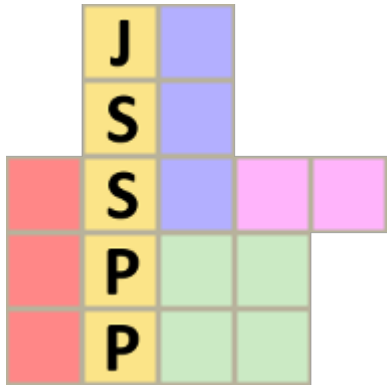


JSSPP 2020

23rd Workshop on Job Scheduling Strategies for Parallel Processing

Improving Resource Isolation of Critical Tasks in a Workload

Meghana Thiyyakat, Subramaniam Kalambur, Dinkar Sitaram

Centre for Cloud Computing and Big Data

PES University

Bangalore, India

Centre for Cloud Computing and Big Data

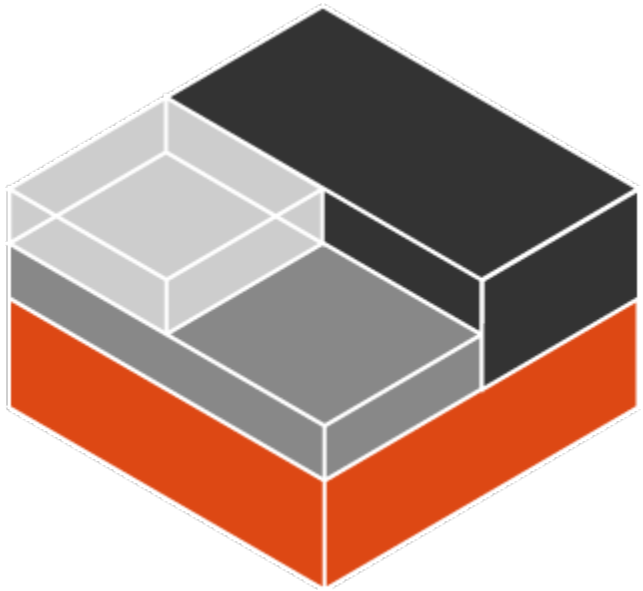
PES UNIVERSITY

Objective

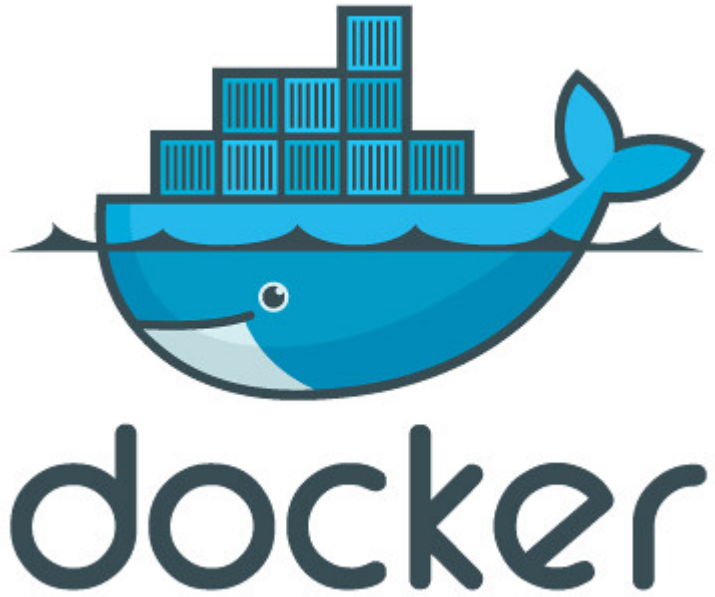
To provide robust resource isolation to critical tasks in heterogeneous workloads

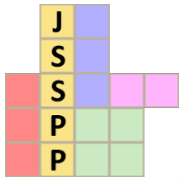
Colocation of Tasks

- Problems with colocation: security and resource contention
- Solution?



Containers





Containers

PES UNIVERSITY

Cgroups – quotas and shares

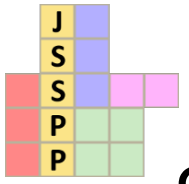
Centre for Cloud Computing and Big Data

Cgroups

- CPU shares

$$\frac{group\_p\_shares}{group\_p\_shares + root\_shares} = share\%$$

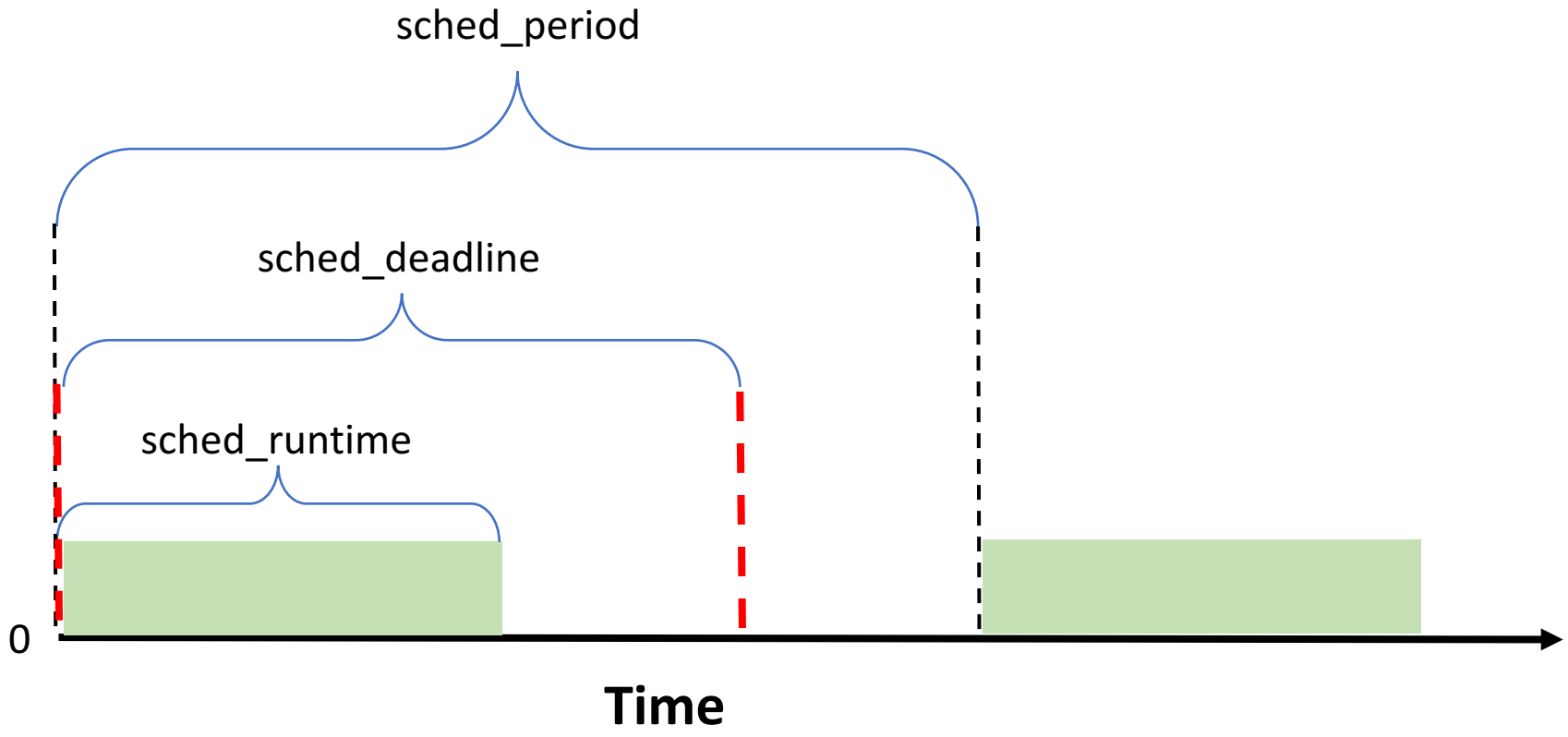
Example: 19456 / (19456+1024) = 19456 / 20480 = 0.95 = 95%



SCHED_DEADLINE

Based on Constant Bandwidth Server, and Earliest Deadline First.

