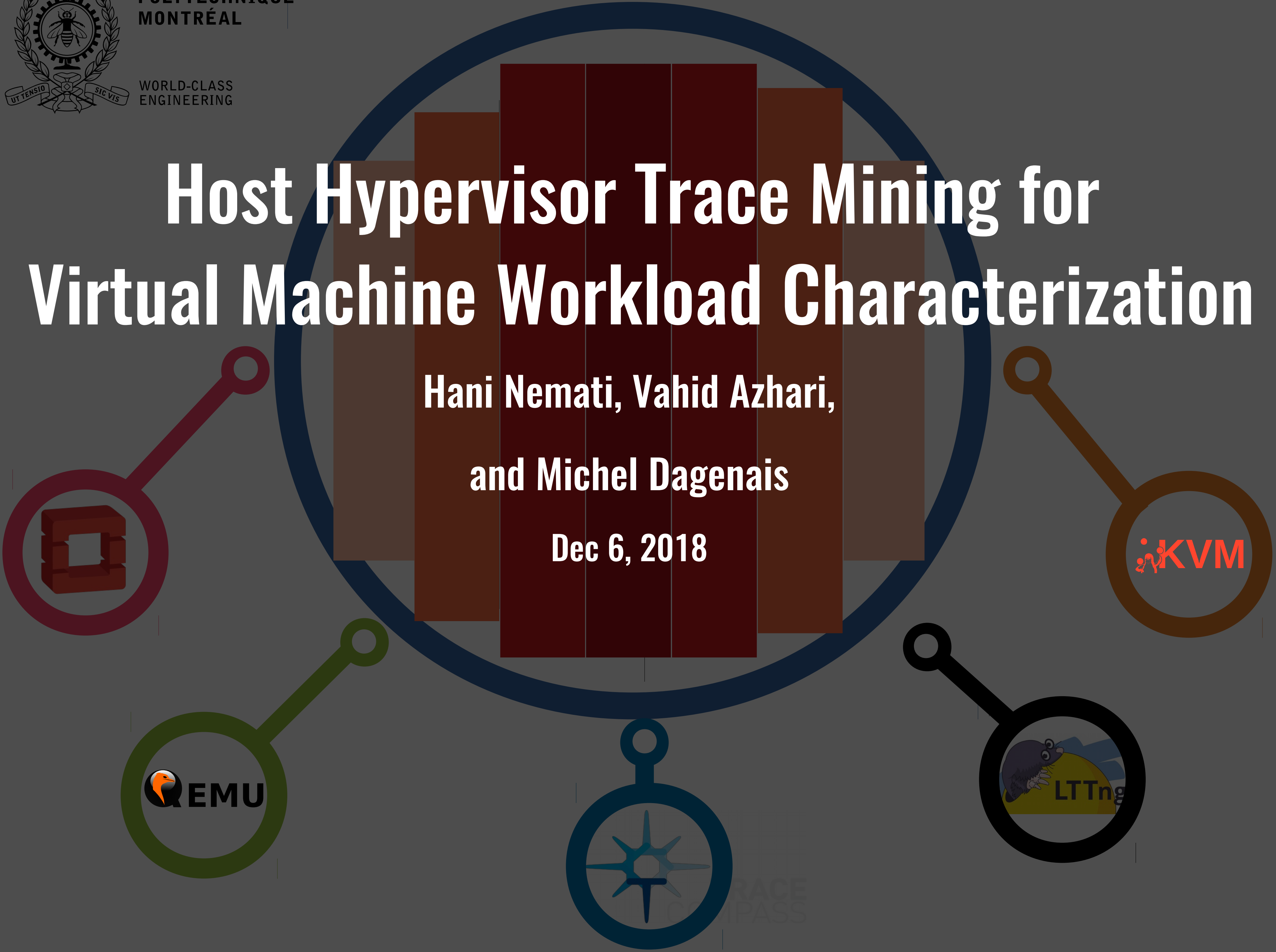
POLYTECHNIQUE
MONTRÉAL

WORLD-CLASS
ENGINEERING



Host Hypervisor Trace Mining for Virtual Machine Workload Characterization

Hani Nemati, Vahid Azhari,

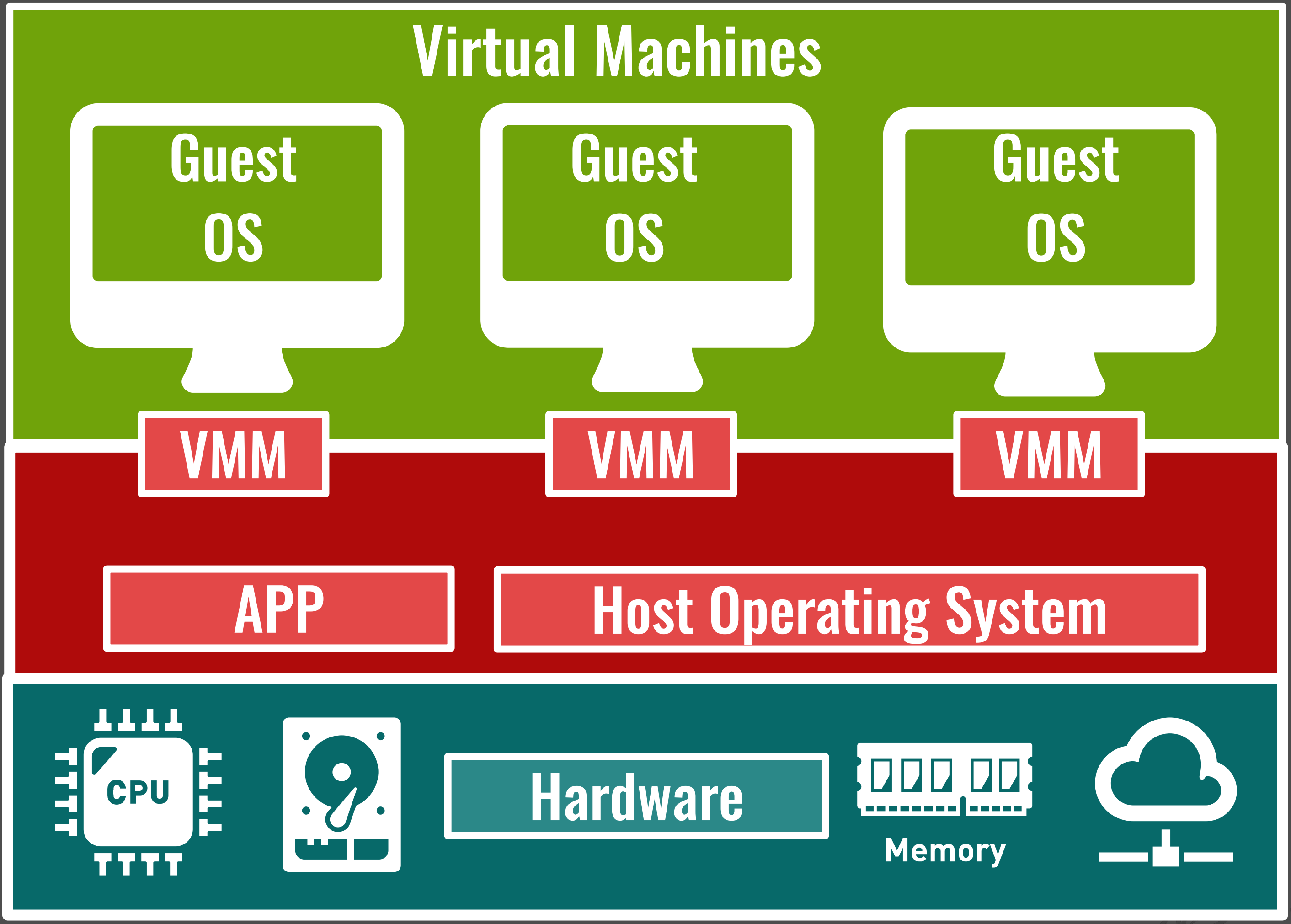
and Michel Dagenais

Dec 6, 2018

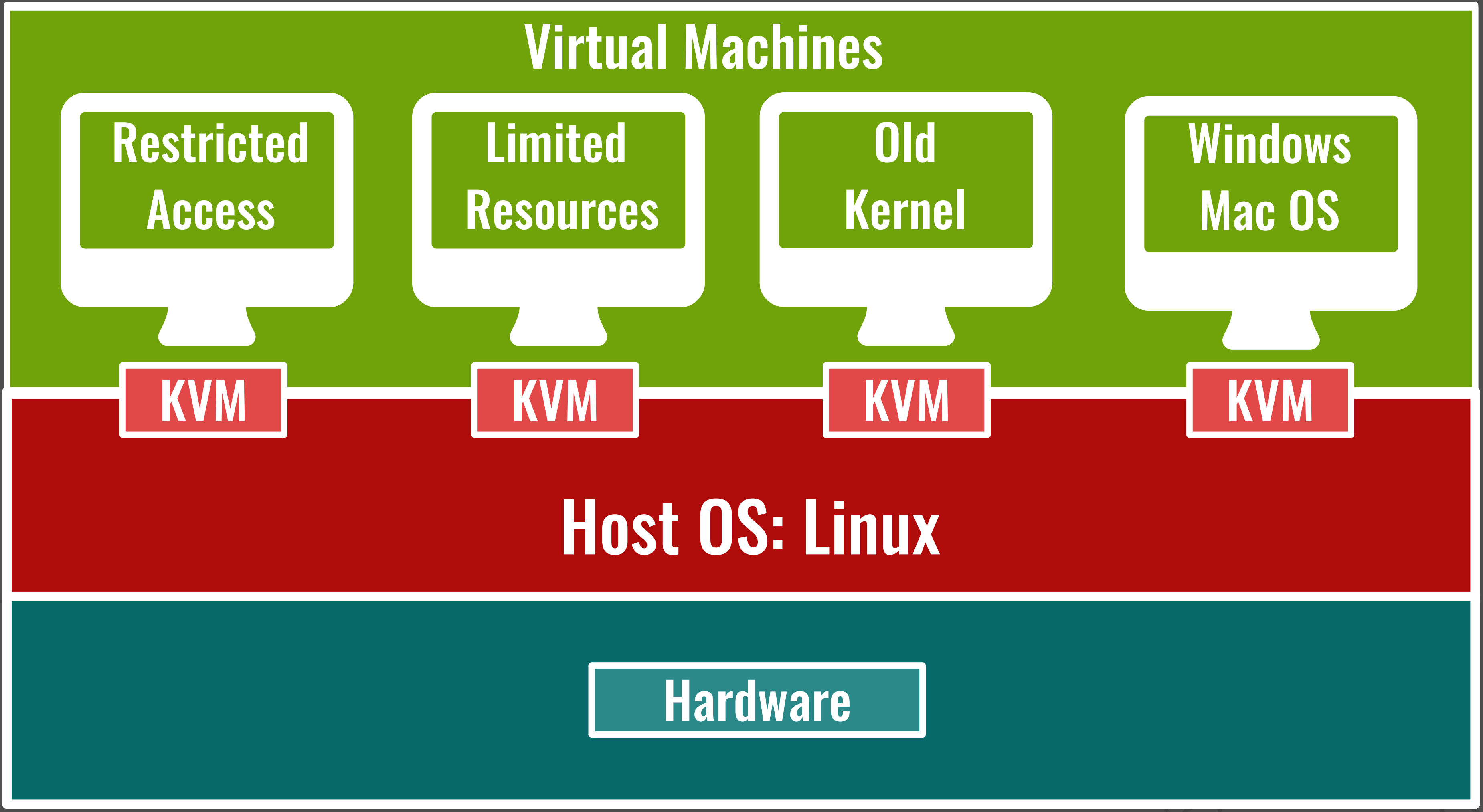
Outline

1	2	3	4	5
Introduction	**Previously on VM Analysis**	**Feature Extraction**	**VM Clustering**	**Conclusion & Future Work**
Agent-less	New Tracepoint	Virtual Interrupt	K-MEANS	Trace mining
Restricted Access	Nested VM	CPU Metrics	CPU Intensive	VM Clustering
Old Kernel	Wait Analysis	Disk Metrics	Disk Intensive	vCPU features
Limited Resources	Critical Path	Network Metrics	Network Intensive	Process Clustering

VM Architecture



Motivation


Virtual Machines
Restricted Access
Limited Resources
Old Kernel
Windows Mac OS
KVM
KVM
KVM
KVM
Host OS: Linux
Hardware

