DCMIX: Generating Mixed Workloads for the Cloud Data Center

XingWang Xiong, Lei Wang, **WanLing Gao**, Rui Ren, Ke Liu, Chen Zheng, Yu Wen, YiLiang

Institute of Computing Technology, Chinese Academy of Sciences

中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY

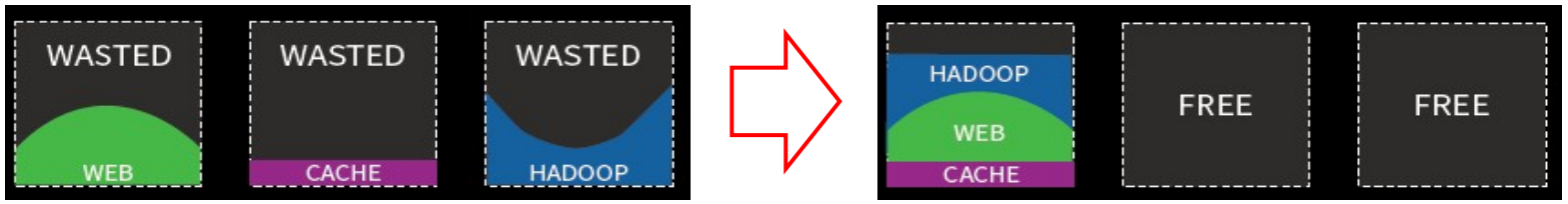
Bench 2018, Seattle, USA

Outline

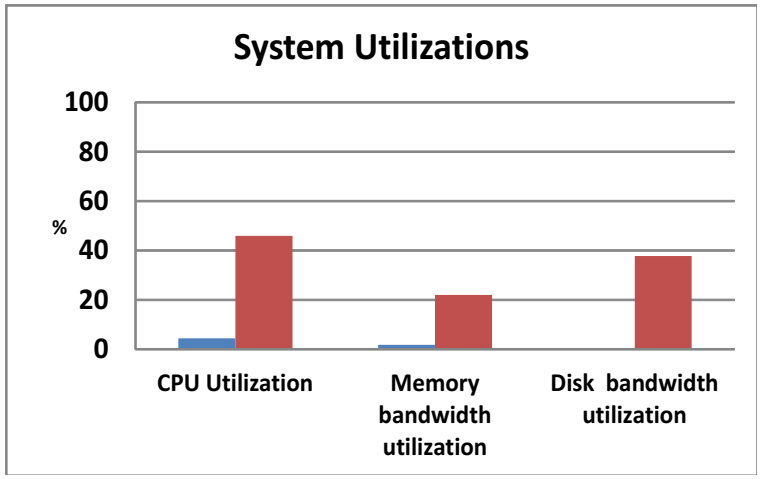
- **Motivation & challenges**
- What is DCMIX?
- System Entropy
- Experiment
- Conclusion

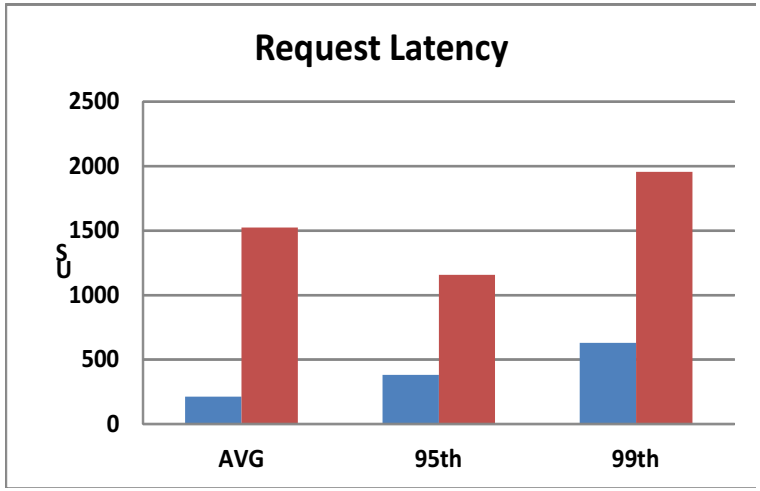
Cloud Data Center

- **Co-locating workloads on shared resources**



- **State-of-Practise**
  - Higher utilizations but poor user experience (**Resources Contentions**)