Scheduling with SCHED_DEADLINE

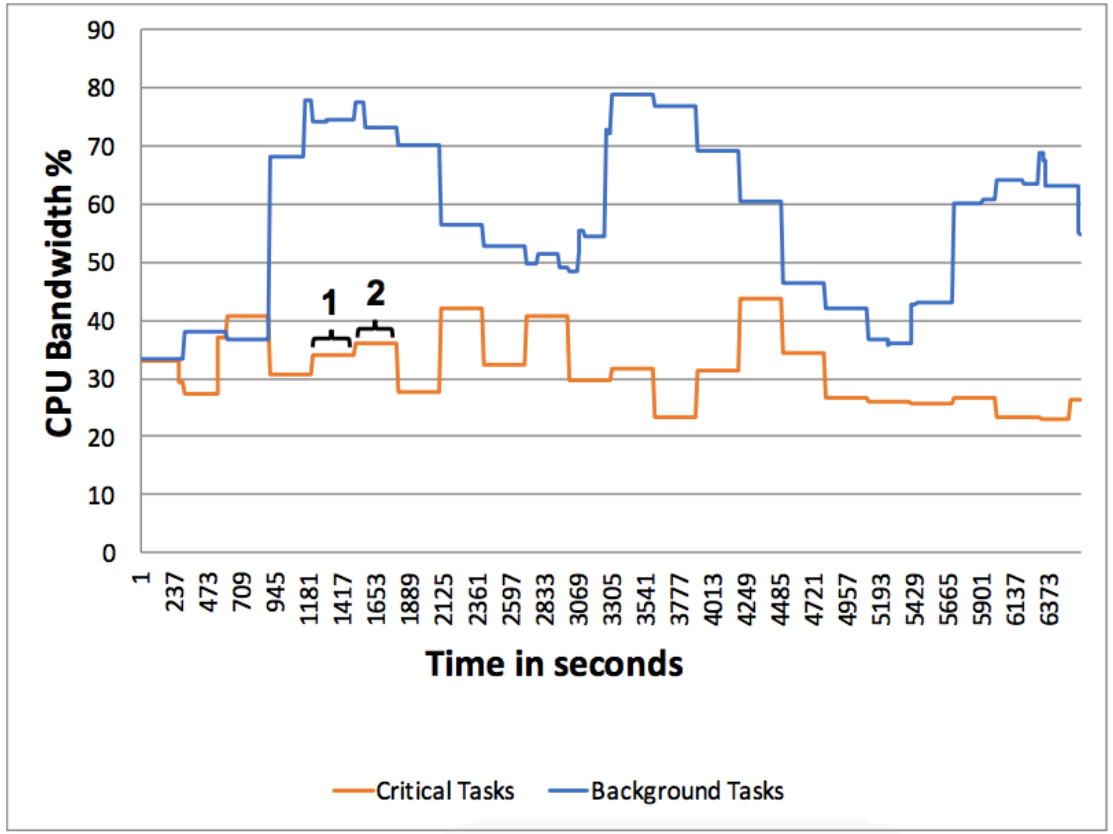


SCHED_DEADLINE

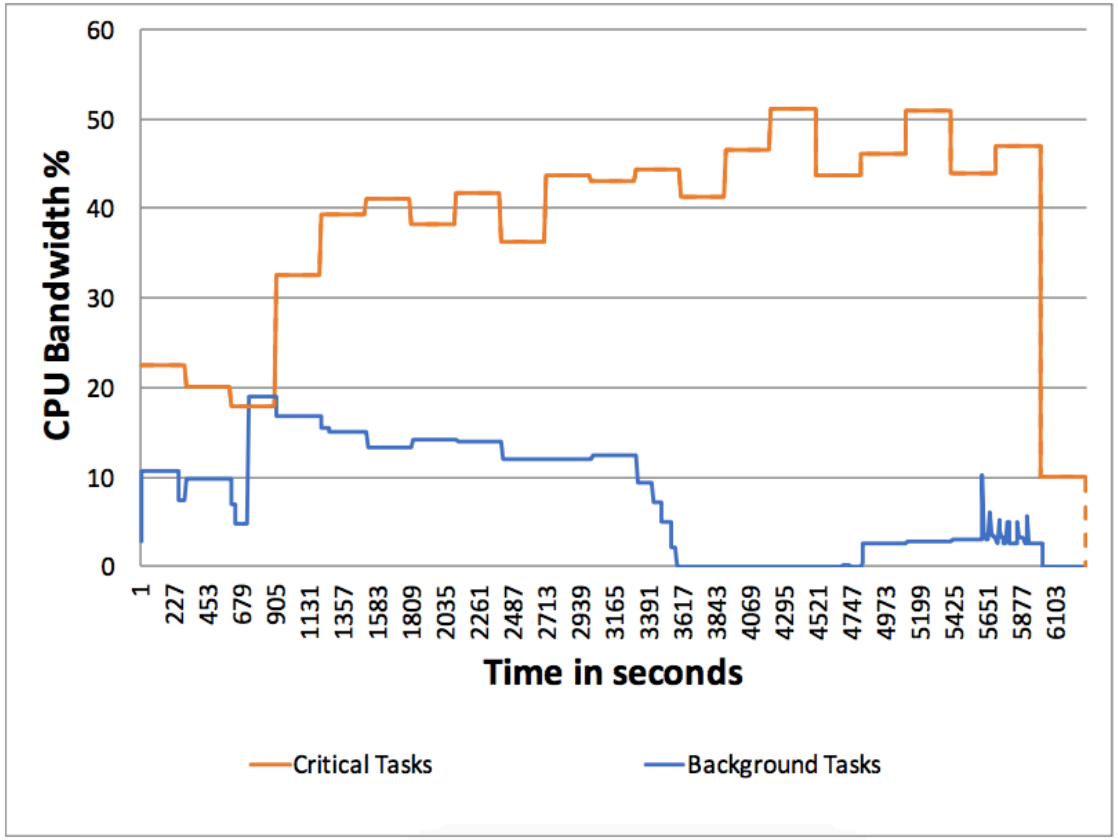
$$\sum_{i=1}^{n} \frac{sched\_runtime}{sched\_period} <= N \times rt\_quota\%$$

