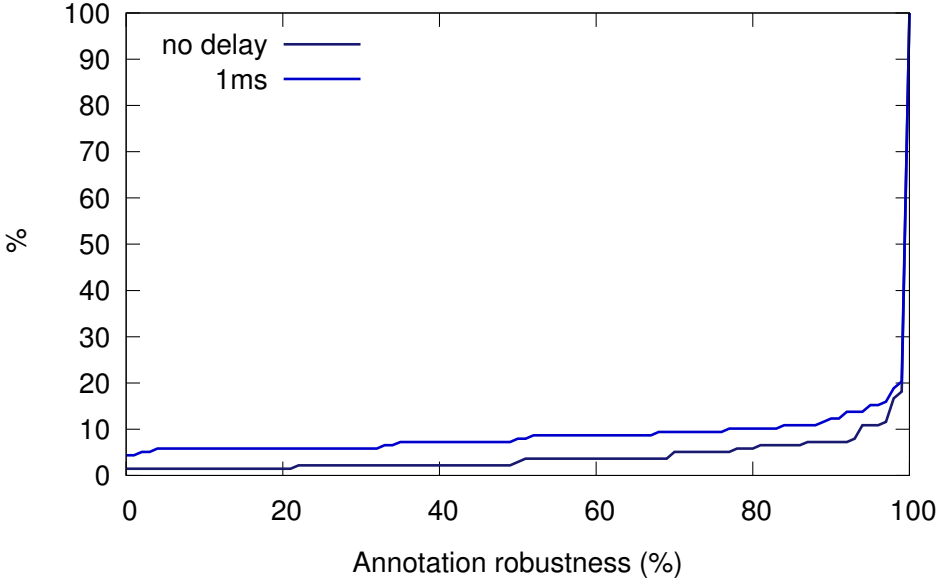
Anomaly Injection



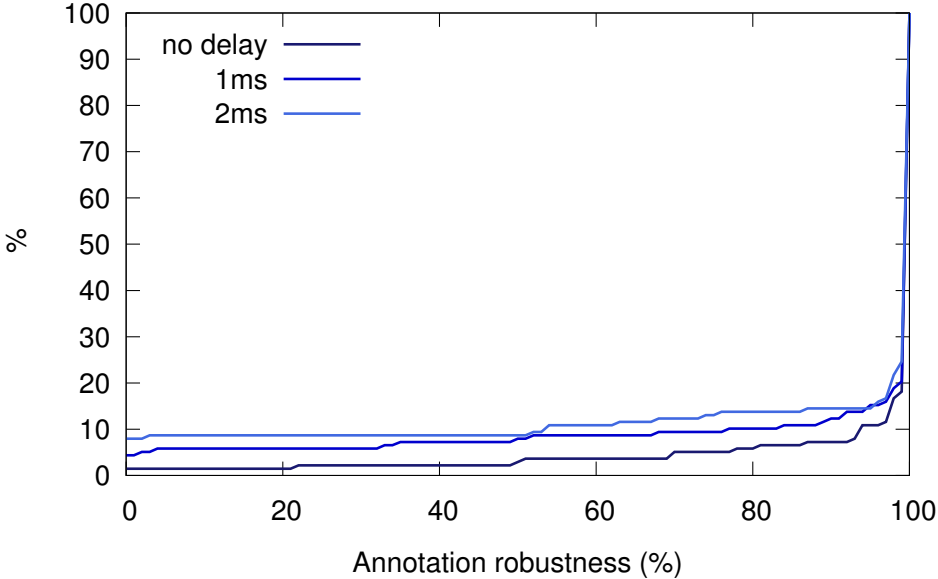
Annotations Catch The Anomaly



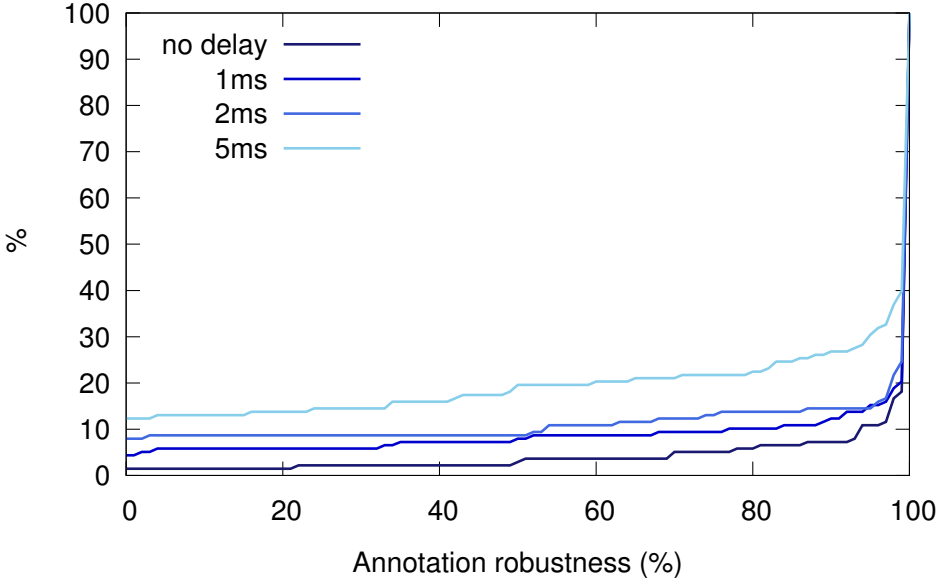
Annotations Catch The Anomaly



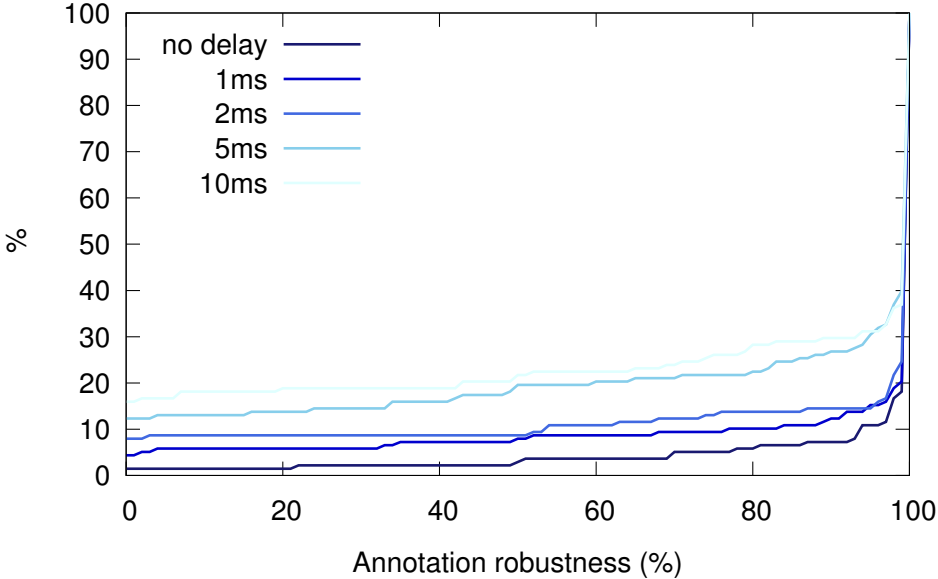
Annotations Catch The Anomaly



Annotations Catch The Anomaly



Annotations Catch The Anomaly



Future

Ongoing Work - Discussion

- Machine learning techniques fine tuning

Ongoing Work - Discussion

- Machine learning techniques fine tuning
- Annotations composition
  - stack analysis

Ongoing Work - Discussion

- Machine learning techniques fine tuning
- Annotations composition
  - stack analysis
- Workload creation
  - can we forge workloads that expose specific behaviors?

Ongoing Work - Discussion

- Machine learning techniques fine tuning
- Annotations composition
  - stack analysis
- Workload creation
  - can we forge workloads that expose specific behaviors?
- Feature Selection
  - Is a heuristically built set of basic features enough?
  - Can we exploit programmers' knowledge of the system?

Ongoing Work - Discussion

- Machine learning techniques fine tuning
- Annotations composition
  - stack analysis
- Workload creation
  - can we forge workloads that expose specific behaviors?
- Feature Selection
  - Is a heuristically built set of basic features enough?
  - Can we exploit programmers' knowledge of the system?
- Extensive testing
  - Java, DaCapo benchmarks

Performance Annotations for Cloud Computing

Daniele Rogora[*] Steffen Smolka[%] Antonio Carzaniga[*] Amer Diwan[$]
Robert Soulé[*~]

presented by
Daniele Rogora

[*]Università della Svizzera italiana [%]Cornell University [$]Google [~]Barefoot Networks

HotCloud 2017