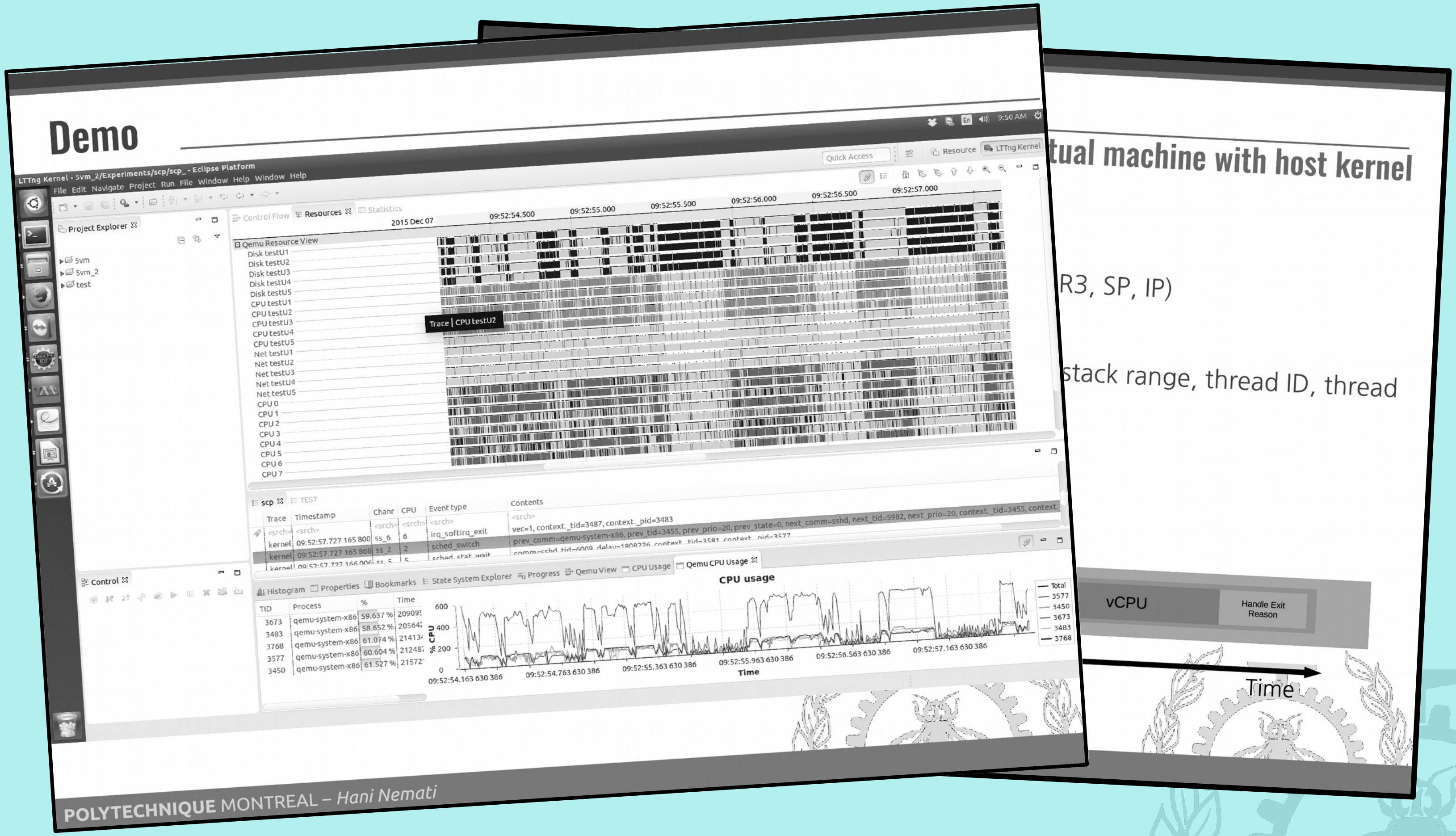
Previously on VM Analysis

10 Dec 2015

A

Step 01-Monitoring

- Fine-grain resource Monitoring
  Retrieve guest state from VMCS
  Analyzing VM interference
  Disk View, vCPU view , Netrowk view



TID	Process	%
3673	qemu-system-x86	59.637 %
3483	qemu-system-x86	58.652 %
3768	qemu-system-x86	61.074 %
3577	qemu-system-x86	60.604 %
3450	qemu-system-x86	61.527 %

Previously on VM Analysis

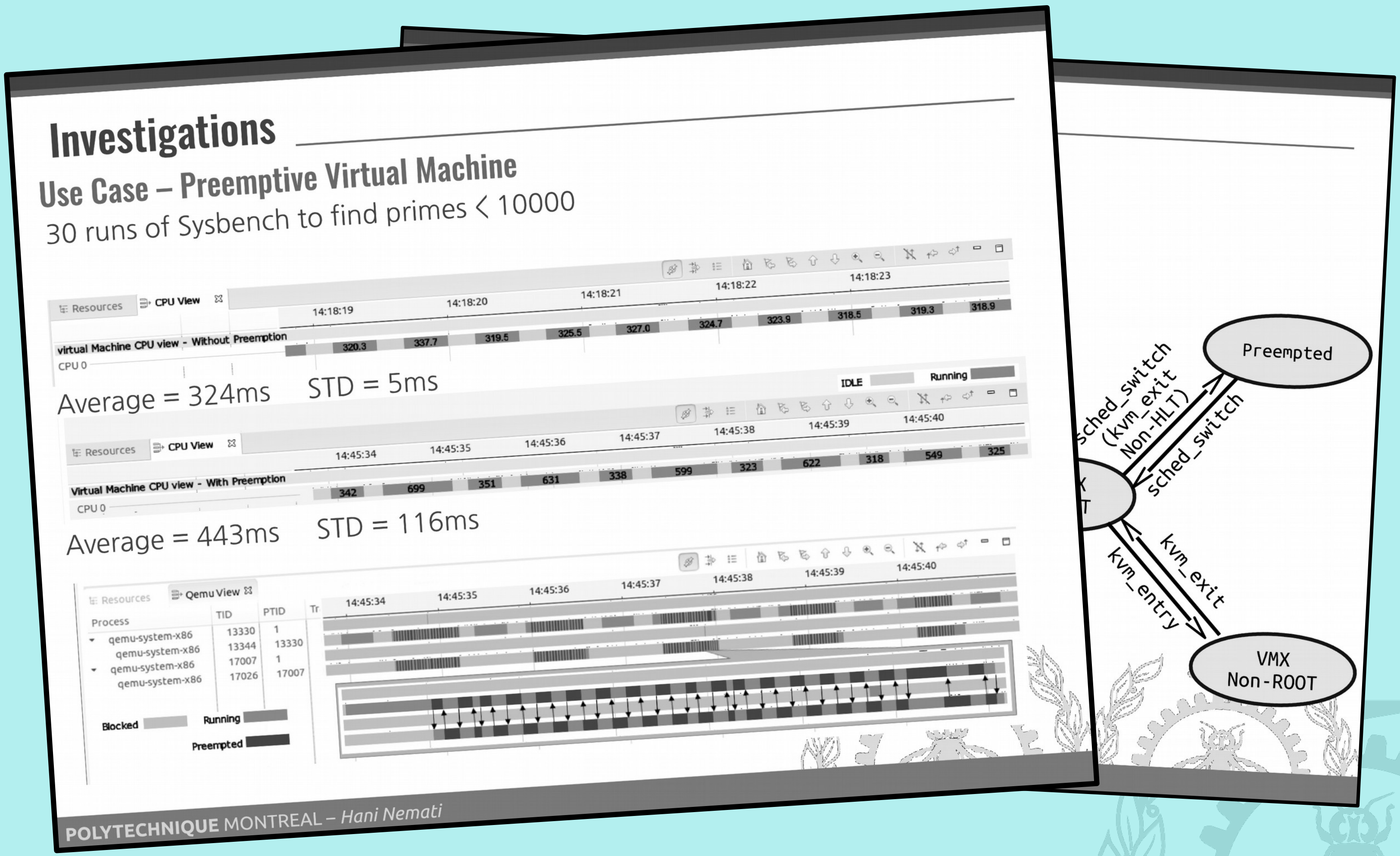
- **Resource Overcommitment Detection**

  CPU Running State Analysis
  CPU overcommitment Analysis
  Memory overcommitment Analysis

B

5 May 2016

Step 02-Overcommitment


Investigations

Use Case – Preemptive Virtual Machine

30 runs of Sysbench to find primes < 10000

Average = 324ms STD = 5ms

Average = 443ms STD = 116ms

POLYTECHNIQUE MONTREAL – *Hani Nemati*

Previously on VM Analysis

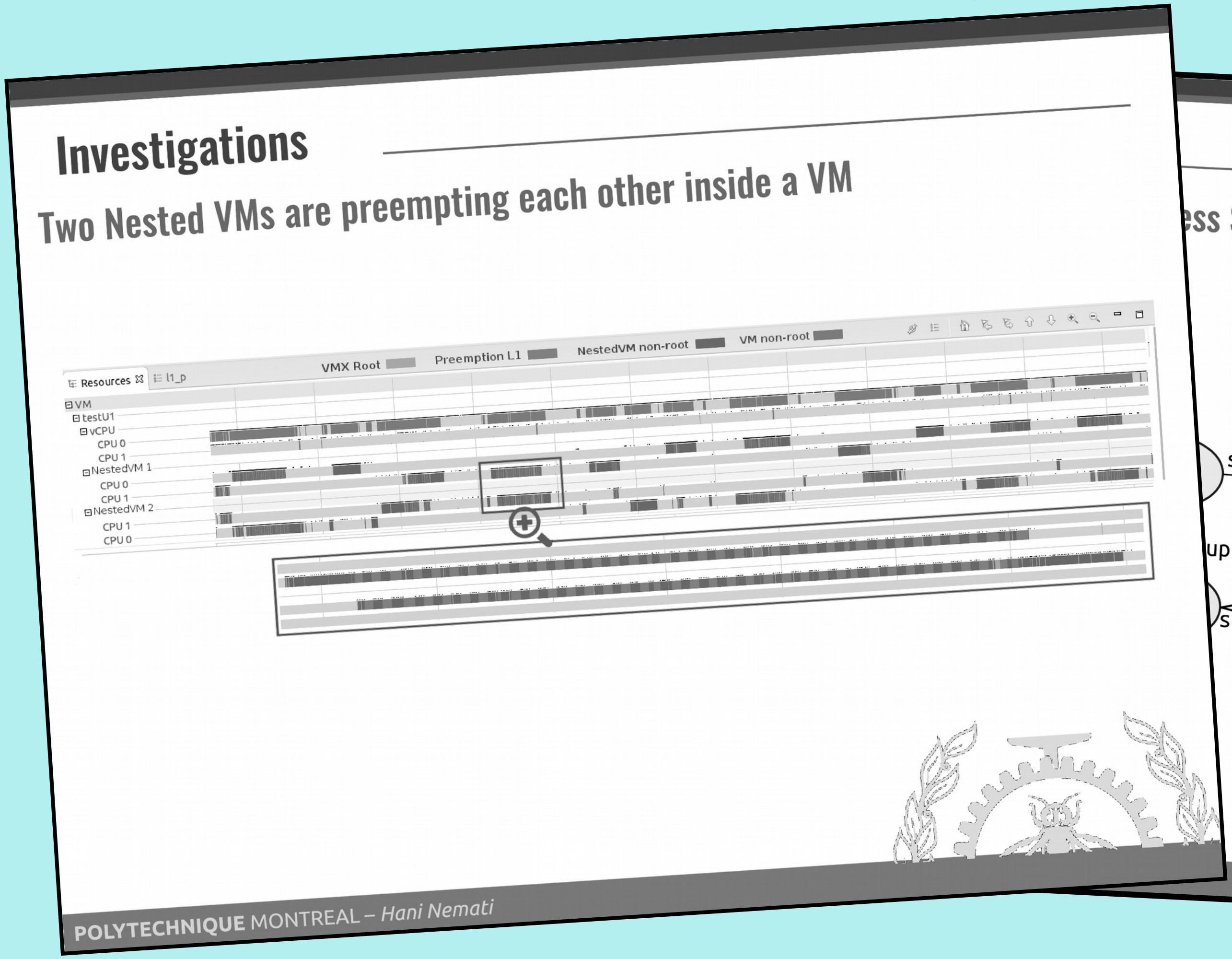
12 Dec 2016

C

Step 03-Nested VM

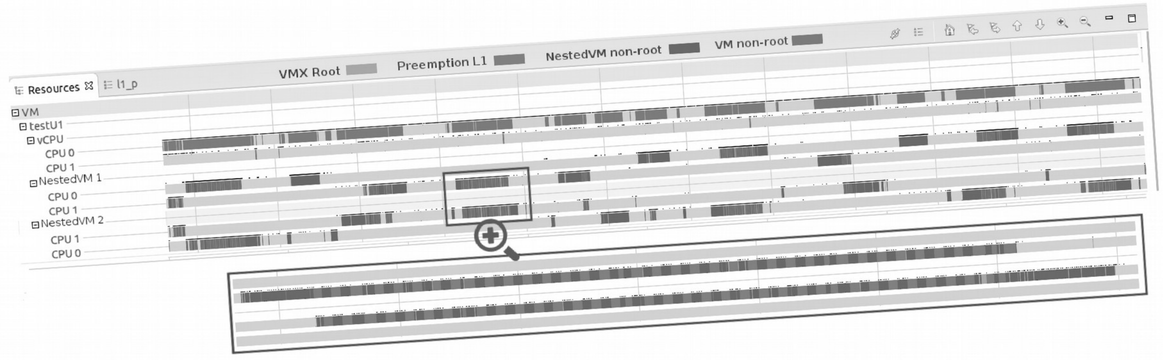
- **Any level Nested VM State Analysis**