Workload- Google Cluster Trace

- High CPU Utilization Workload

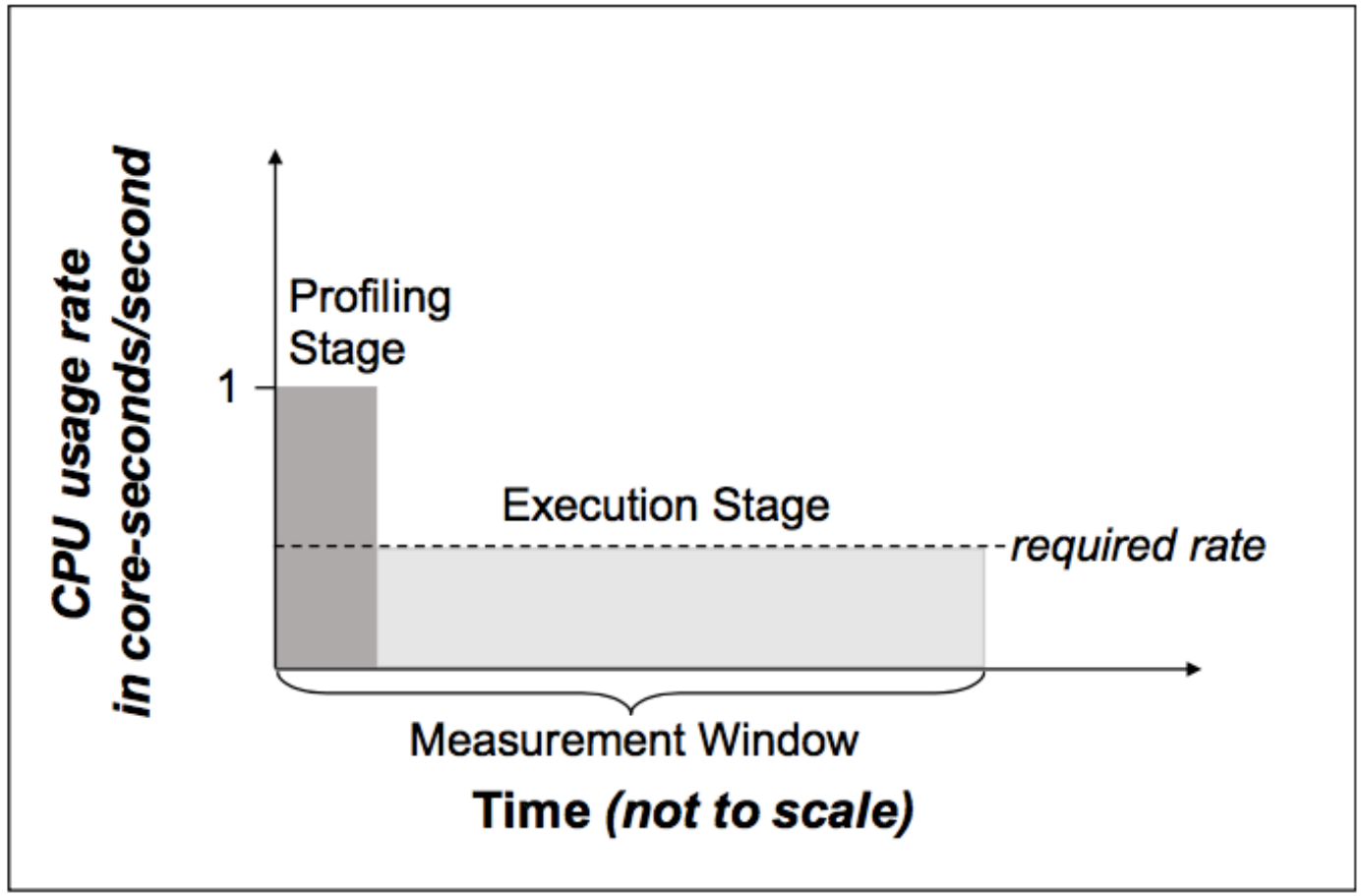


- Low CPU Utilization Workload



Scheduling with SCHED_DEADLINE

- Parameter estimation



Experiments

- Cgroups: 95% CPU shares, 99% CPU shares
- SCHED_DEADLINE: upper limit- 95%

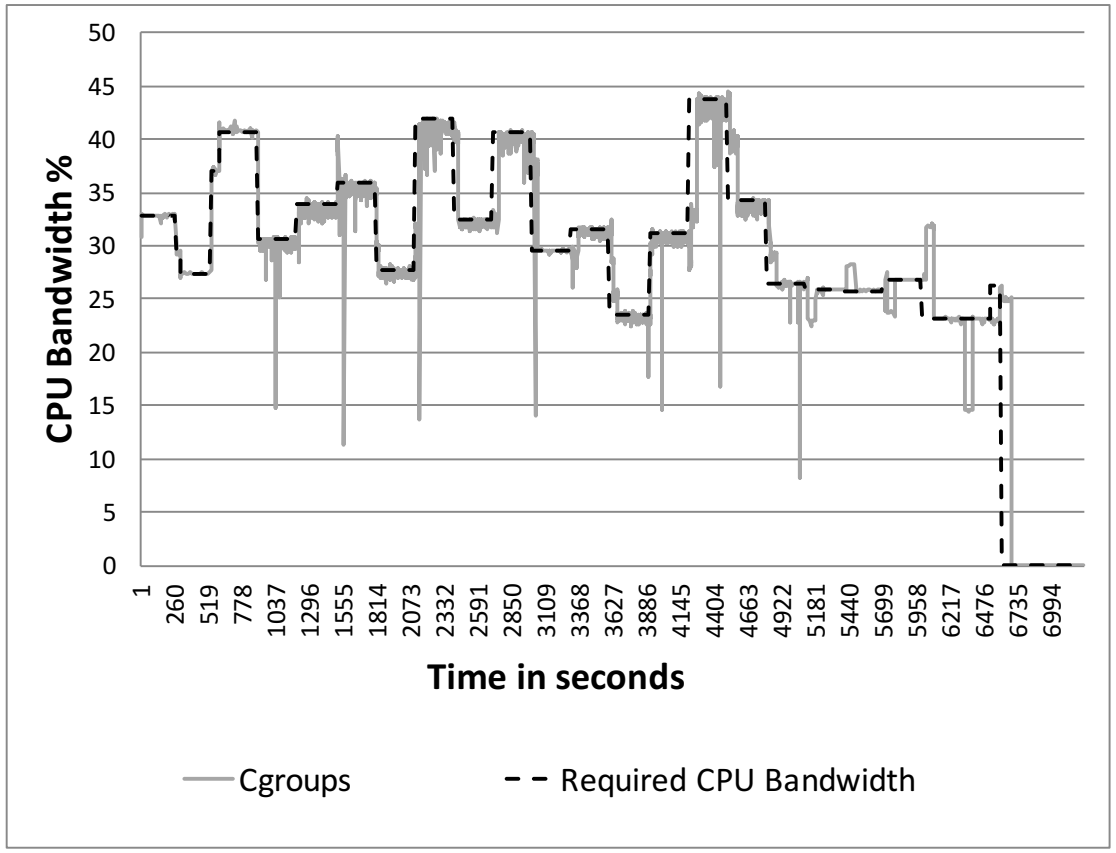
Type	CPU Speed GHz	Logical Cores or vCPUs	Linux Kernel Version
EC2 Instance- c5n.xlarge	3.0	4	4.15
EC2 Instance- c4.2xlarge	2.6	8	4.15
Intel Xeon E3-1220	3.1	4	5.5
Intel Xeon E5-2683 v4 (with HT)	2.1	32	5.5

Results

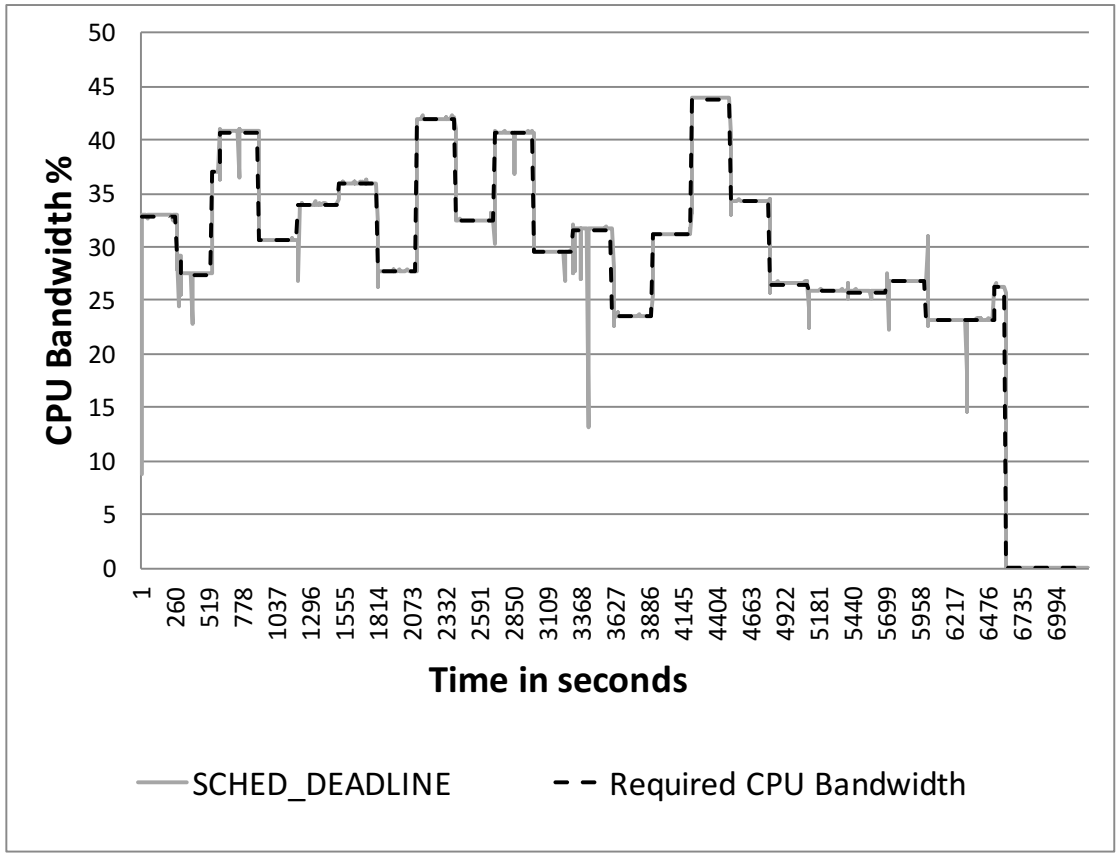
1. CPU Bandwidth Allocation of Critical Tasks
2. Fluctuation in allocated bandwidth
3. Delay in Tasks
4. Delay vs CPU Utilization