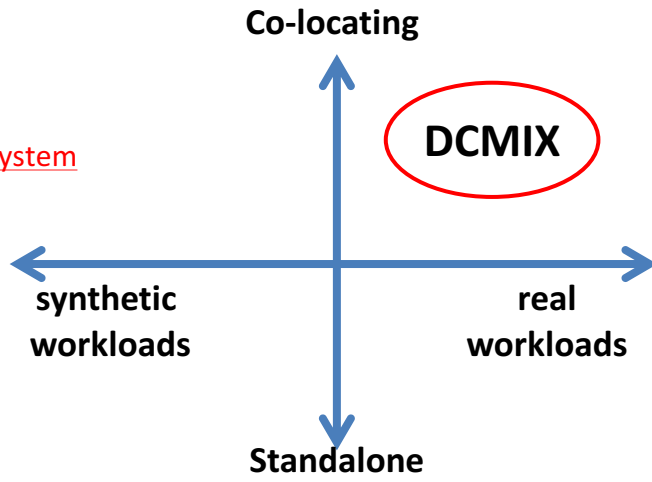


Workloads: Redis+Sort+MD5+ WordCount

Benchmarking Cloud data center

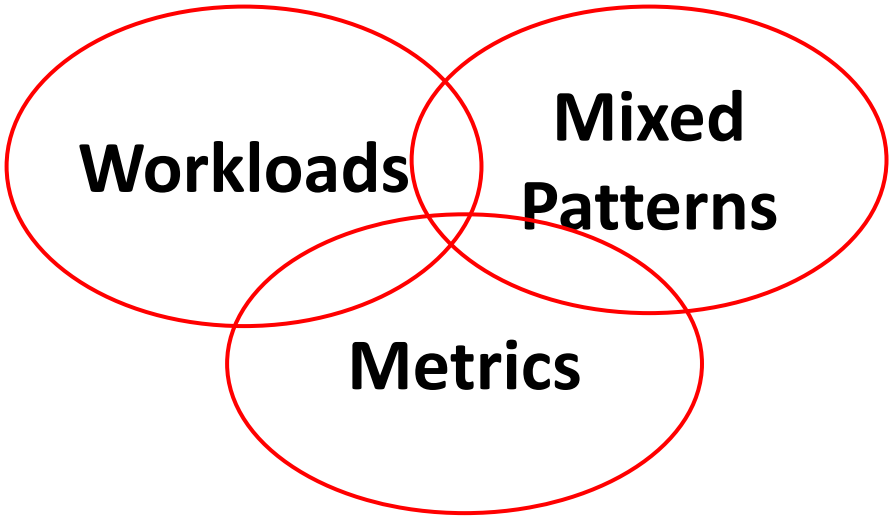
- **More and more efforts try to do it: user experiences & system utilizations**
  - Cache Allocation Technology, Linux Containers Technology ……

- **Benchmark is the first step**
  - Measuring the **Co-locating workloads** system
    - Benchmarks: vendors and users embrace it
      - From the view of users: User Experiences (latency, tail latency)
      - From the view of vendors: utilizations and contentions of the target system
  - State-of-Art Benchmarks
    - **Generating multiple workloads individually**
      - BigBench, BigDataBench, TailBench, CloudSuite
    - **Synthetic Co-locating workloads according to the trace**
      - SWIM, CloudMix



How to generate real Co-locating workloads and measure the corresponding system?

Challenges and Our contributions



Benchmarks: DCMIX

a) 17 typical cloud data center workloads, latencies of workloads range from microseconds to minutes.
b) Mixture of serial execution and parallel execution.

Metrics: the system entropy

a) The joint entropy of four system-level performance metric
b) Obtaining easily & not interfere with workloads

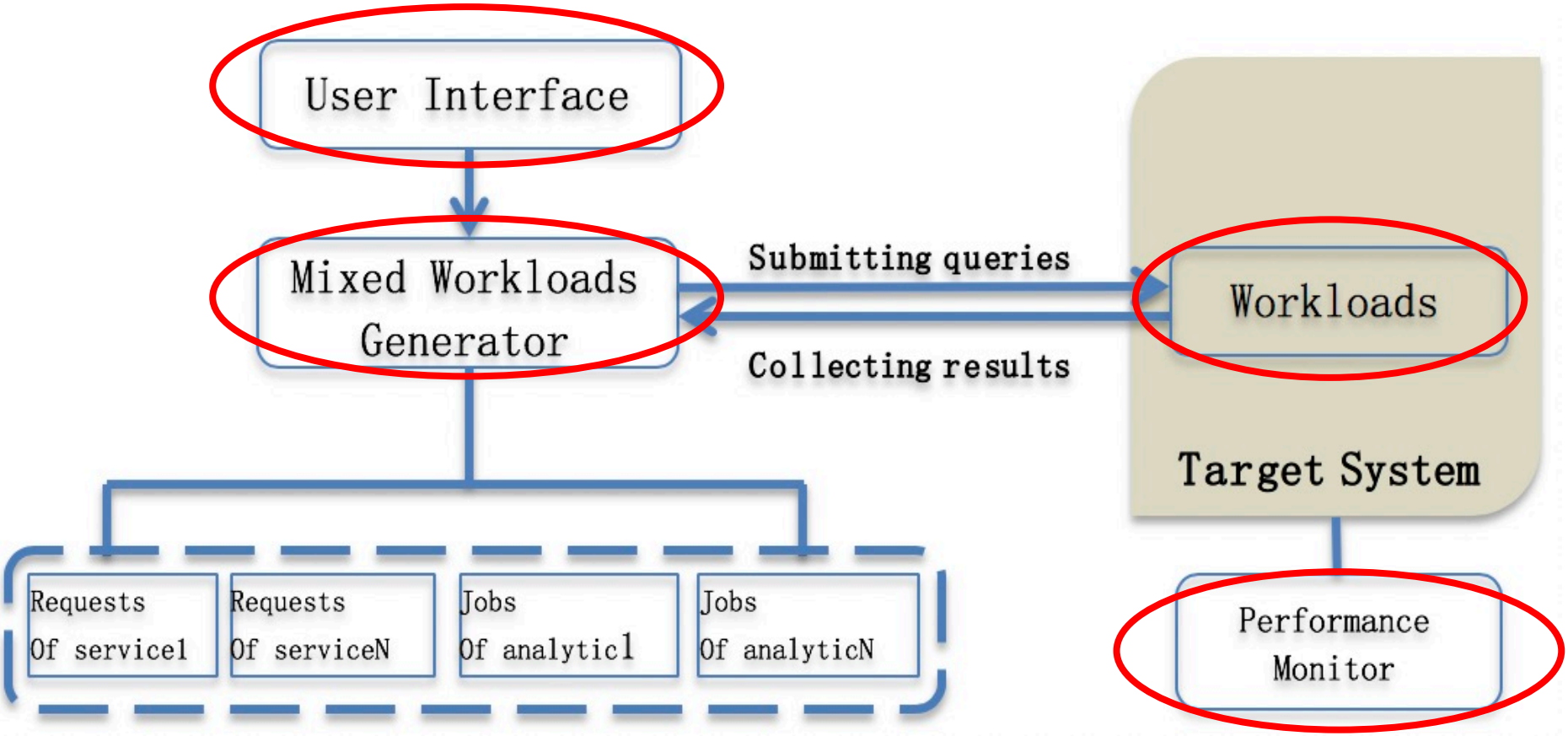
Experiments: four different modes on the X86 platform

