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

Switch 1

Machine 1
Machine 2
Machine 3
Machine 4
Machine 5

Switch 2

CPU  HDD  NET

Machine 6
Machine 7
Machine 8
Machine 9
Machine 10

Switch 3

Machine 11
Machine 12
Machine 13
Machine 14
Machine 15
... But They Are Complex
... But They Are Complex
... But They Are Complex

Virtual Machines

Virtual Network

Operating System

CPU   HDD   NET

Machine 1
Machine 2
Machine 3
Machine 4
Machine 5

Switch 1

Machine 6
Machine 7

Switch 2

Machine 8
Machine 9
Machine 10

Switch 3

Machine 11
Machine 12
Machine 13
Machine 14
Machine 15

Operating System

CPU   HDD   NET

Virtual Machines

Virtual Network
... But They Are Complex

Software

Virtual Machines

Virtual Network

Operating System

CPU  HDD  NET

Machine 1

Machine 2

Machine 3

Machine 4

Machine 5

Switch 1

Switch 2

Switch 3

Machine 6

Machine 7

Machine 8

Machine 9

Machine 10

Machine 11

Machine 12

Machine 13

Machine 14

Machine 15

CPU

HDD

NET

Operating System

Virtual Machines

Virtual Network

Software

Switch 1

Switch 2

Switch 3

Machine 1

Machine 2

Machine 3

Machine 4

Machine 5

Machine 6

Machine 7

Machine 8

Machine 9

Machine 10

Machine 11

Machine 12

Machine 13

Machine 14

Machine 15
... 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 \textit{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

How is it performing now?

Query

Logs → Model → Performance
Goal: Creating a New Model

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

- comprehensive
- live
- interactive

What if we improve the hardware?

Query

Logs ➔ Model ➔ Predictive result
Goal: Creating a New Model

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

- comprehensive
- live
- interactive
PerformanceAnnotations
The response time grows quadratically with the size of the body of the request.
The response time grows quadratically with the size of the body of the request.
The response time grows quadratically with the size of the body of the request.

\[ Time \sim a |in|^2 + b |in| + c \]
Example: a Web Service

System

The memory used grows linearly with the size of the body of the request.

\[ \mathbf{Mem} \sim a \cdot \mid \text{in} \mid + b \]
Example: a Web Service

System

Machine 1

Machine 7

Request → Response
Example: a Web Service

System

VM

PostgreSQL

Request

Response
Example: a Web Service

System

- Machine 1
- Machine 7
- VM

Apache WS

PostgreSQL

Request → Response
Example: a Web Service

System

Request

Response

Machine 1

Machine 7

VM

Apache WS

OwnCloud

PostgreSQL
Example: a Web Service
Example: a Web Service

System

Apache WS

OwnCloud

PostgreSQL

Request

Response

getFile(string f)

normalizePath(string p)

parseQuery(string q)

execute(query q)
Memory (kbytes) vs. String length

@\text{Mem} \sim \text{Constant (8kB)}
@Mem \sim \text{Constant (8kB)}
@Mem~Constant (8kB)

@Mem~Norm (5ms*path_len(path), 5ms)
@Mem \sim \text{Constant} (8kB)

@Mem \sim \text{Norm}(5ms \times \text{path\_len(path)}, 5ms)
@Mem~Constant (8kB)

@Mem~Norm (5ms*path_len(path), 5ms)

@Mem~Constant (4.2kB)

V

@Mem~Constant (0.5kB)
\[ @\text{Mem} \sim \text{Constant}(8\text{kB}) \]

\[ @\text{Mem} \sim \text{Norm}(5\text{ms} \times \text{path\_len}(\text{path}), 5\text{ms}) \]

\[ @\text{Mem} \sim \text{Constant}(4.2\text{kB}) \]

\[ \lor \]

\[ @\text{Mem} \sim \text{Constant}(0.5\text{kB}) \]
@Mem~Constant (8kB)

@Mem~Norm (5ms*\text{path\_len(path)}, 5ms)

@Mem~Constant (4.2kB)

\lor

@Mem~Constant (0.5kB)

@Mem~?
How To Create Annotations
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
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