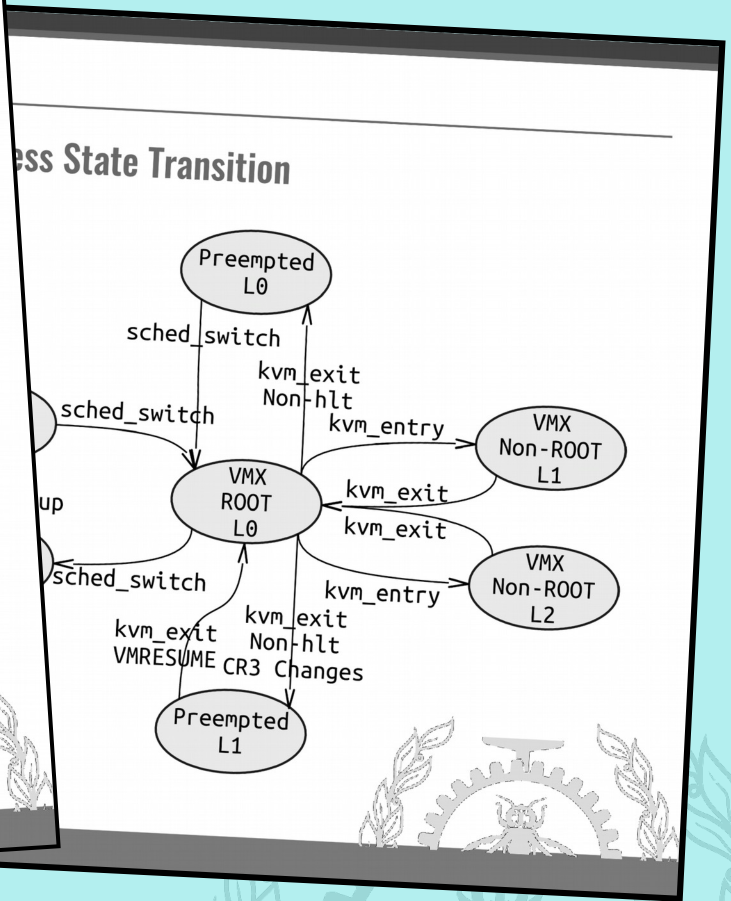
  Nested VM vCPU state analysis
  Nested VM overcommitment detection
  Virtualization overhead analysis



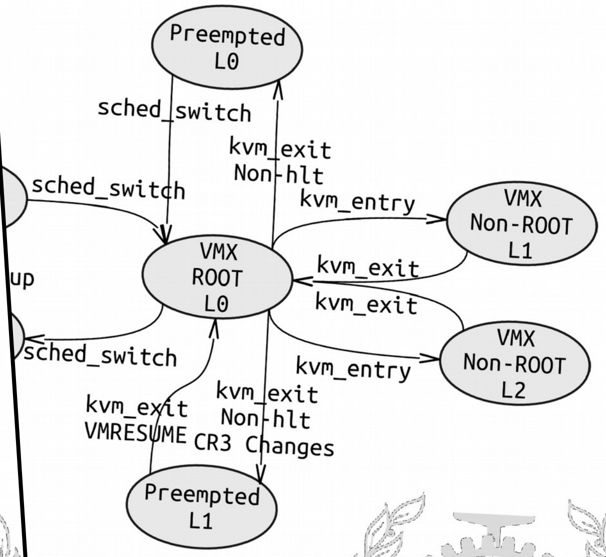
Investigations

Two Nested VMs are preempting each other inside a VM





ess State Transition



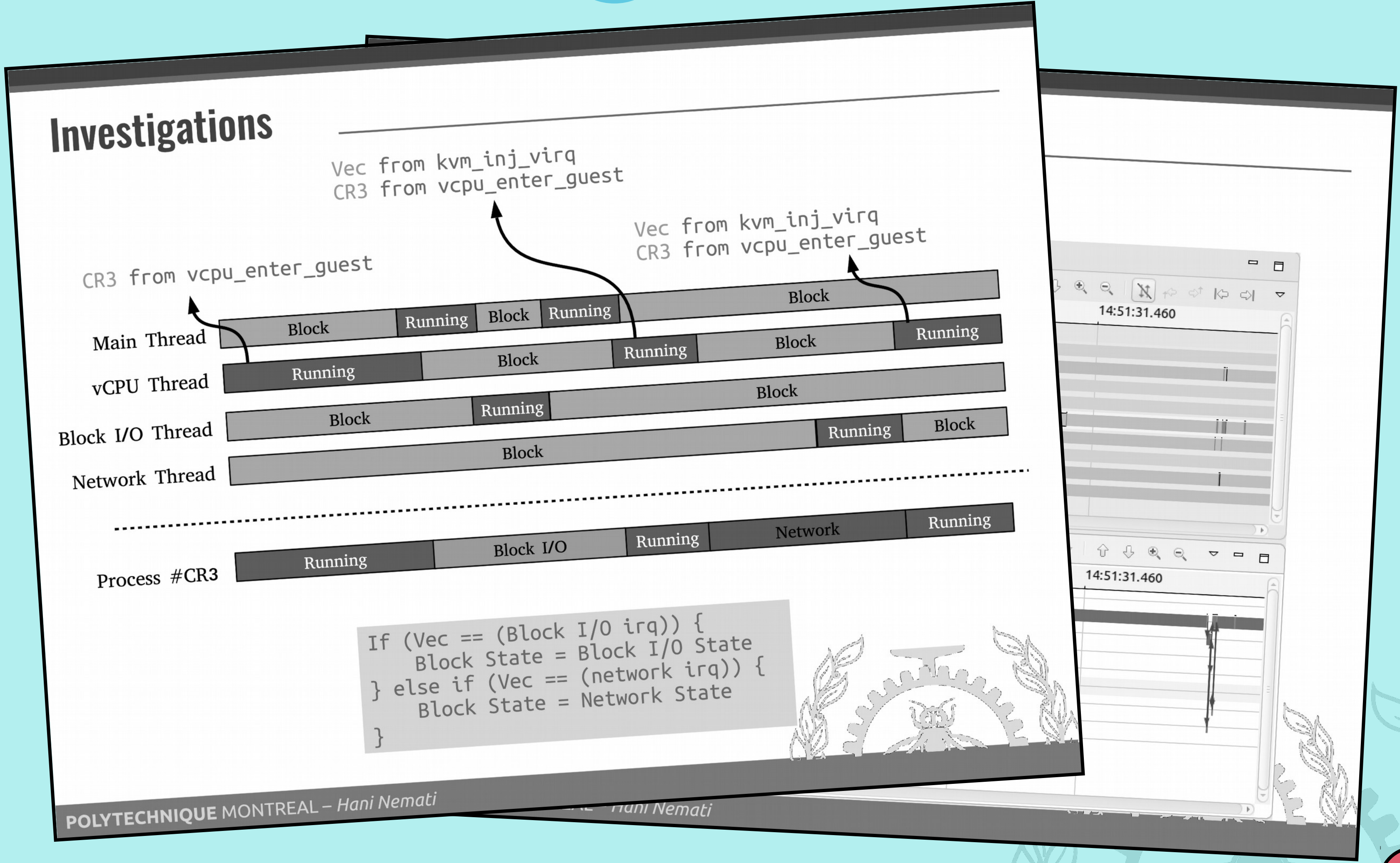
Previously on VM Analysis

- **vCPU idle State Analysis**
  Virtual interrupt injection Analysis
  Analyzing idle State of VM
  eBPF based VM analysis

D

5 May 2017

Step 04-Wait Analysis



```
If (Vec == (Block I/O irq)) {
    Block State = Block I/O State
} else if (Vec == (network irq)) {
    Block State = Network State
}
```

Previously on VM Analysis

Step 05-Critical Path

7 Dec 2017

E

- **Execution Path Analysis of VM**
  Contention of Disk
  Cap on virtual resources
  VM boot-up comparison

Previously on VM Analysis

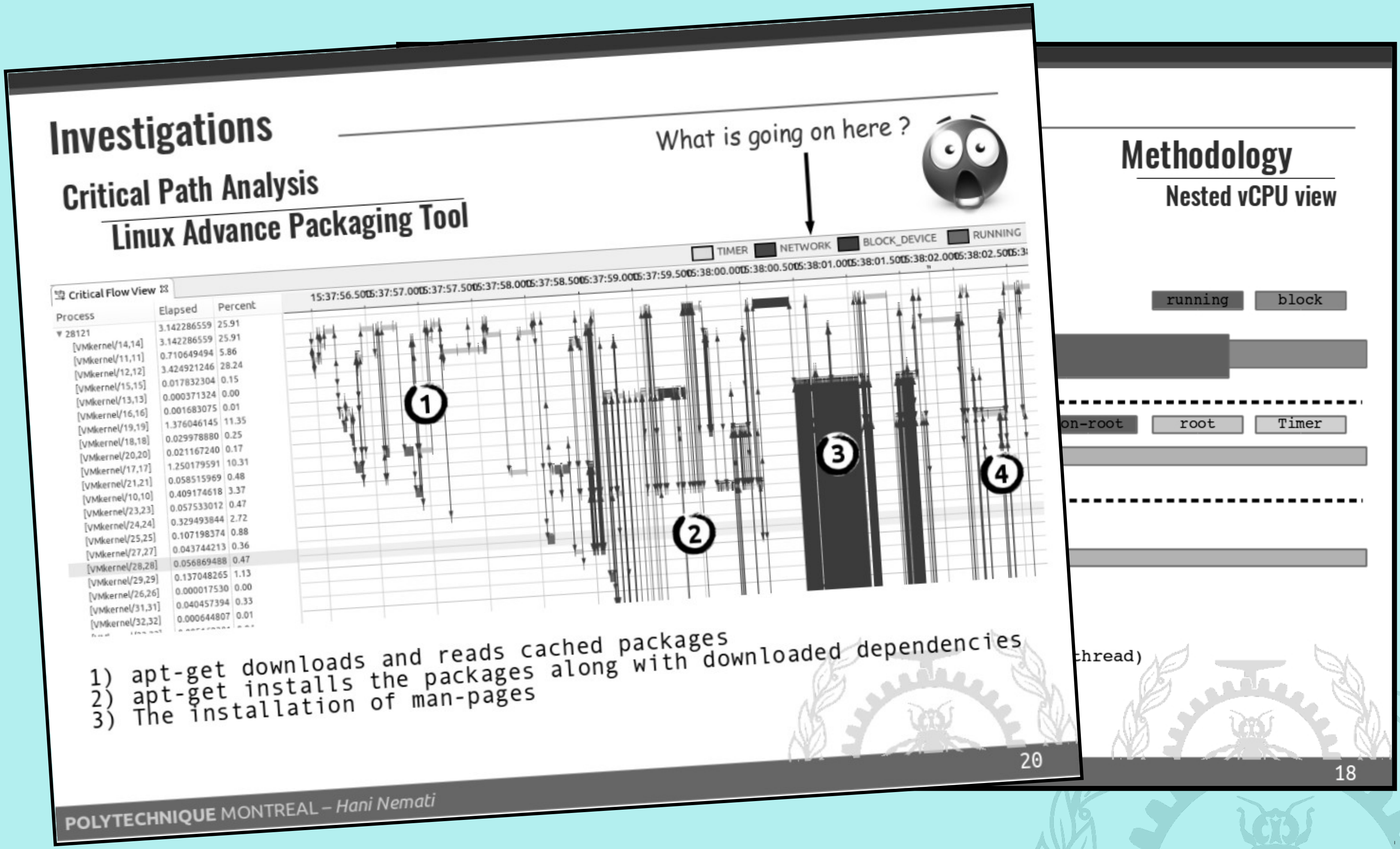
- **Hierarchical Distributed Critical Path**

  Critical Path for Nested VM
  Critical Path for distributed VMs

F

10 May 2018

Step 06-Distributed VM



Investigations

Critical Path Analysis

Linux Advance Packaging Tool

What is going on here ?