a) Mixed mode V.s. Standalone: the latency increased 7 times, and resource utilizations increased 10 times.
b) The isolation mechanism has some efforts under the mixed mode, especially the CPU-affinity mechanism.

Outline

- Motivation & challenges
- **What is DCMIX?**
- System Entropy
- Experiment
- Conclusion

DCMIX's Framework

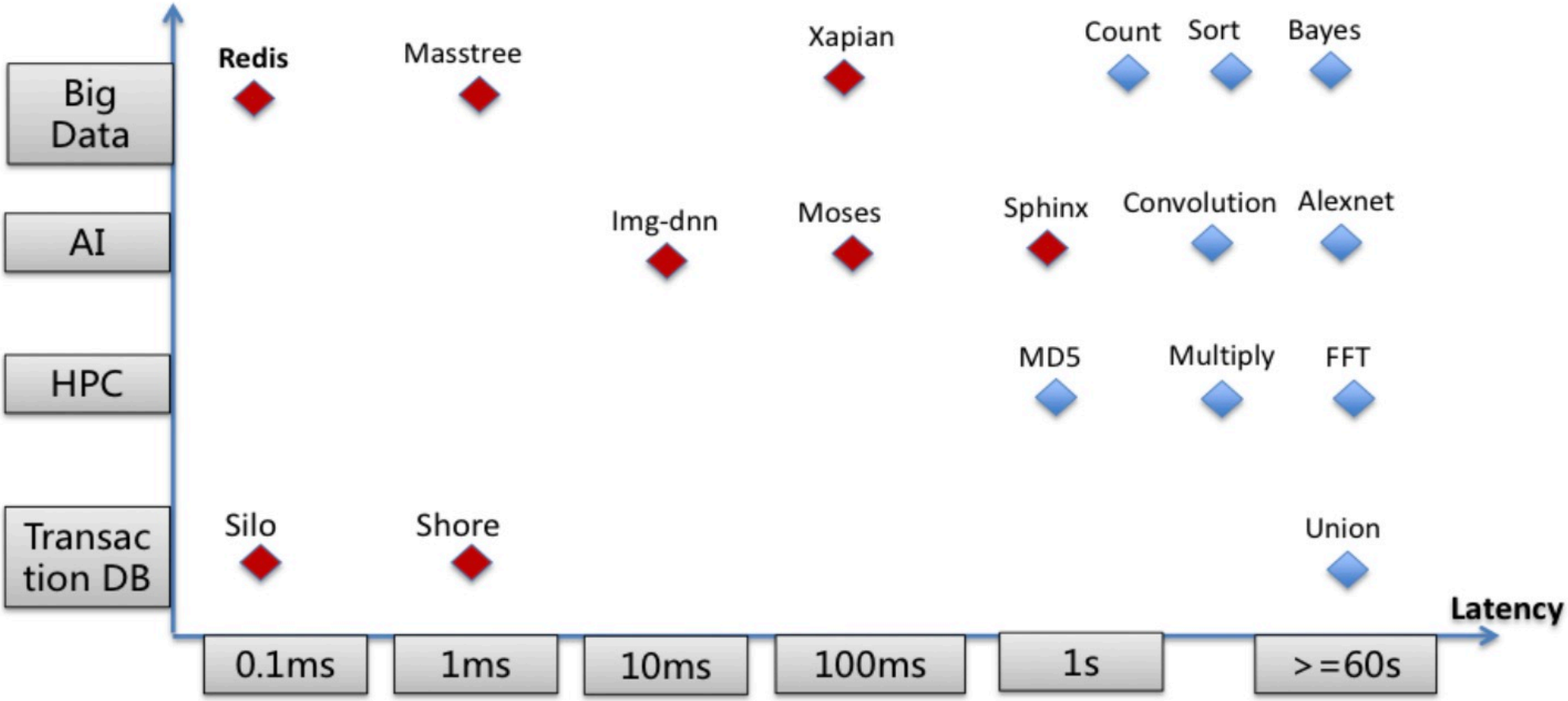


DCMIX's Workloads

- **Workloads**
  - Online services
  - Data analytic workloads
- **Application fields**
  - Big data
  - AI
  - High performance computing
  - Transaction databases

DCMIX's Workloads(Cont'd)

- Coverage both in application fields and latency



DCMIX's Workload Generator

- Mixture execution
  - Serial execution
  - Parallel execution
- User-defined configurations
  - Request intensity(i.e. QPS)
  - Number of requests
  - Number of warm-up requests
  - Data size of jobs
  - Threads number of jobs

```
17 mode = "serial" #"parallel"or"serial"
18 online_tasks  = (0 1 + 1)
19 offline_tasks = (2 3)
20 app_names = ("xapian" "redis" "sort" "wordcount")
```