1a. CPU Bandwidth Allocation of Critical Tasks

- HCU Workload on 4-core Intel Xeon physical machine



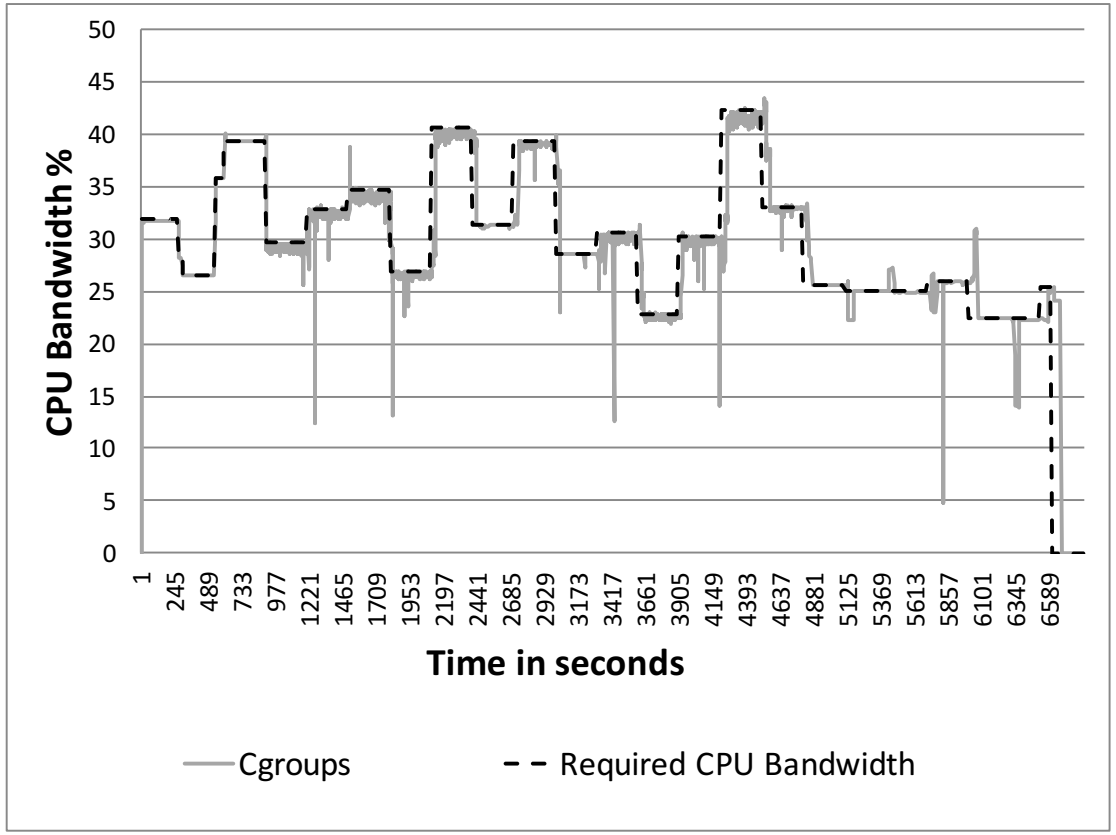
Cgroup with 95% shares



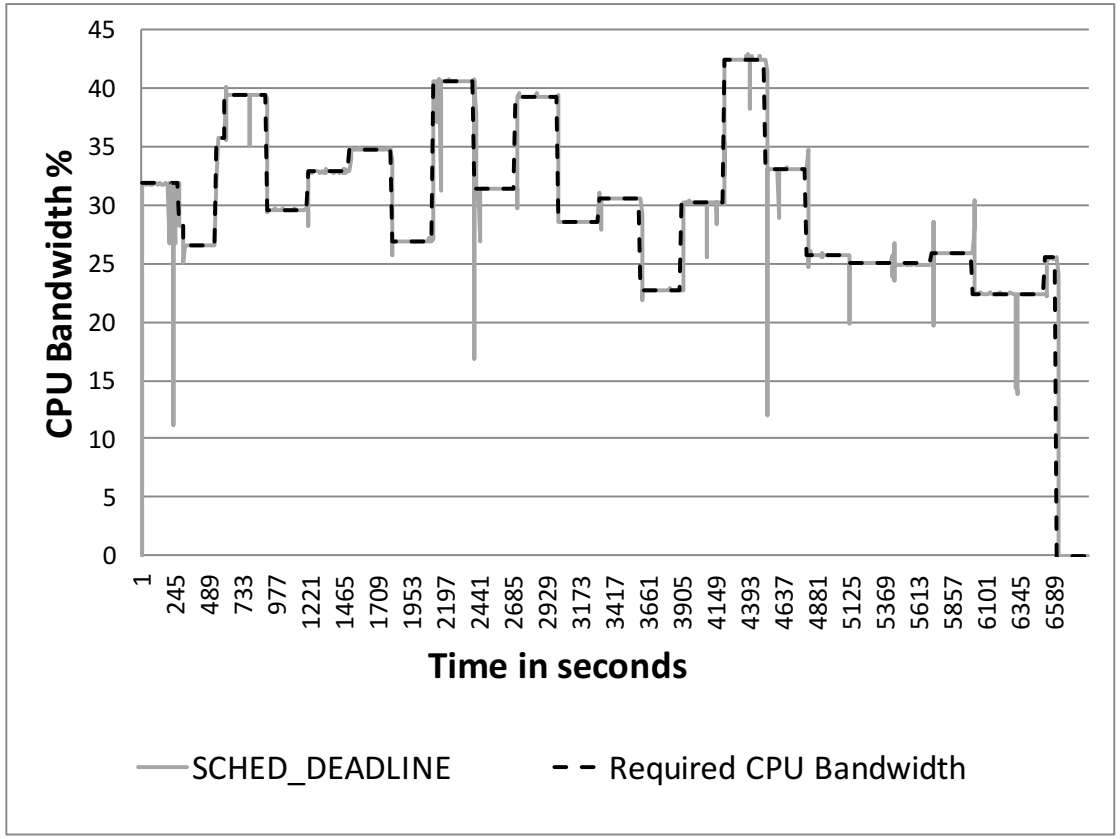
SCHED_DEADLINE

1b. CPU Bandwidth Allocation of Critical Tasks

- HCU Workload on c4.2xlarge AWS EC2 instance



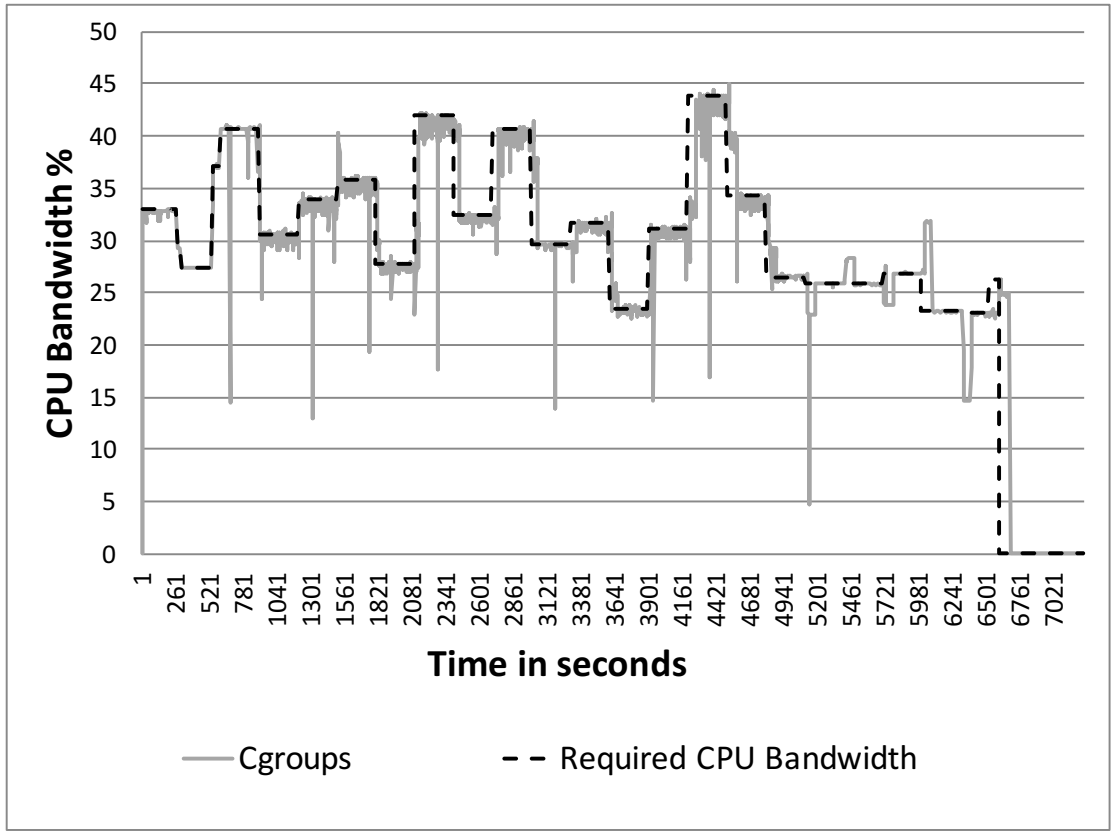
Cgroup with 95% shares



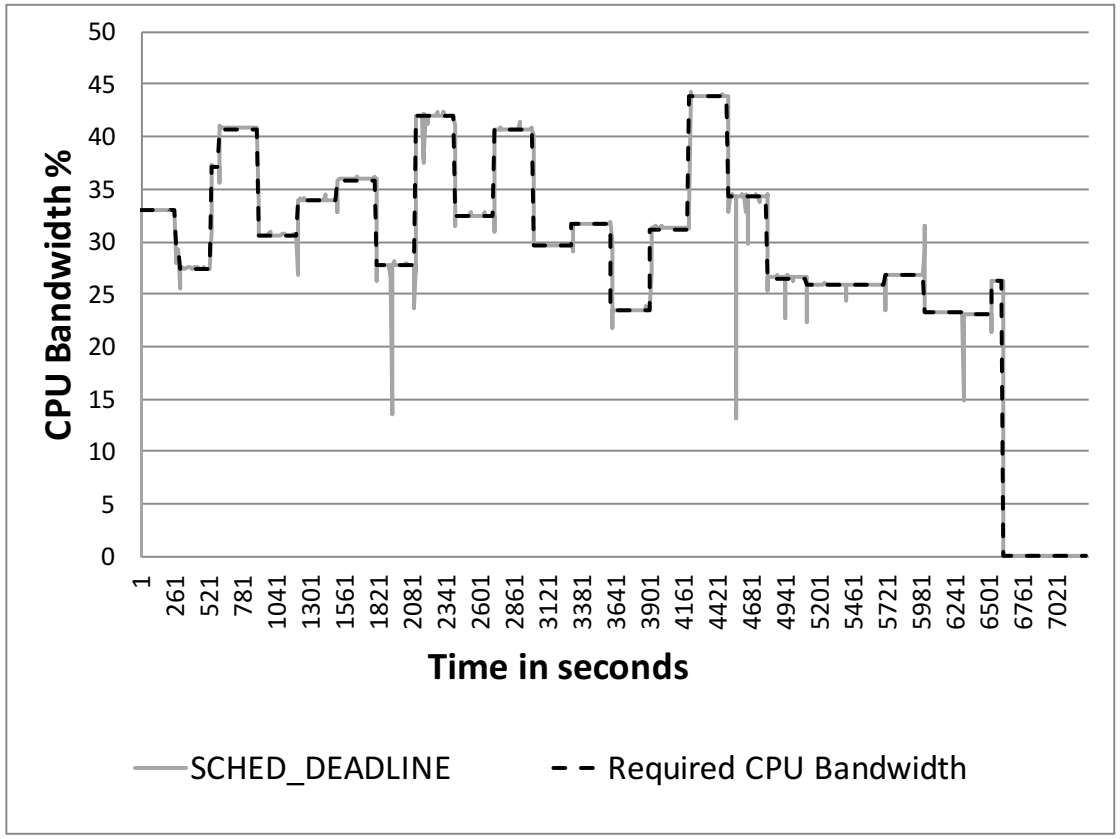
SCHED_DEADLINE

1c. CPU Bandwidth Allocation of Critical Tasks