Critical Flow View

Process	Elapsed	Percent
▾ 28121	3.142286559	25.91
[VMkernel/14,14]	3.142286559	25.91
[VMkernel/11,11]	0.710649494	5.86
[VMkernel/12,12]	3.424921246	28.24
[VMkernel/15,15]	0.017832304	0.15
[VMkernel/13,13]	0.000371324	0.00
[VMkernel/16,16]	0.001683075	0.01
[VMkernel/19,19]	1.376046145	11.35
[VMkernel/18,18]	0.029978880	0.25
[VMkernel/20,20]	0.021167240	0.17
[VMkernel/17,17]	1.250179591	10.31
[VMkernel/21,21]	0.058515969	0.48
[VMkernel/10,10]	0.409174618	3.37
[VMkernel/23,23]	0.057533012	0.47
[VMkernel/24,24]	0.329493844	2.72
[VMkernel/25,25]	0.107198374	0.88
[VMkernel/27,27]	0.043744213	0.36
[VMkernel/28,28]	0.056869488	0.47
[VMkernel/29,29]	0.137048265	1.13
[VMkernel/26,26]	0.000017530	0.00
[VMkernel/31,31]	0.040457394	0.33
[VMkernel/32,32]	0.000644807	0.01

```
1) apt-get downloads and reads cached packages
2) apt-get installs the packages along with downloaded dependencies
3) The installation of man-pages
```

20

POLYTECHNIQUE MONTREAL – *Hani Nemati*

Methodology

Nested vCPU view

18

Previously on VM Analysis

F

10 May 2018

Step 06-Distributed VM

- Hierarchical Distributed Critical Path

E

7 Dec 2017

Step 05-Critical Path

- Execution Path Analysis of VM

D

5 May 2017

Step 04-Wait Analysis

- vCPU idle State Analysis

C

12 Dec 2016

Step 03-Nested VM

- Any level Nested VM State Analysis

B

5 May 2016

Step 02-Overcommitment

- Resource Overcommitment Detection

A

10 Dec 2015

Step 01-Monitoring

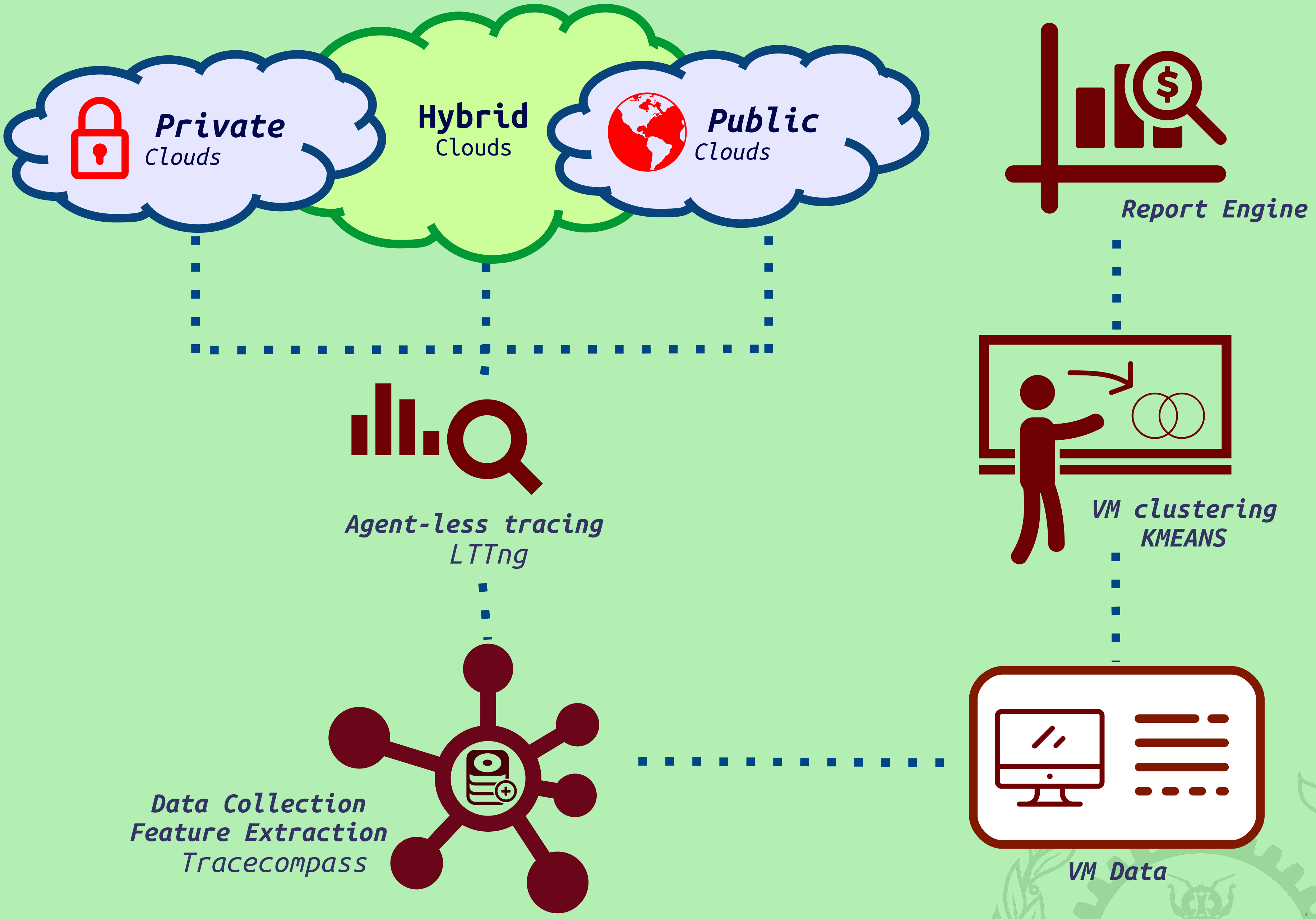
- Fine-grain resource Monitoring

Automation

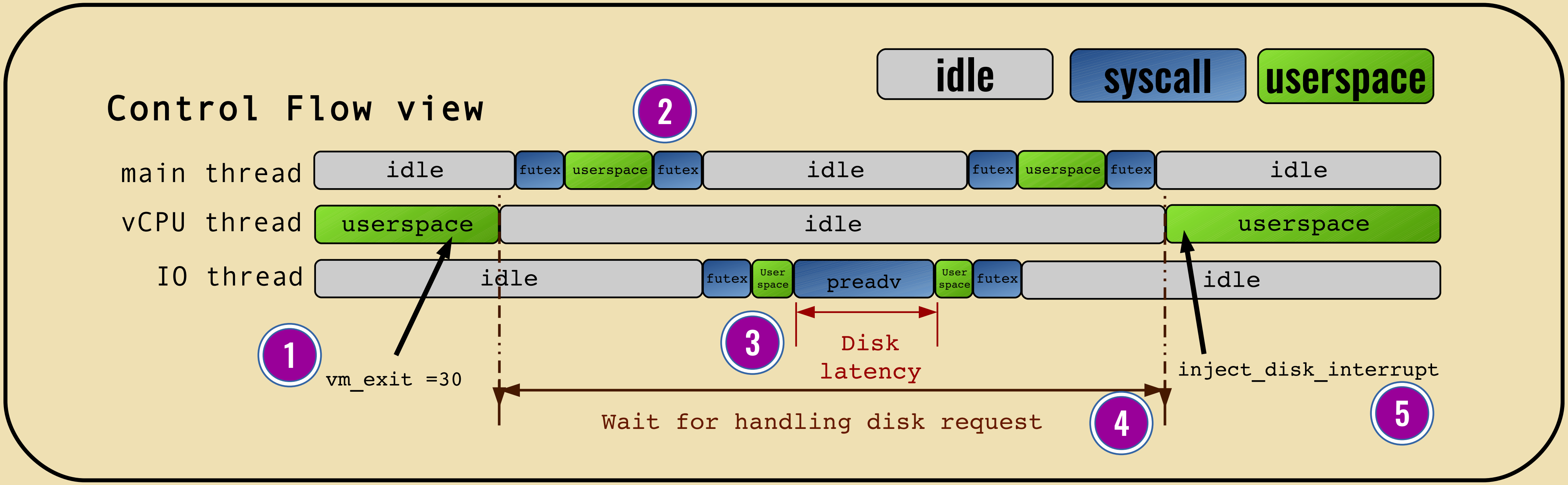
How about

Automation?

VM Clustering Architecture

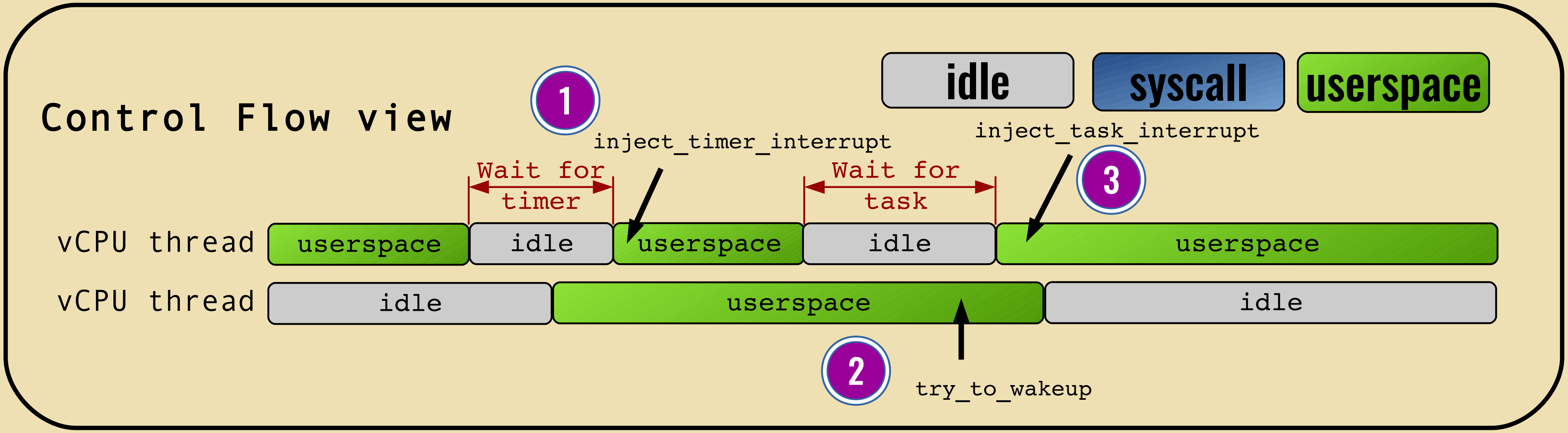


Disk Metrics



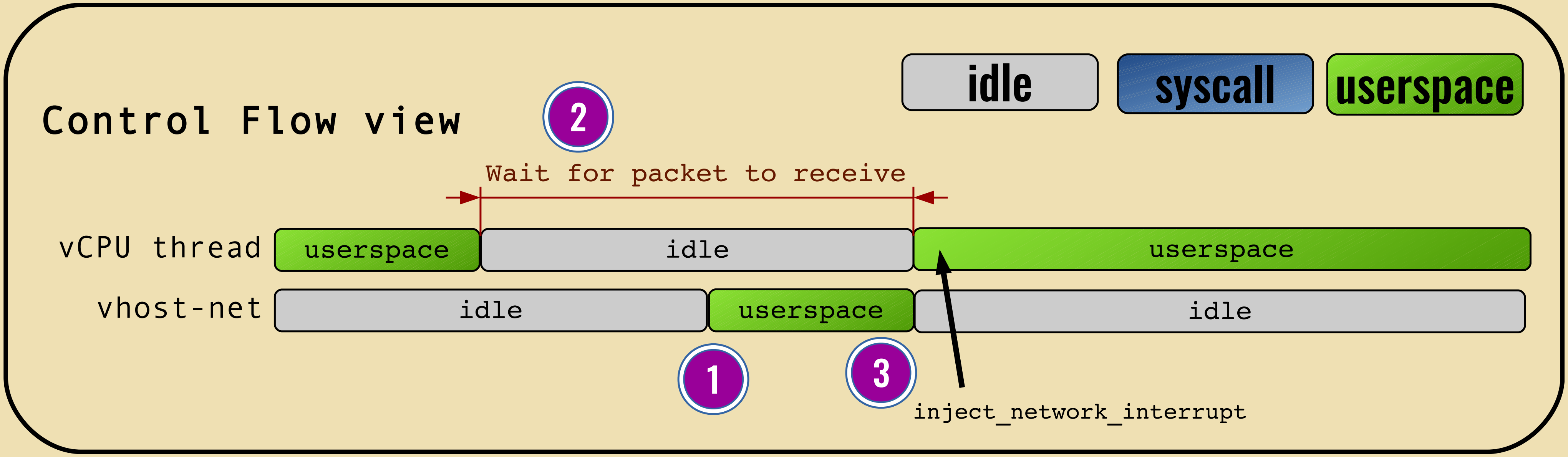
1. VM exits from guest mode to host mode to handle disk request
2. Qemu main thread submits the disk request
3. Qemu IO thread handles the disk request
4. Two metrics: wait for handling disk request and disk latency
5. Virtual disk interrupt is injected to VM