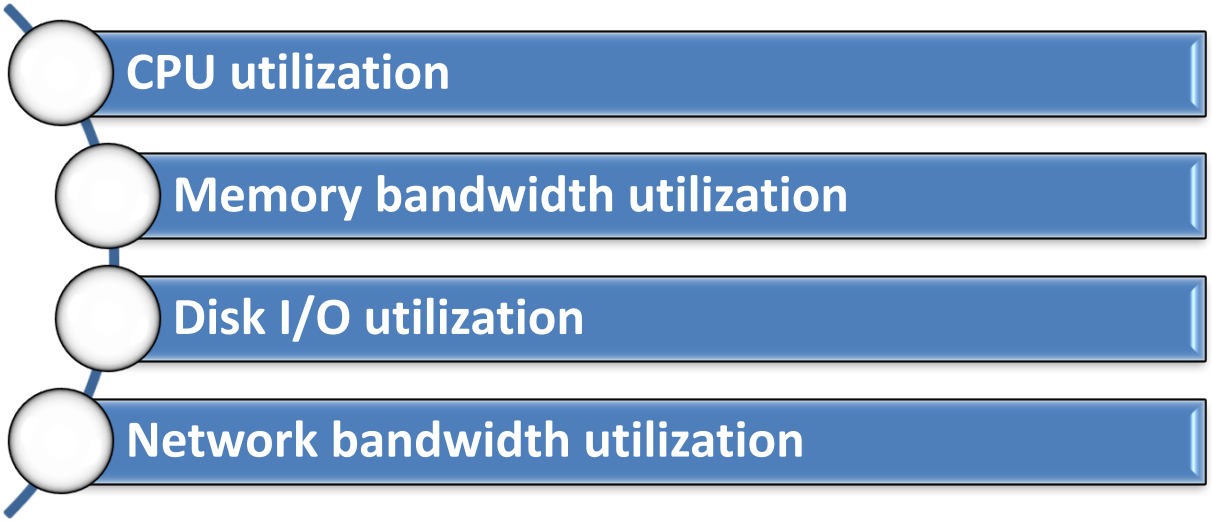
```
 7 NSERVERS=1
 8 QPS=500
 9 WARMUPREQS=1000
10 REQUESTS=3000
11 MINSLEEPNS=10000
```

Outline

- Motivation & challenges
- What is DCMIX?
- **System Entropy**
- Experiment
- Conclusion

Motivations

- Metrics that the cloud vendor considered
  - System resource utilizations: **benefits**



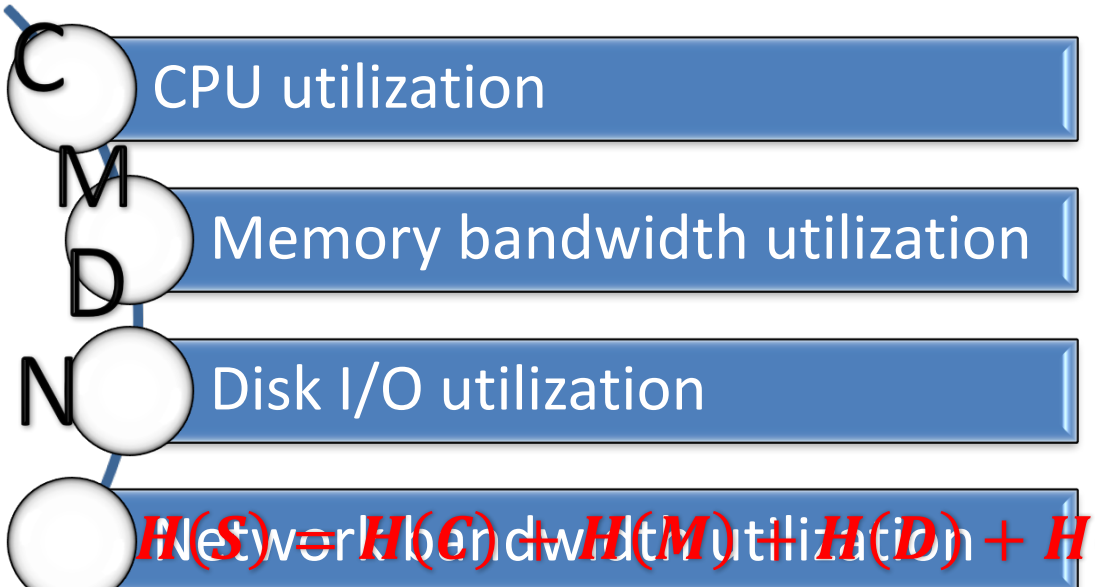
  - System resource contentions： **disturbances**
    - **The system entropy**
      - the uncertainty associated with resources usage

The Definition of System Entropy

- Based on Shannon entropy
  - the uncertainty associated with a random variable

$$H(X) = -\sum_{x \in X} p(x) * \log_2 p(x)$$

- Joint entropies of four system-level metrics



$$H(S) = H(C) + H(M) + H(D) + H(N)$$

System Entropy Calculation

- The sum of four element's entropies

$$H(S) = H(C) + H(M) + H(D) + H(N)$$

- Element's entropies based on Shannon entropy

$$H(X) = -\sum_{x \in X} p(x) * \log_2 p(x)$$

$$p(c) = \frac{Num(c)}{n}$$

Outline

- Motivation & challenges
- What is DCMIX?
- System Entropy
- **Experiment**
- Conclusion