- HCU Workload on c5n.xlarge AWS EC2 instance



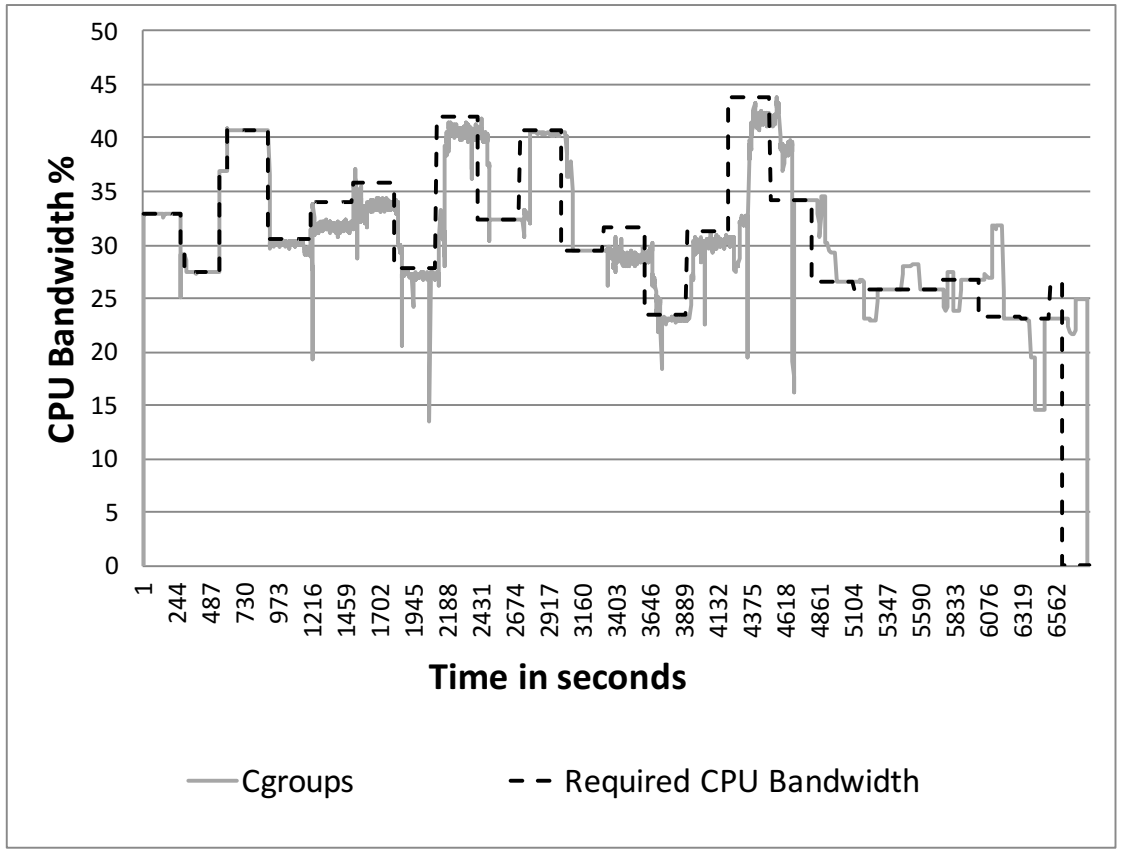
Cgroup with 95% shares



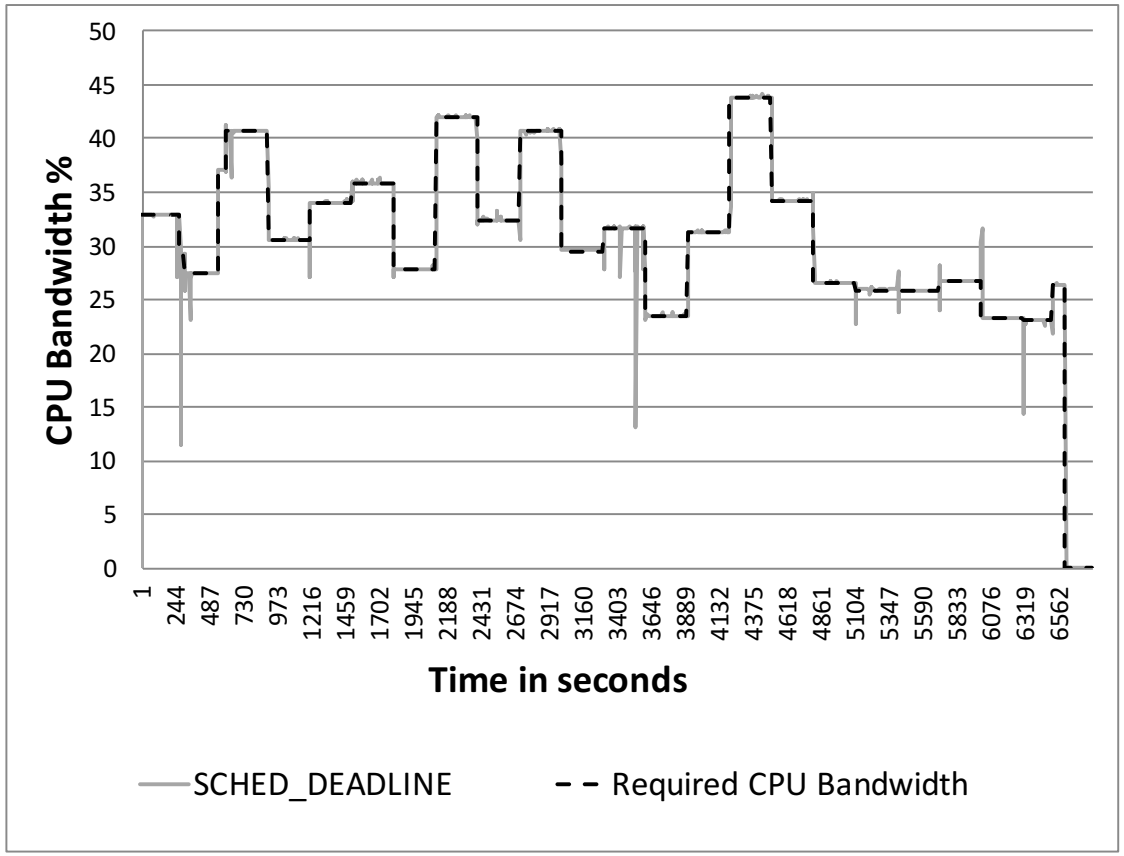
SCHED_DEADLINE

1d. CPU Bandwidth Allocation of Critical Tasks

- HCU Workload on 32-core Intel Xeon physical machine



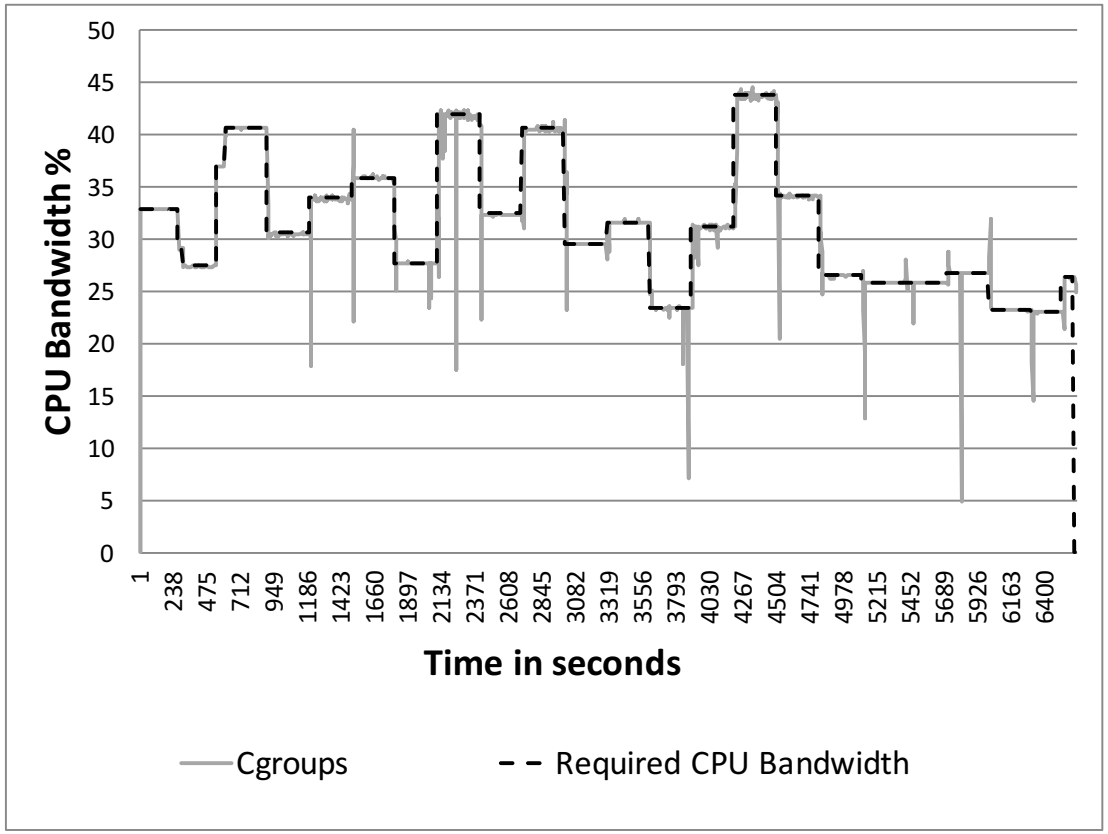
Cgroup with 95% shares



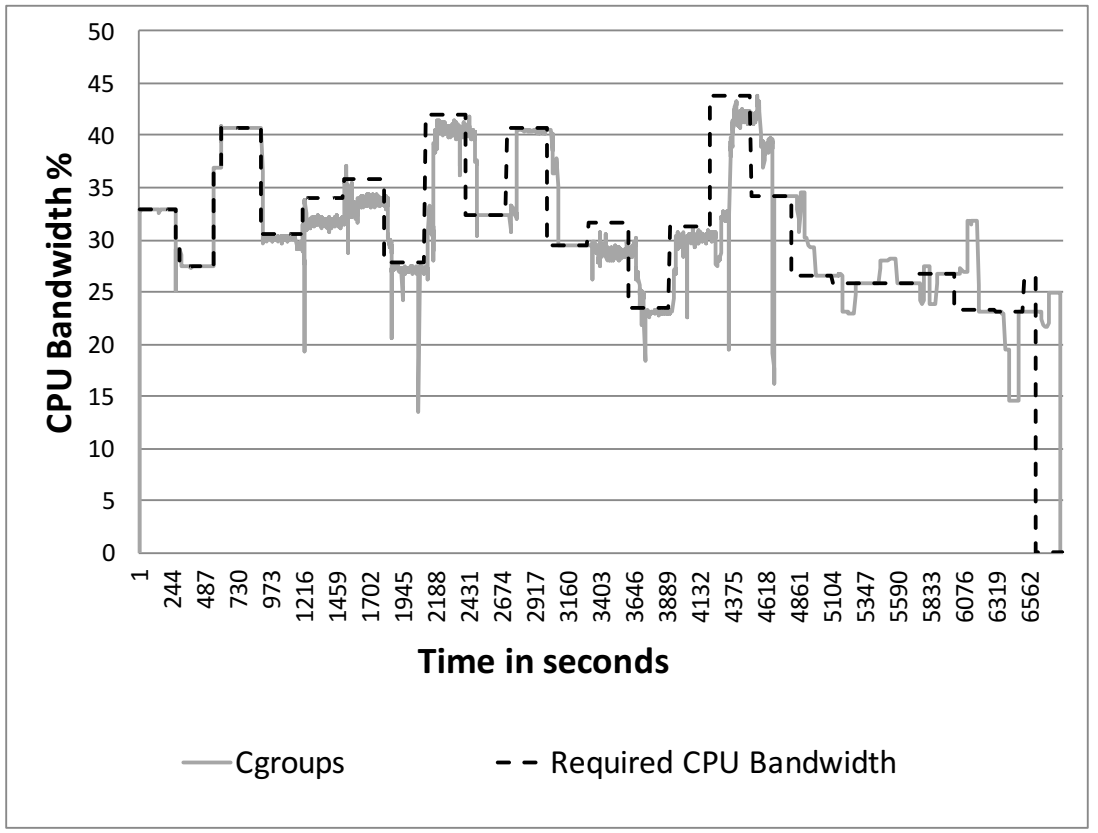
SCHED_DEADLINE

1e. CPU Bandwidth Allocation of Critical Tasks

- HCU Workload on 32-core Intel Xeon machine



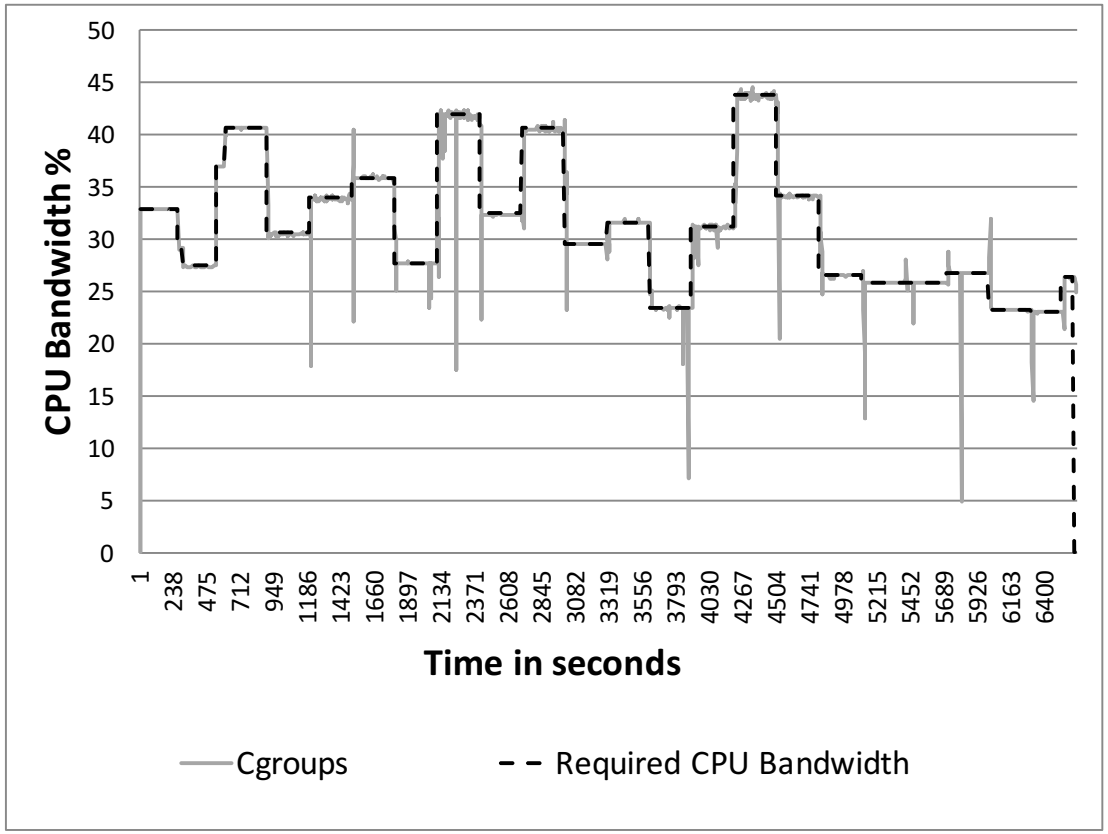