CPU Metrics



1. Virtual timer interrupt is injected to VM
2. A VM process wakes up another VM process
3. Virtual task interrupt is injected to VM

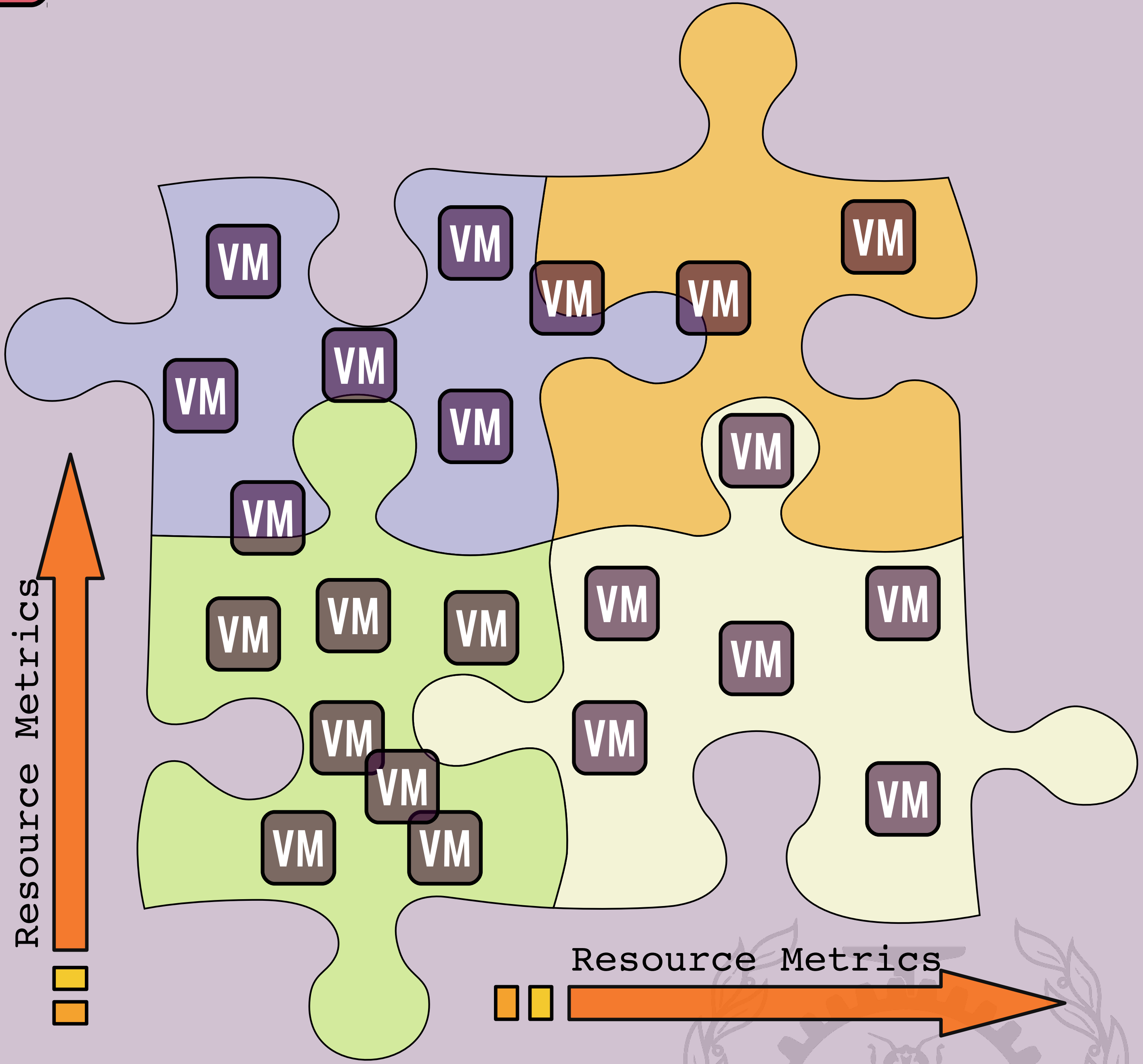
Network Metrics



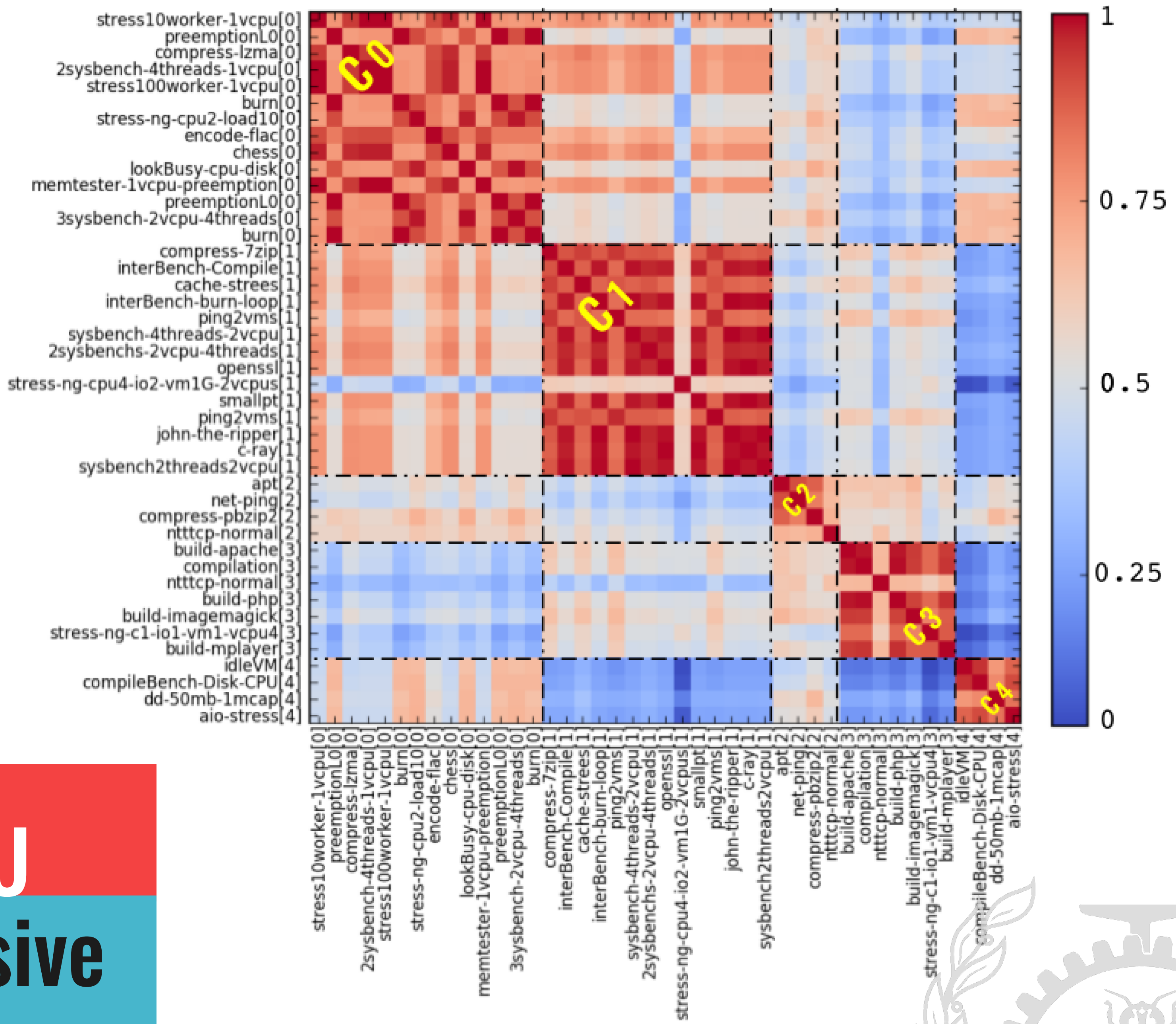
1. A packet is received by vhost-net
2. Wait for a packet to receive
3. Virtual network interrupt is injected to VM