Function: `correctFolderSize`

Feature: collection size

$pcc: 0.5610$

Memory (kbytes) vs. Collection size

Logs
Automatic Annotation Inference

**Function:** `correctFolderSize`  
**feature:** string length  
**pcc:** 0.78673

![Graph showing the relationship between string length and memory usage with a correlation coefficient of 0.78673.](image)
Automatic Annotation Inference

Function: \textit{correctFolderSize}  
feature: path length  
pcc: 0.9473

![Graph showing the relationship between memory (kbytes) and path length with points scattered along a 2D axis and a line of best fit]

Logs
Automatic Annotation Inference

Function: `correctFolderSize`

feature: `path length`

pcc: 0.9473

Logs

regression
Automatic Annotation Inference

**Function:** `broadcastEvent`

**Feature:** `collection size`

**PCC:** -0.1155
Automatic Annotation Inference

**Function:** broadcastEvent

**Feature:** string length

**pcc:** -0.2764
Automatic Annotation Inference

Function: broadcastEvent  no feature

Time (ms) vs Scalar value

Logs → clusters
Annotations Uses
Documentation

- how do functions behave?
- 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

setup
The System: Application Level

Client ➔ WEBDAV ➔ HAProxy ➔ WEBDAV ➔ OwnCloud ➔ PostgreSQL ➔ NFS

TCP connections between components.
The System: Computing Resources
The System: Computing Resources

Host01
Test

Host02
Openstack
Neutron

ETH + VLAN

Host03

Host04

Host05

Host06

Host07

Host08

Host09

Host10

Host11

Host12
The System: Computing Resources
OwnCloud (php)

function foo(){...}
OwnCloud (php)

function foo_inner() {...}
function foo_inner() {...}

function foo() {
    start = time()
    foo_inner()
    end = time()
    log(end - start)
}
function write_to_db() {
    ...
}

OwnCloud (php)
function write_to_db() {
    ...
    trace_id = rnd_string()
    ...
}
Cross-Platform Instrumentation

OwnCloud (php)

function write_to_db() {
  ...
  trace_id = rnd_string()
  ...
}

PostgreSQL (C)

function execute_query() {
  ...
  get(trace_id)
  ...
}
Cross-Platform Instrumentation

OwnCloud (php)

```php
function write_to_db() {
    ...
    trace_id = rnd_string()
    ...
}
```

PostgreSQL (C)

```c
function execute_query() {
    ...
    get(trace_id)
    ...
}
```

Marshaller (C)

```
marshall() {
    ...
    put(trace_id)
    ...
}
```

Unmarshaller (C)

```
unmarshall() {
    ...
    get(trace_id)
    ...
}
```
Case Study
Case Study

annotations
$\hat{\text{Mem}} \sim 721.362B \cdot |in| + 8851.16B$
It Works!

\OC\Files\View::getOwner()

@Mem \sim 8.240\text{kB}
\texttt{\textbackslash Sabre\DAV\Server::generateMultiStatus()}

$$\text{Time} \approx 3.6 \times 10^{-3} B \times |in|^2 + 3.8 \times 10^{-5} B \times |in| + 6.6 \times 10^{-6} B$$
It Works!

\texttt{\OC\Files\Cache\Cache::normalize()}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{chart.png}
\caption{Memory usage vs. string length}
\end{figure}

$\text{Mem} \sim \text{Norm}(411.9, 15.35) \lor \text{Norm}(435.3, 24.41) \lor \text{Norm}(448, 0) \lor \text{Norm}(459.6, 19.48) \lor \text{Norm}(477.6, 23.18) \lor \text{Norm}(502.0, 18.58)$
Case Study
Case Study

anomaly detection
Anomaly Injection

Client ➔ WEBDAV ➔ HAProxy ➔ WEBDAV ➔ OwnCloud ➔ PostgreSQL ➔ TCP

Added Latency: 1-10ms
Annotations Catch The Anomaly

Graph showing annotation robustness for 'no delay'.
Annotations Catch The Anomaly

![Graph showing the robustness of annotations with and without delay. The graph compares the robustness of annotations with no delay and 1ms delay.]
Annotations Catch The Anomaly

![Graph showing annotation robustness with different delays](image-url)
- 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