Experiment Configurations

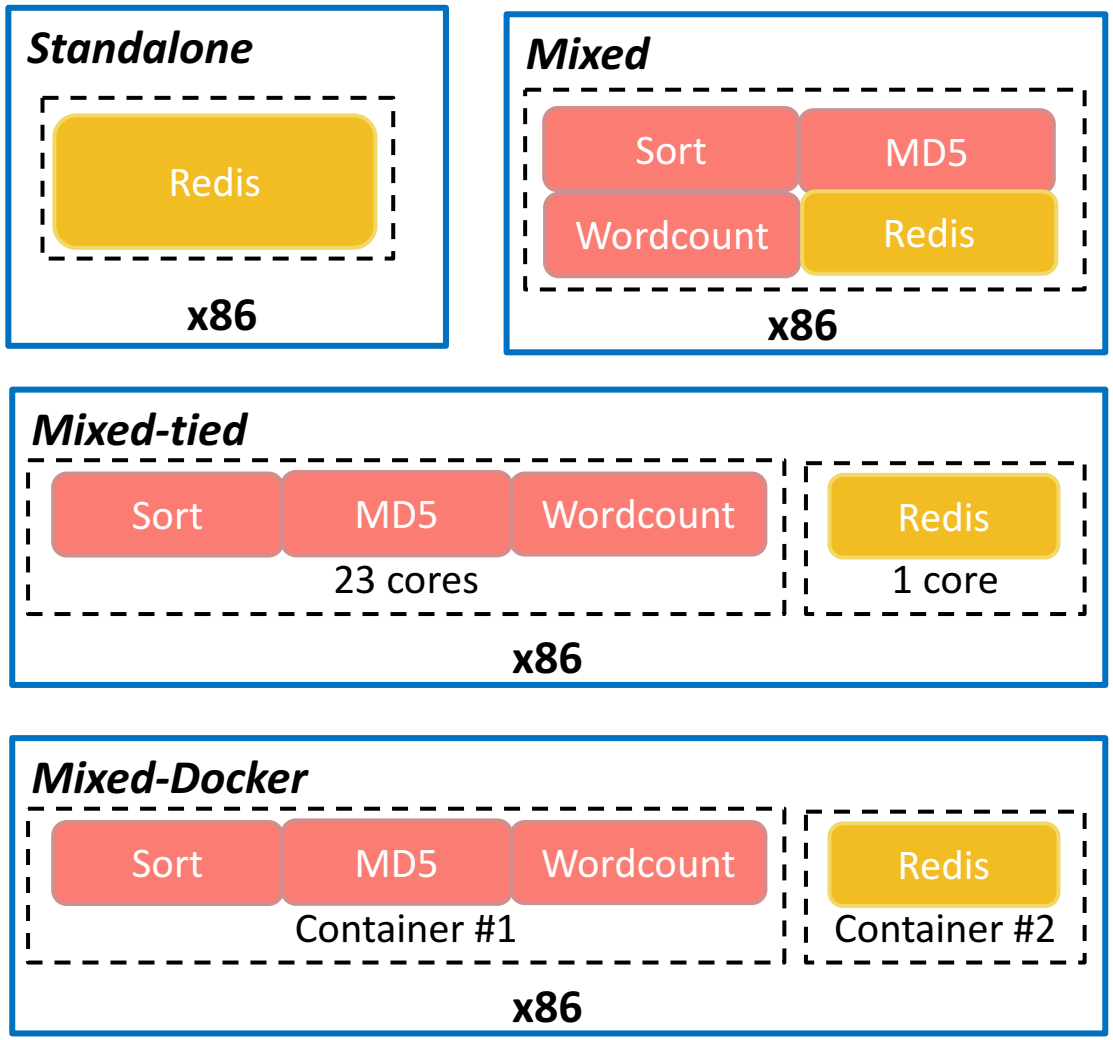
- Two Nodes:
  - The Client Node
    - workloads generator
  - The Server Node
    - target node
- Data analytic workloads(8GB text data)
  - Sort, MD5, WordCount
- Redis Requests
  - 50,000 requests per second

CPU	Intel(R) Xeon(R) E5645 2.40G
Memeory	96GB DDR3 1333MHz bandwidth:8GB/s
Network	Ethernet 1G bandwidth:943Mbits/s
Disk	SATA 1T bandwidth:154.82MB/s
OS	Ubuntu 16.04 and the kernel is 4.13.0-43-generic
GCC	4.3
Redis	4.2.5