Experimental Evaluation

k-means
Clustering

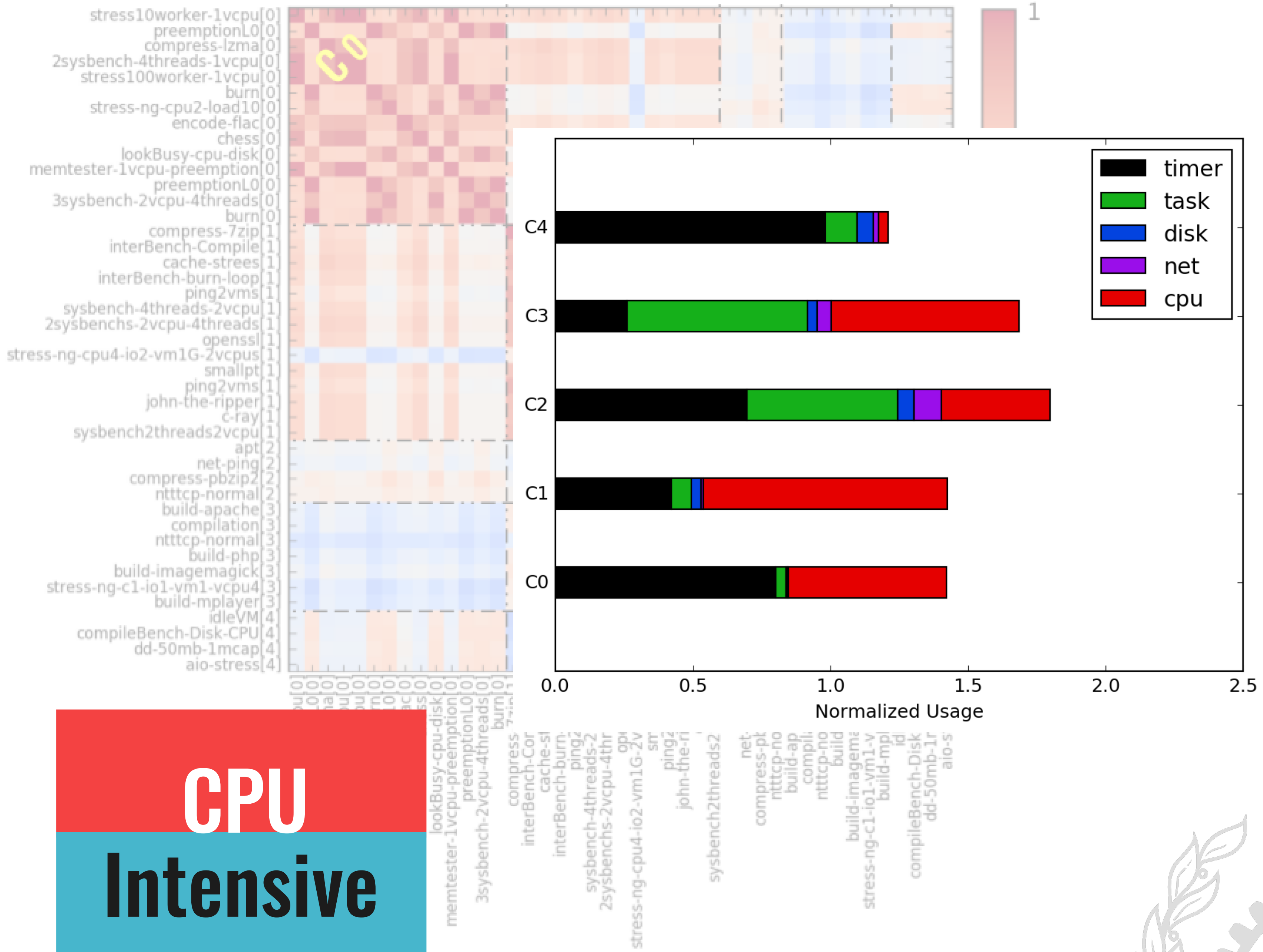


Experimental Evaluation



CPU Intensive

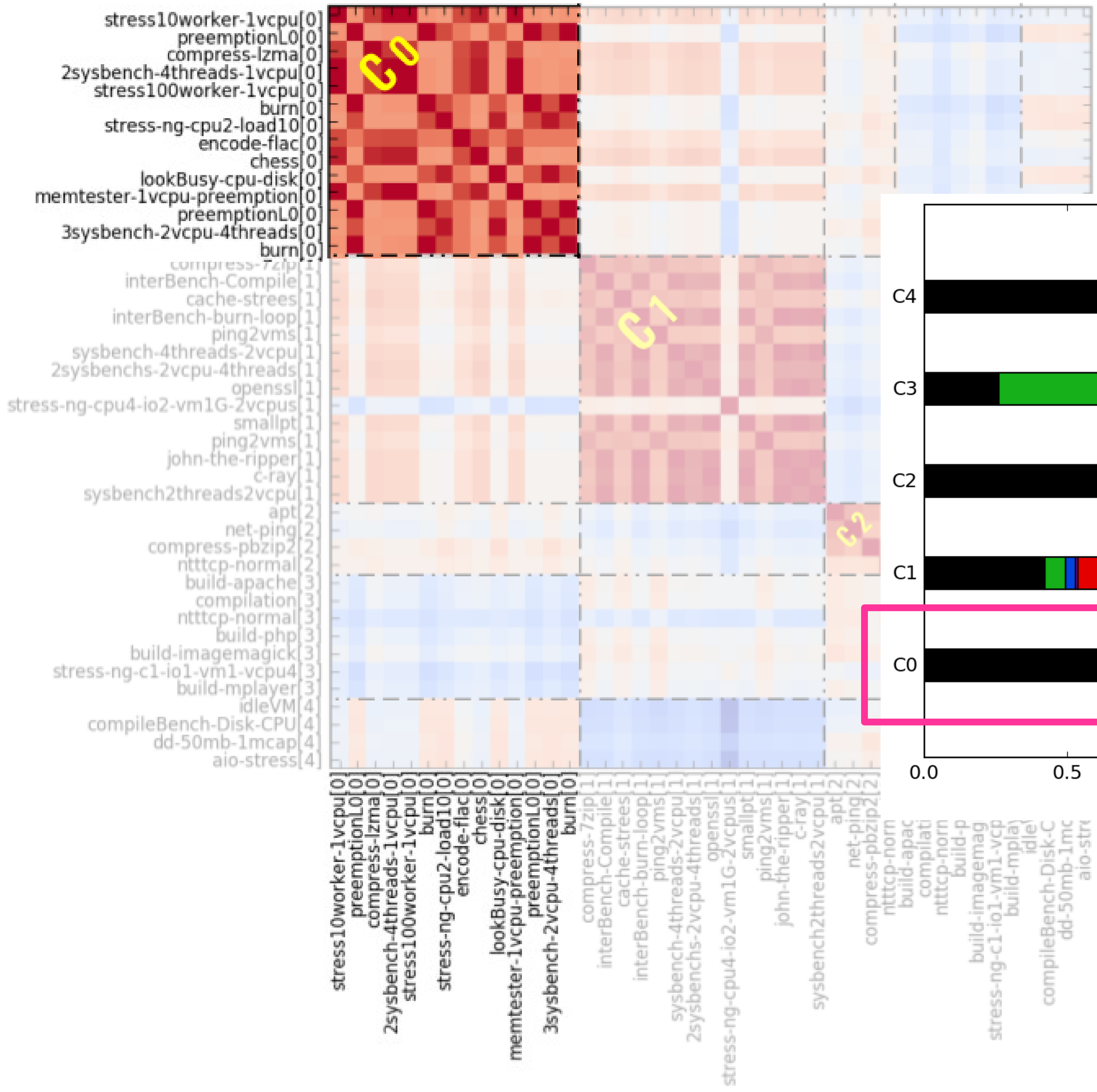
Experimental Evaluation

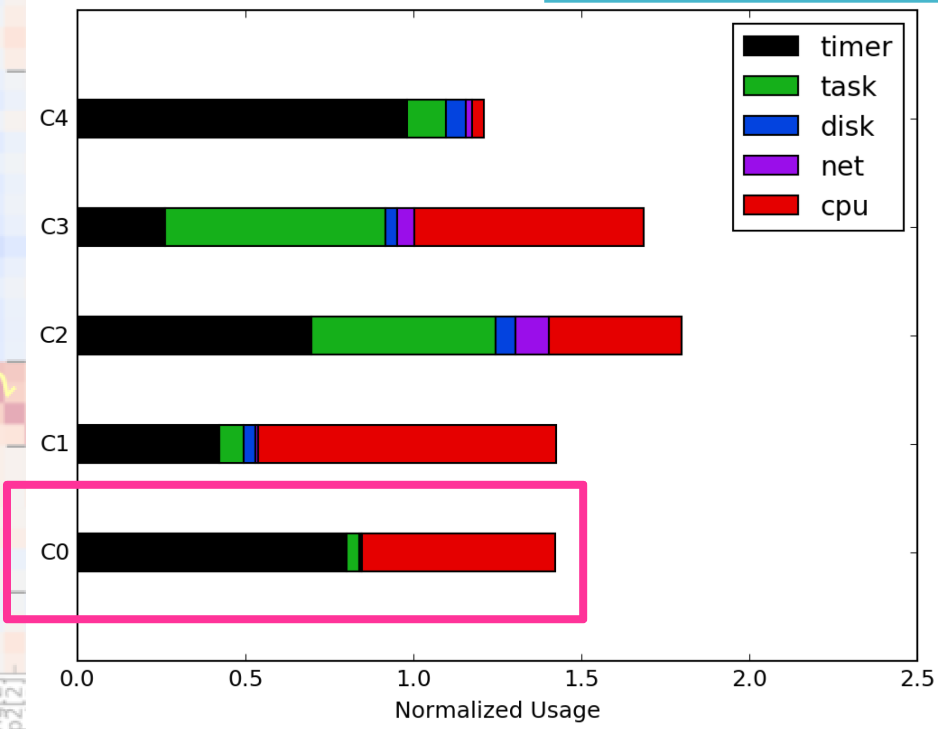


CPU Intensive

Experimental Evaluation

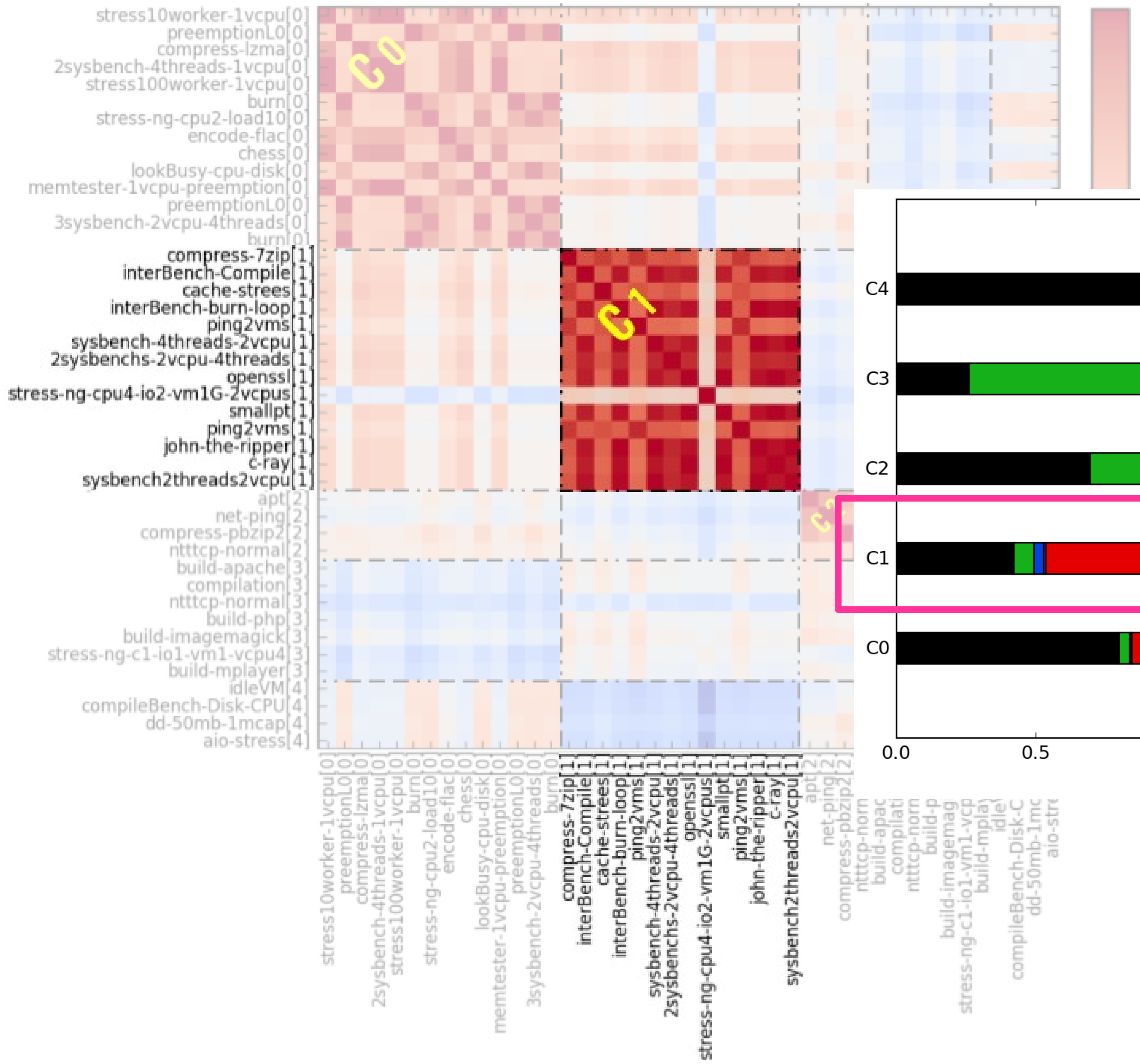
CPU
Intensive

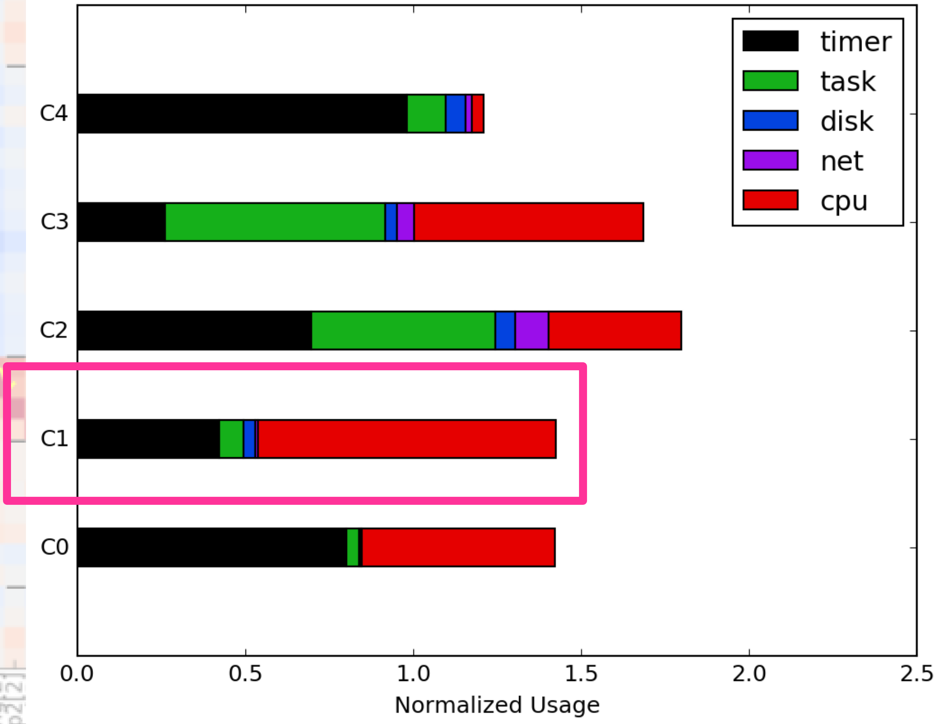




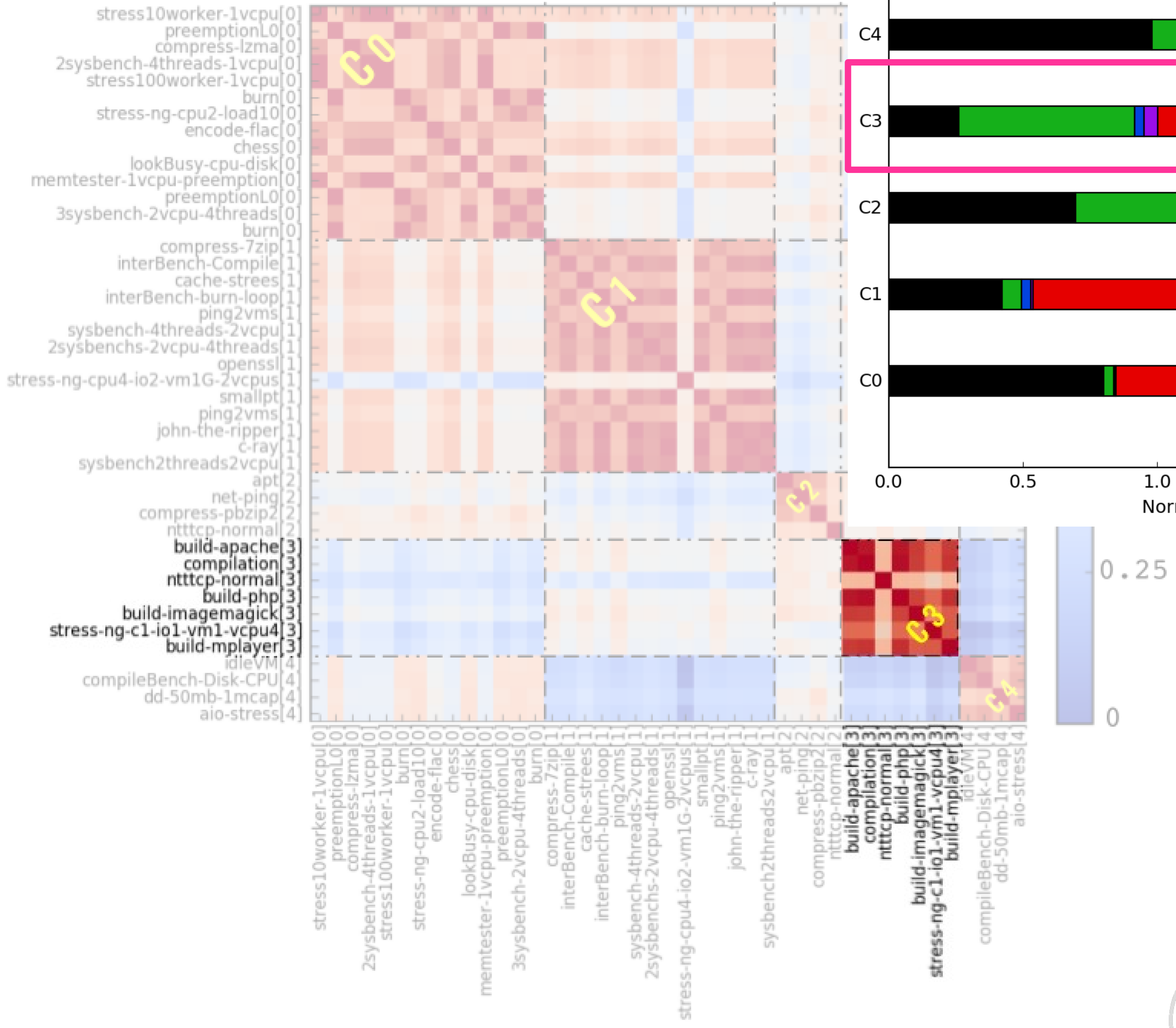
Experimental Evaluation

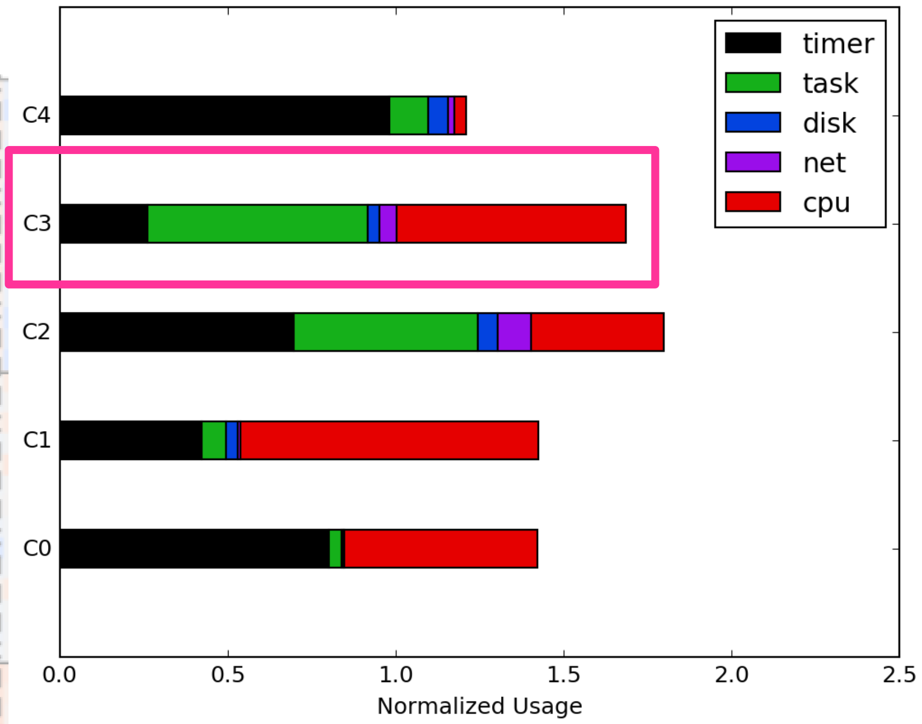
CPU Intensive





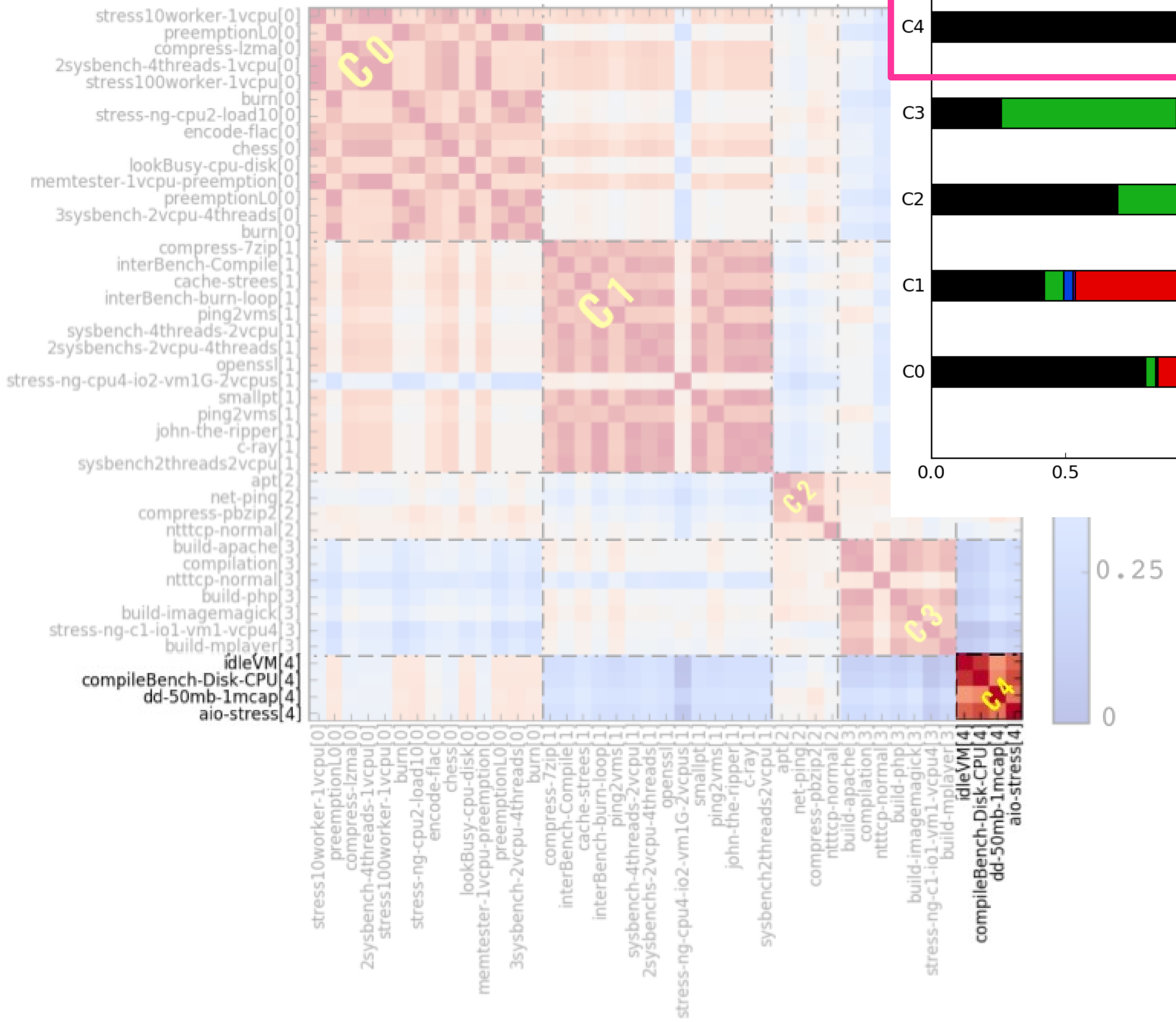