Cgroup with 99% shares



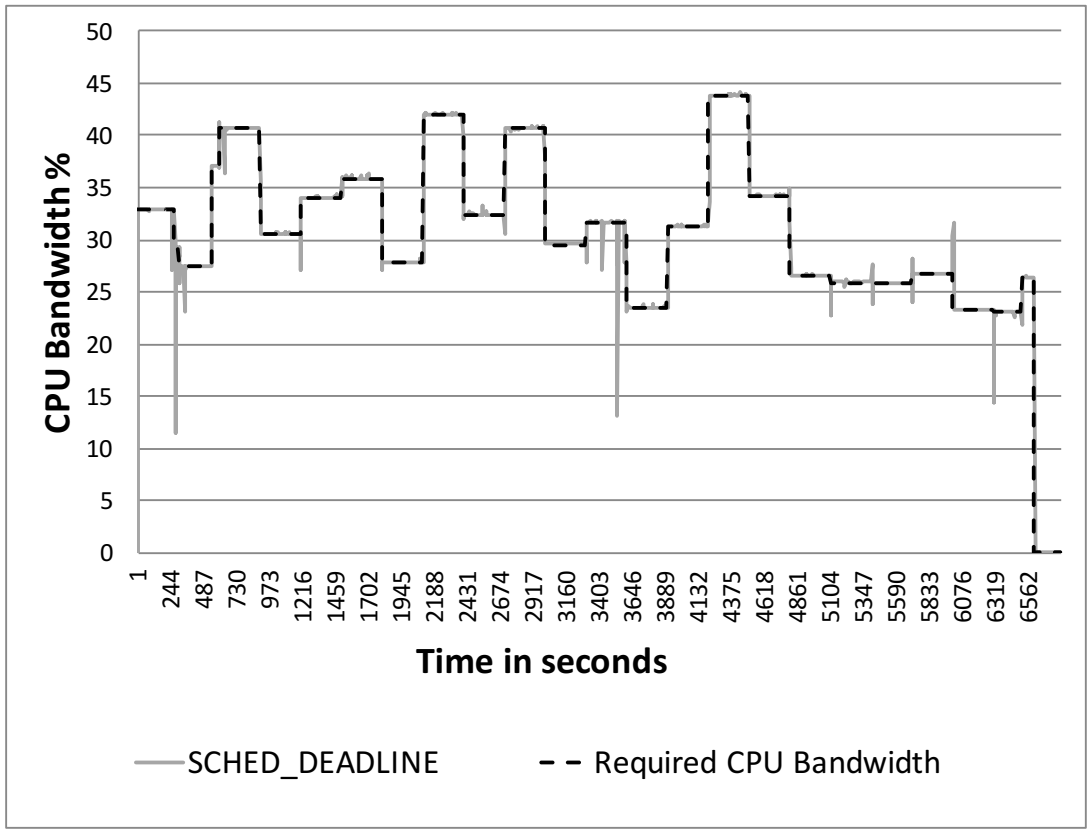
Cgroup with 95% shares

1e. CPU Bandwidth Allocation of Critical Tasks

- HCU Workload on 32-core Intel Xeon machine



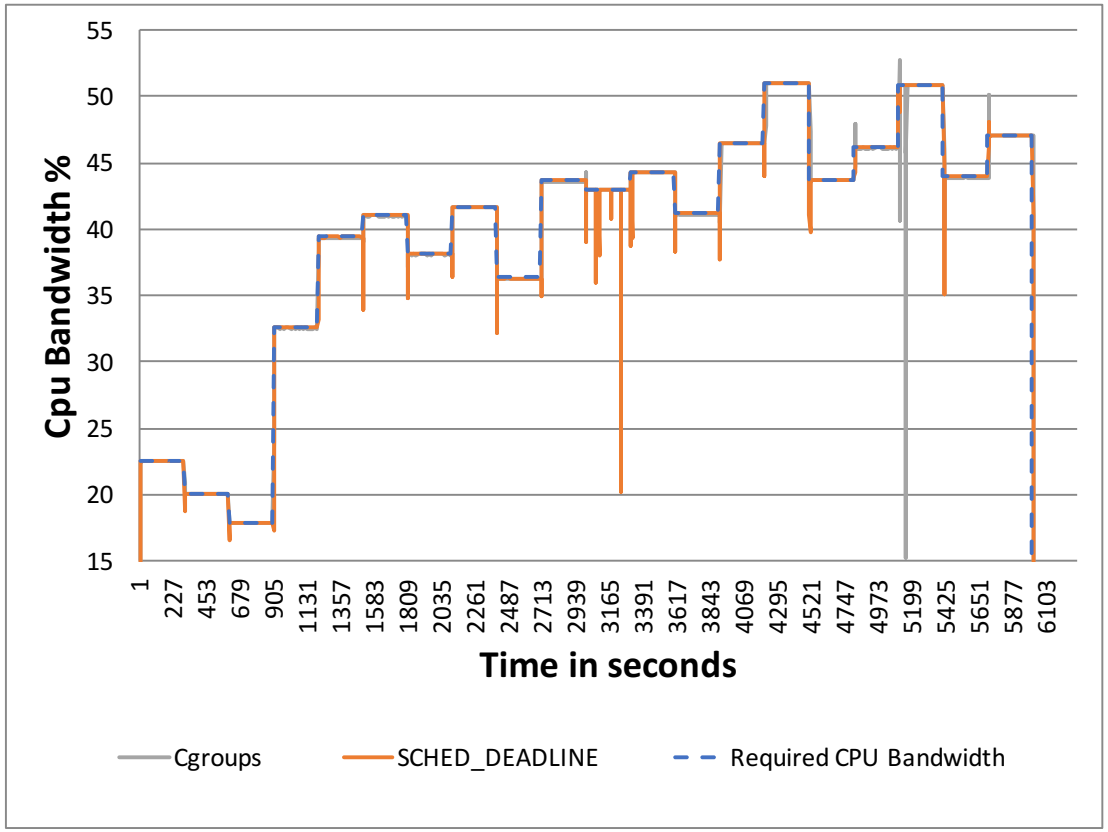
Cgroup with 99% shares



SCHED_DEADLINE

1f. CPU Bandwidth Allocation of Critical Tasks

- LCU Workload on 32-core Intel Xeon machine



Distribution of CPU bandwidth allocated

	SCHED_DEADLINE	Cgroups
1st Quartile	36.317	36.282
Median	41.6641	41.6143
3rd Quartile	44.283	44.246