Configurations of The Server Node

Experimental Methodology

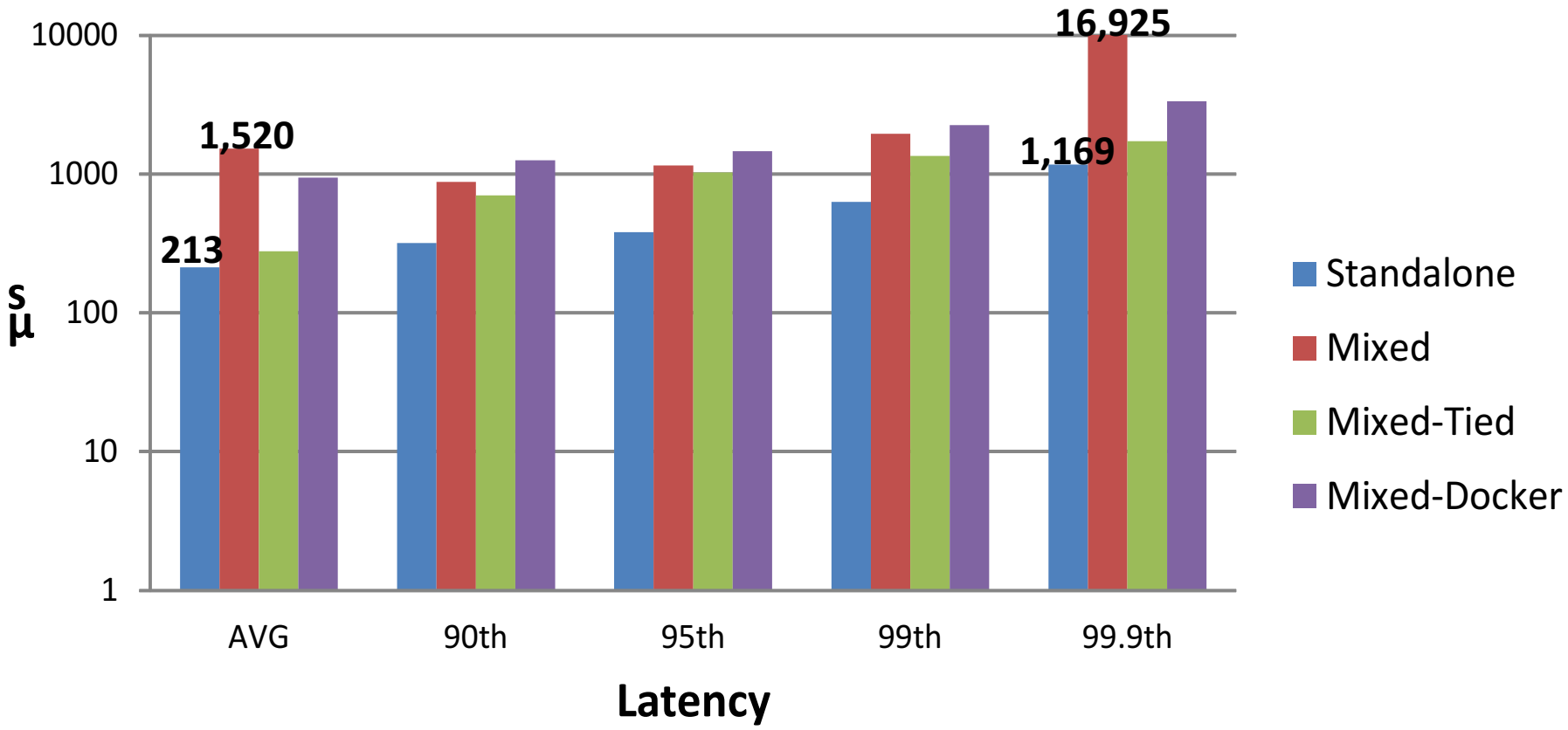
- **Four modes:**
  - Standalone
  - Mixed
  - Mixed-Tied
  - Mixed-Docker



Metrics

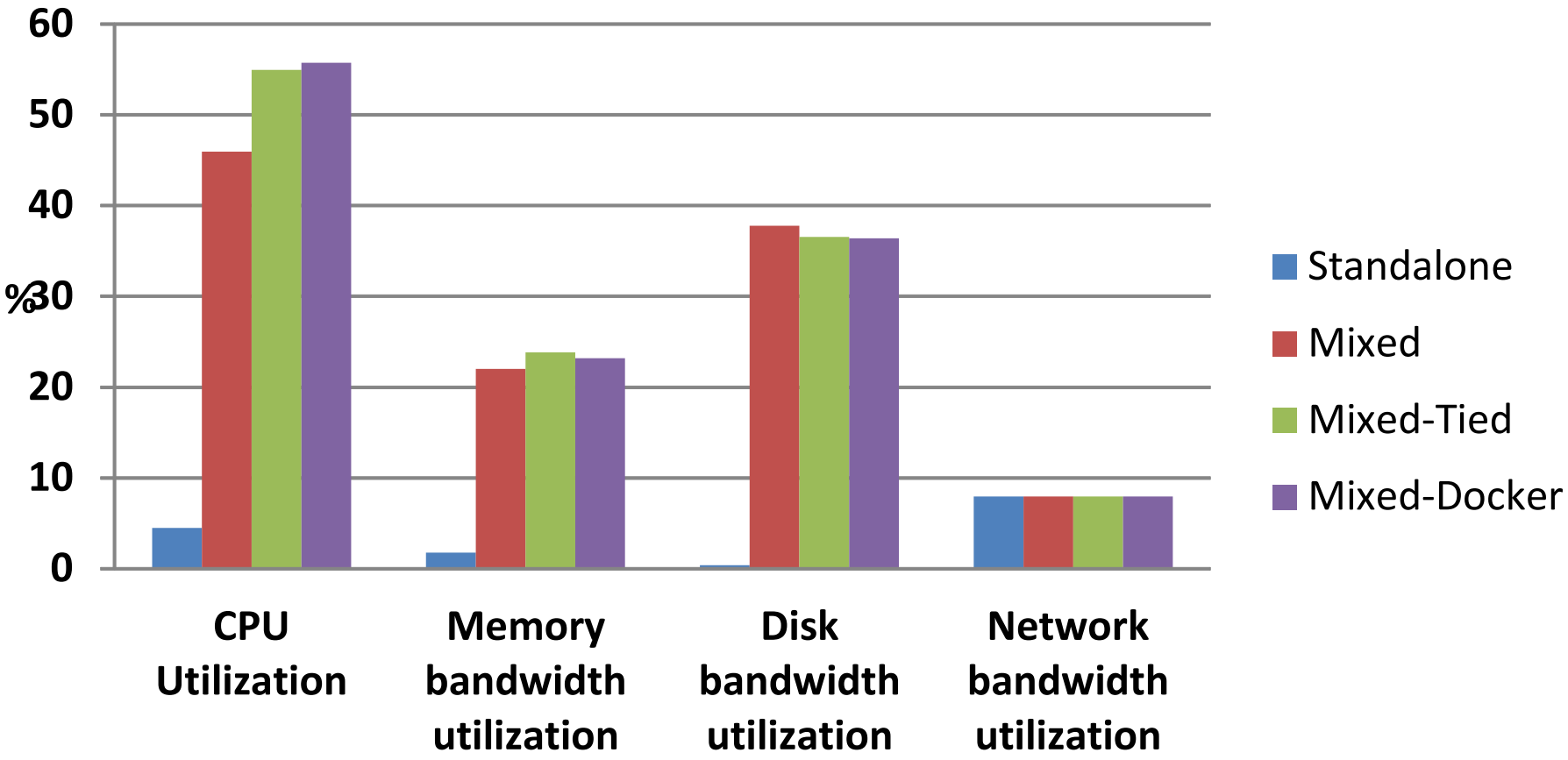
- User-observed metrics
  - Latency
- System level metrics
  - CPU utilization
  - Memory bandwidth utilization
  - Disk bandwidth utilization
  - Network bandwidth utilization
- Micro-architectural metrics
  - L1I Cache Miss
  - L2 Cache Miss
  - L3 Cache Miss
  - DTLB Cache Miss
  - ITLB Cache Miss
- The System entropy