Experimental Evaluation

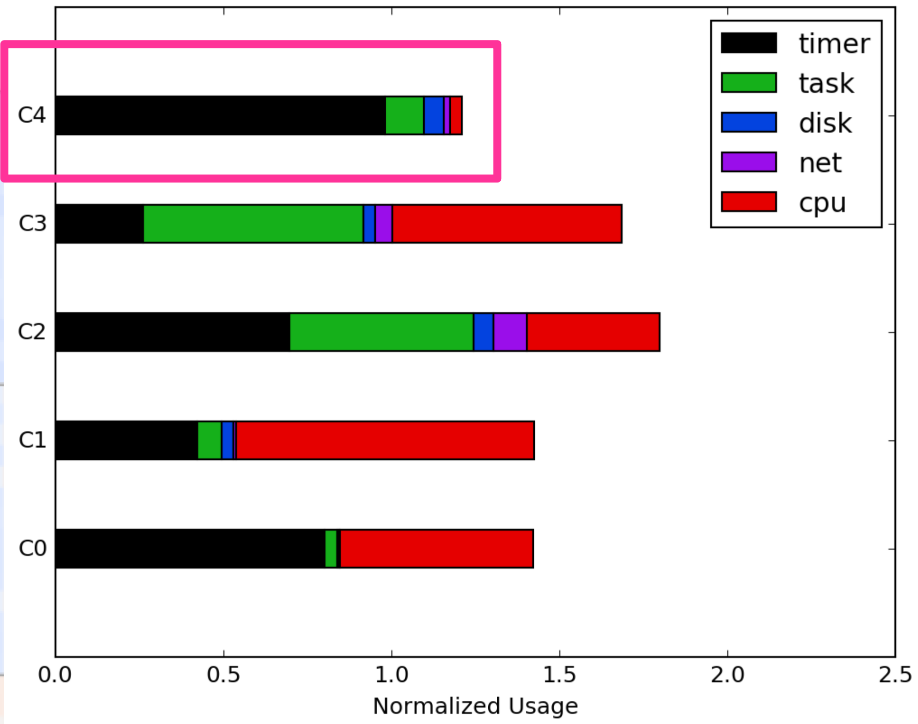




CPU
Intensive

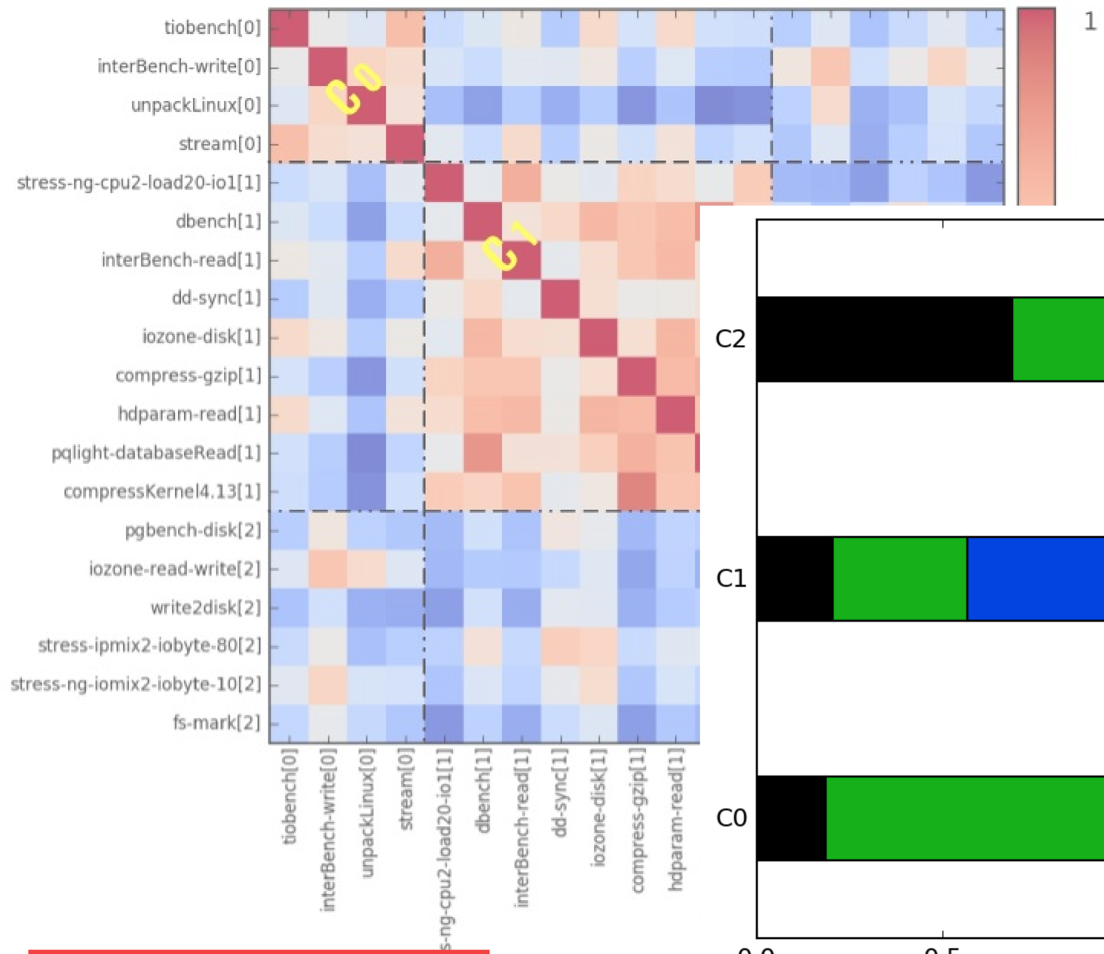
Experimental Evaluation

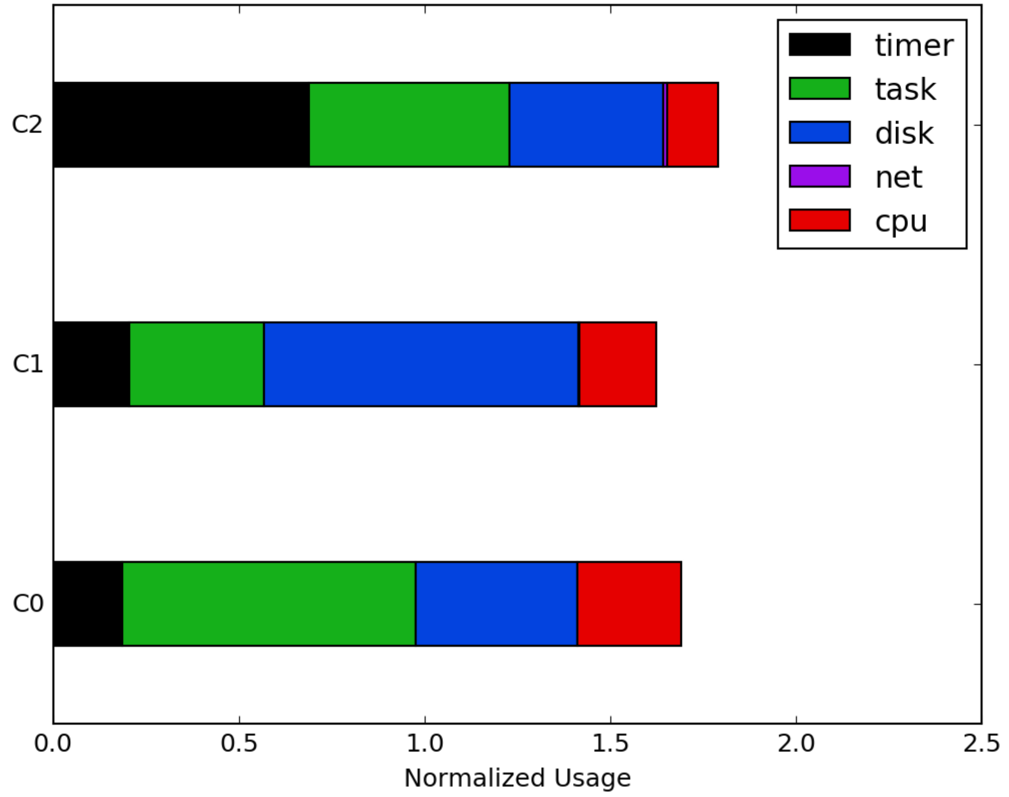




CPU Intensive

Experimental Evaluation

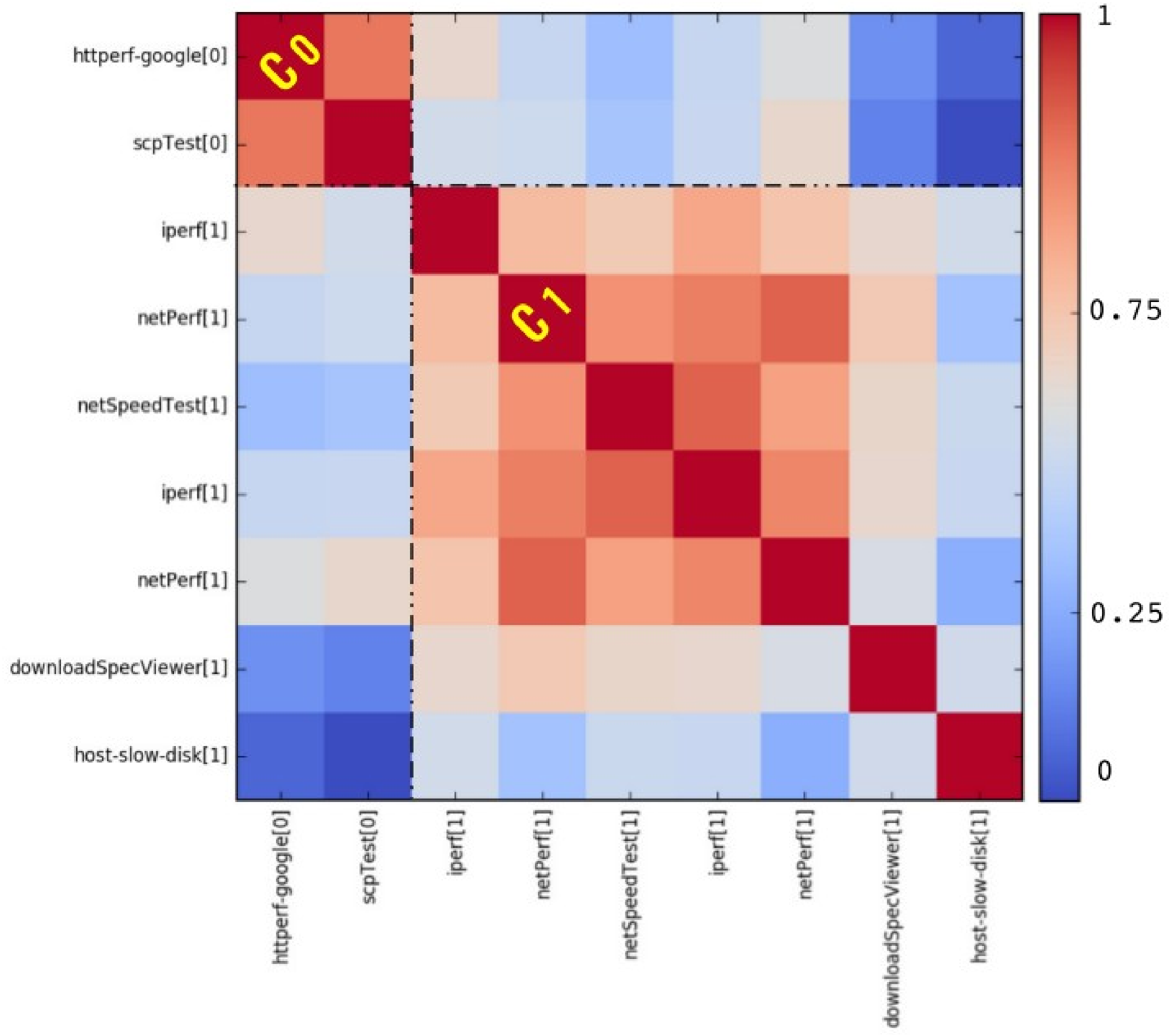


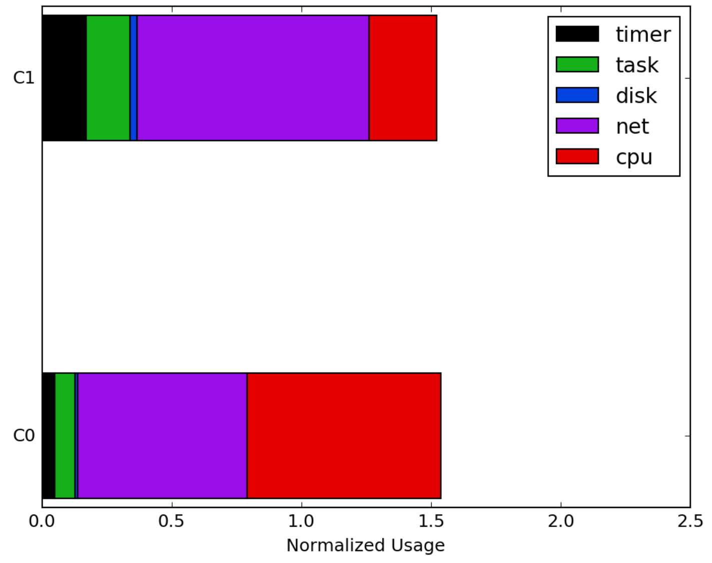


Disk Intensive

Experimental Evaluation

Network Intensive





Experimental Evaluation

How to Try

1. Access to Host only
2. Run LTTng on Host with my new added tracepoint (vcpu_enter_guest)
3. Clone TraceCompass from github (incubator:https://github.com/Nemati/org.eclipse.tracecompass.incubator.git)
4. Git checkout vm2
5. Clone lamiminer from github (https://github.com/azharivs/lamiminer.git)

Conclusions

1. Clustering VM based on injected interrupts to VM
2. CPU intensive group has high timer interrupt and high CPU usage
3. Disk intensive group has high disk interrupt and high task interrupt
4. Network intensive group has high network interrupt and high CPU usage

Questions

azharivs@gmail.com
http://github.com/azharivs

Hani.nemati@polymtl.ca
http://github.com/Nemati