2a. Fluctuation in bandwidth allocated

- Standard deviation of CPU bandwidth % allocated in intervals 1 & 2.

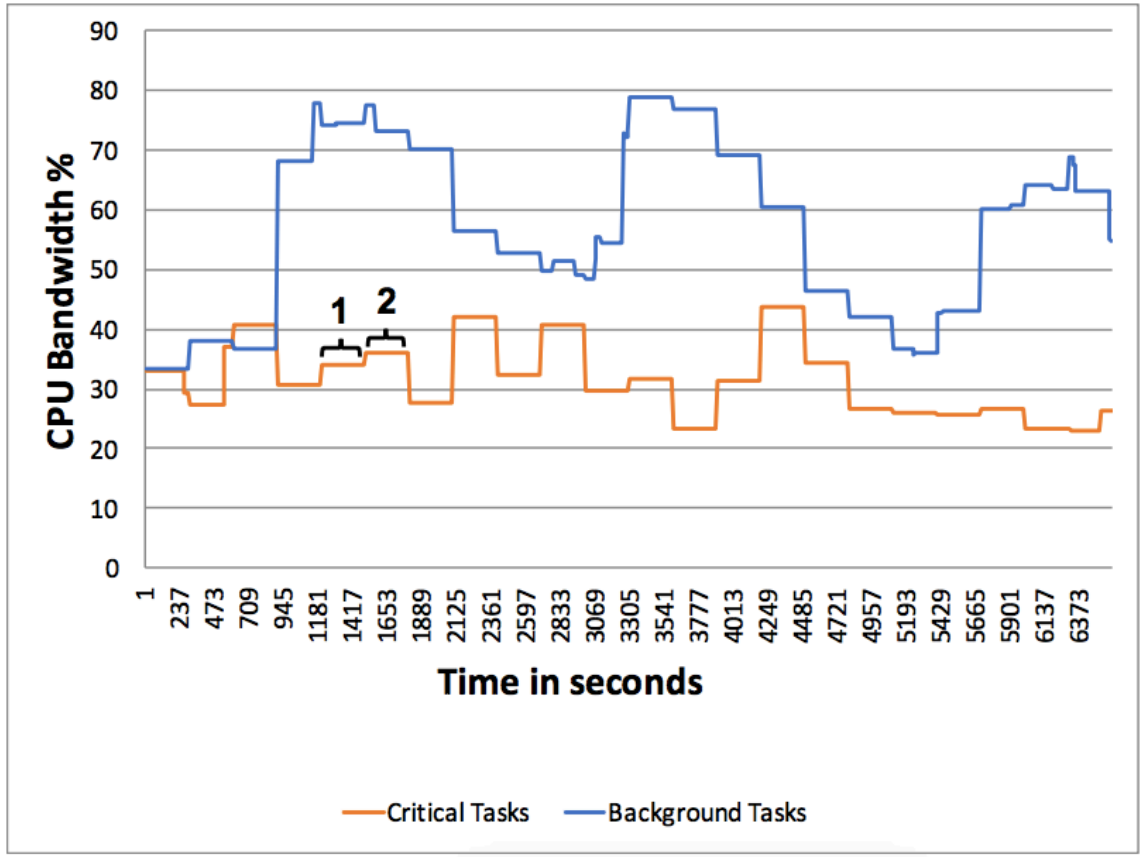
(a) Interval 1

Server	SCHED_DEADLINE	Cgroups
4-core Physical Machine	0.032	0.422
32-core Physical Machine	0.031	1.235
EC2 Instance- c4.2xlarge	0.026	1.409
EC2 Instance- c5n.xlarge	0.049	1.518

(b) Interval 2

Server	SCHED_DEADLINE	Cgroups
4-core Physical Machine	0.044	1.745
32-core Physical Machine	0.054	0.578
EC2 Instance- c4.2xlarge	0.027	0.805
EC2 Instance- c5n.xlarge	0.028	1.120