Latency Behaviors



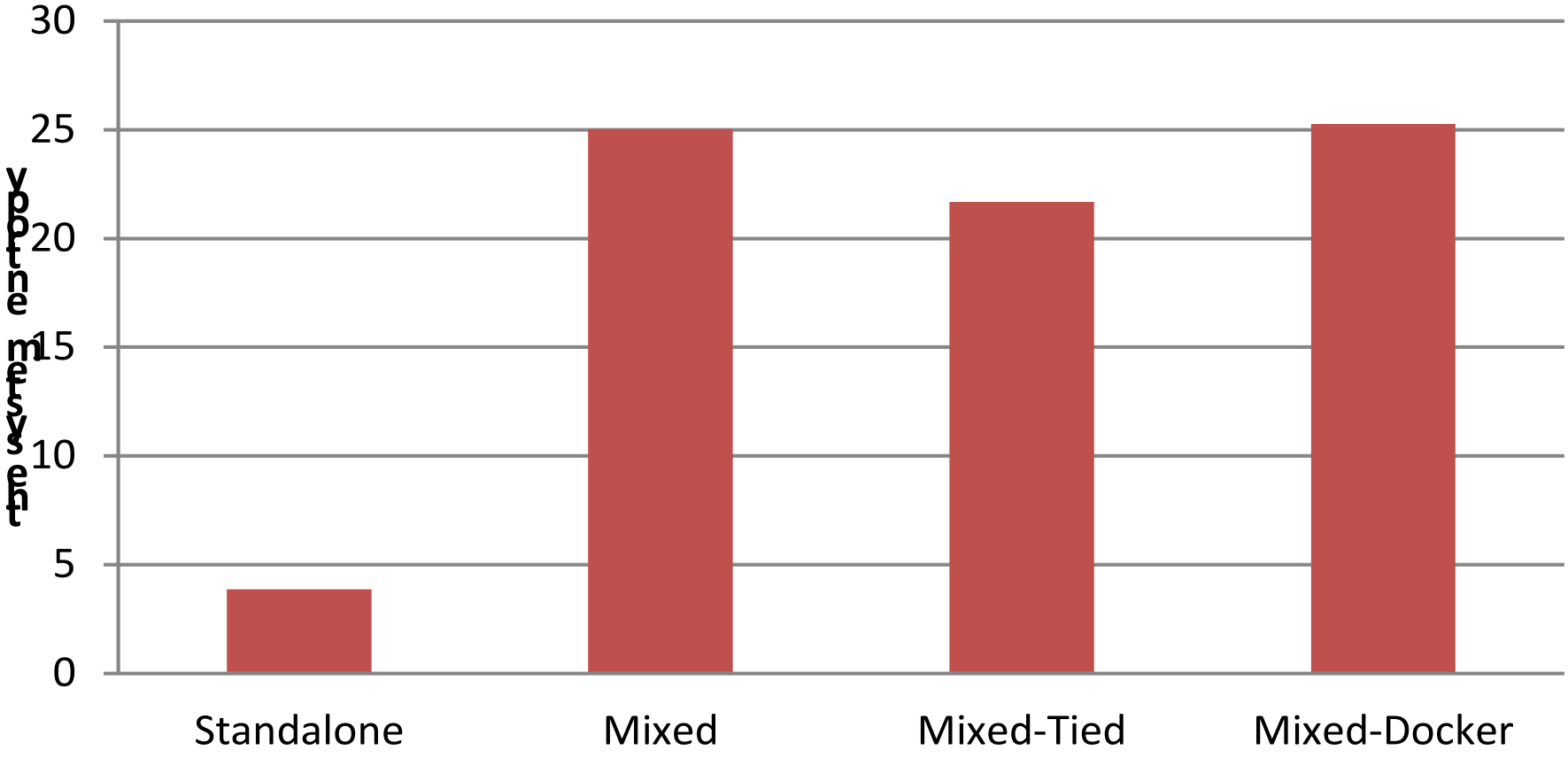
Co-locating deployment without any isolation mechanism incurs the high latency.

System Resource Utilizations



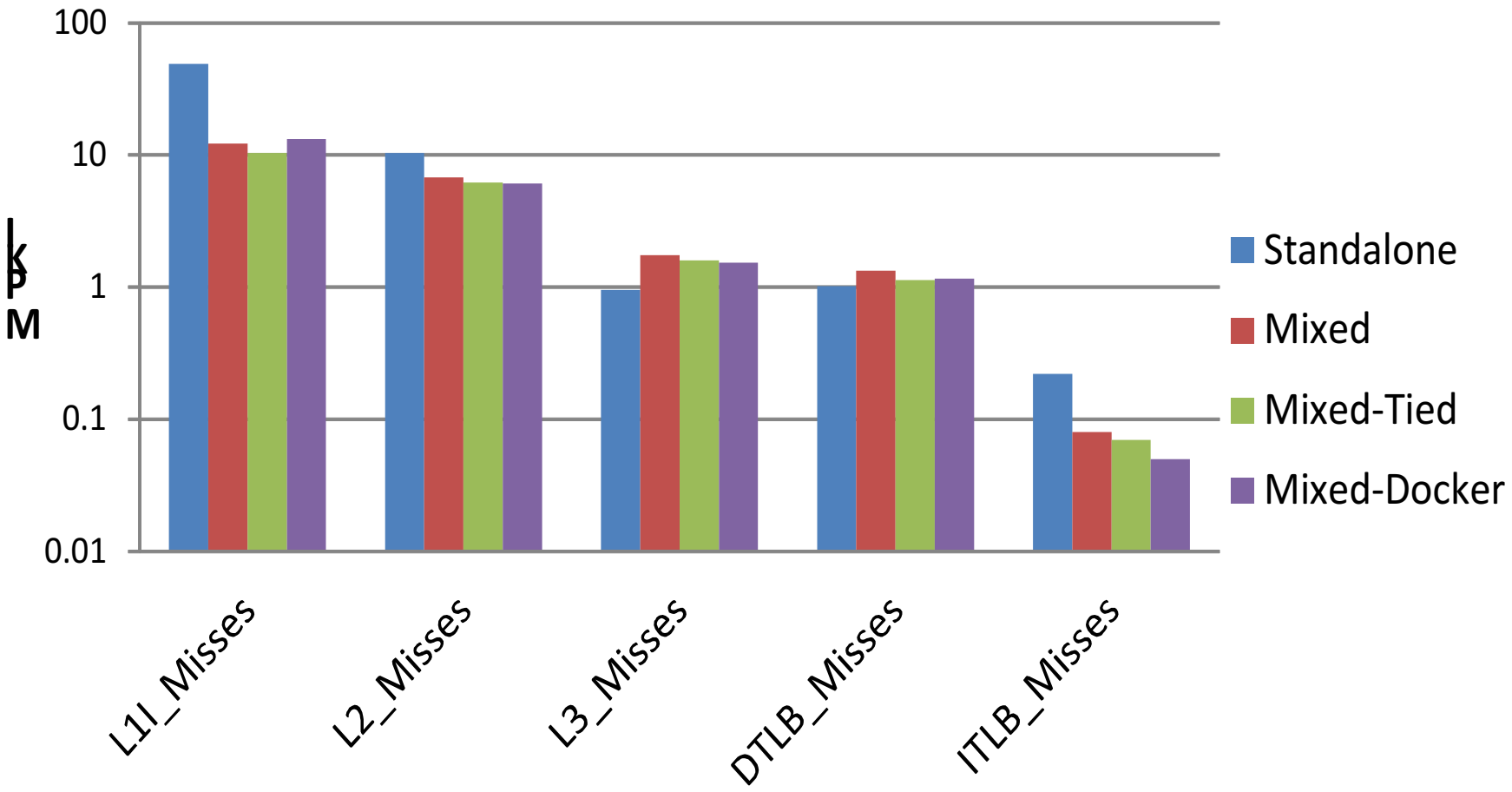
Co-locating deployment can prompt the resource utilizations.

The System Entropy



The entropy of mixed modes is much larger than that of the Standalone mode.

Micro-architecture Behaviors



The micro-architecture metrics can't reflect the system resource contention .

The Experiment Summary

- Co-location brings higher resource utilization and longer latency
  - 10 times resource utilization
  - 7 times the average latency & 14 times the 99.9th latency
- The system entropy can reflect the resource contention
  - System entropy in Standalone mode is 6 times that in Mixed mode
- Isolation mechanisms can reduce the interference between Co-locating workloads
  - CPU-affinity mechanism reduces interference.

Conclusion

- DCMIX
  - **Co-locating workloads for Cloud data center**
  - 17 typical cloud data center workloads, latencies of workloads range from microseconds to minutes. And mixture of serial execution and parallel execution
  - will open sourced soon
    - X86 version
    - RISC-V version


QUESTIONS
And
Answers