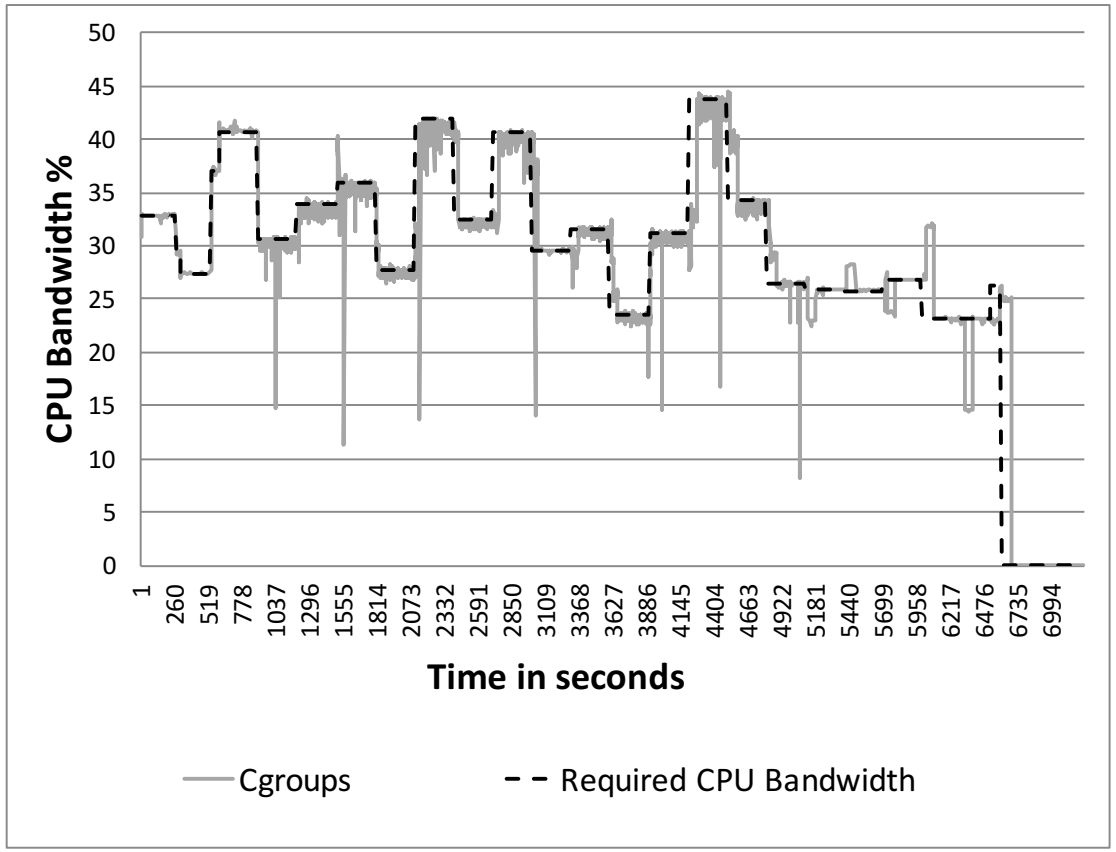
2a. Fluctuation in bandwidth allocated



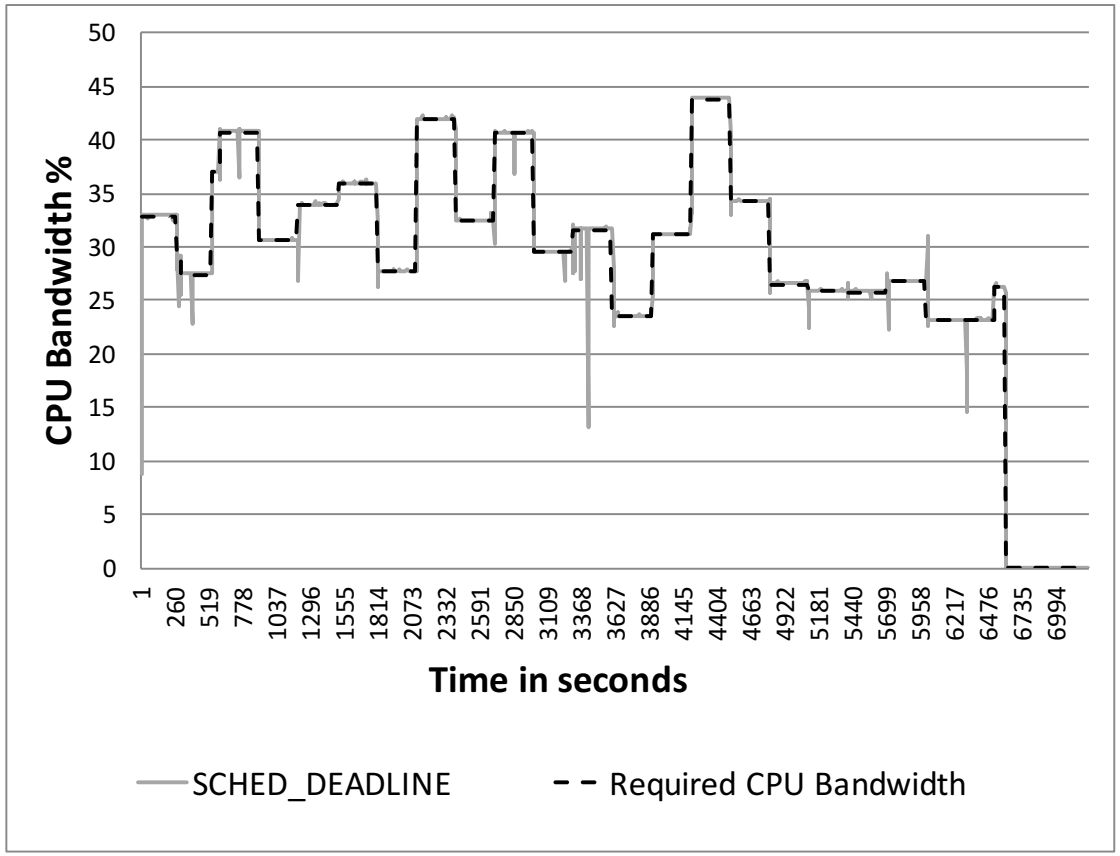
CPU Bandwidth Requirement of HCU Workload

2a. Fluctuation in bandwidth allocated

- HCU Workload on 4-core Intel Xeon physical machine



Cgroup with 95% shares



SCHED_DEADLINE

2a. Fluctuation in bandwidth allocated

- Standard deviation of CPU bandwidth % allocated in intervals 1 & 2.

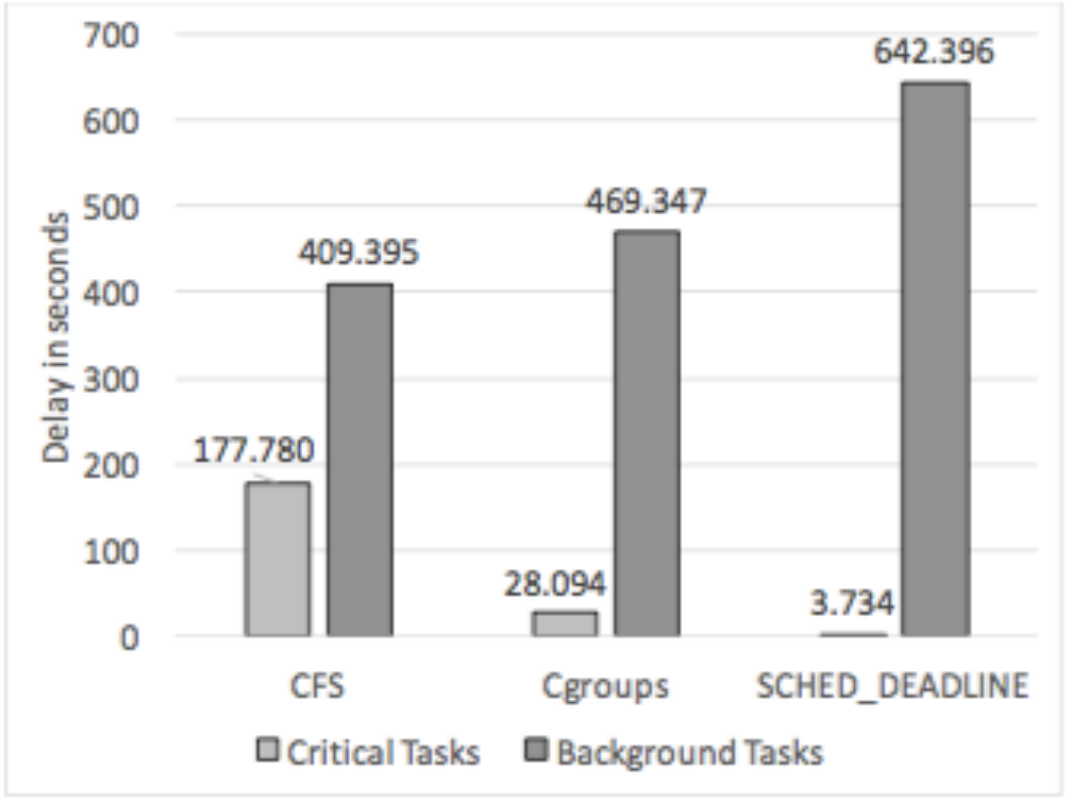
(a) Interval 1

Server	SCHED_DEADLINE	Cgroups
4-core Physical Machine	0.032	0.422
32-core Physical Machine	0.031	1.235
EC2 Instance- c4.2xlarge	0.026	1.409
EC2 Instance- c5n.xlarge	0.049	1.518

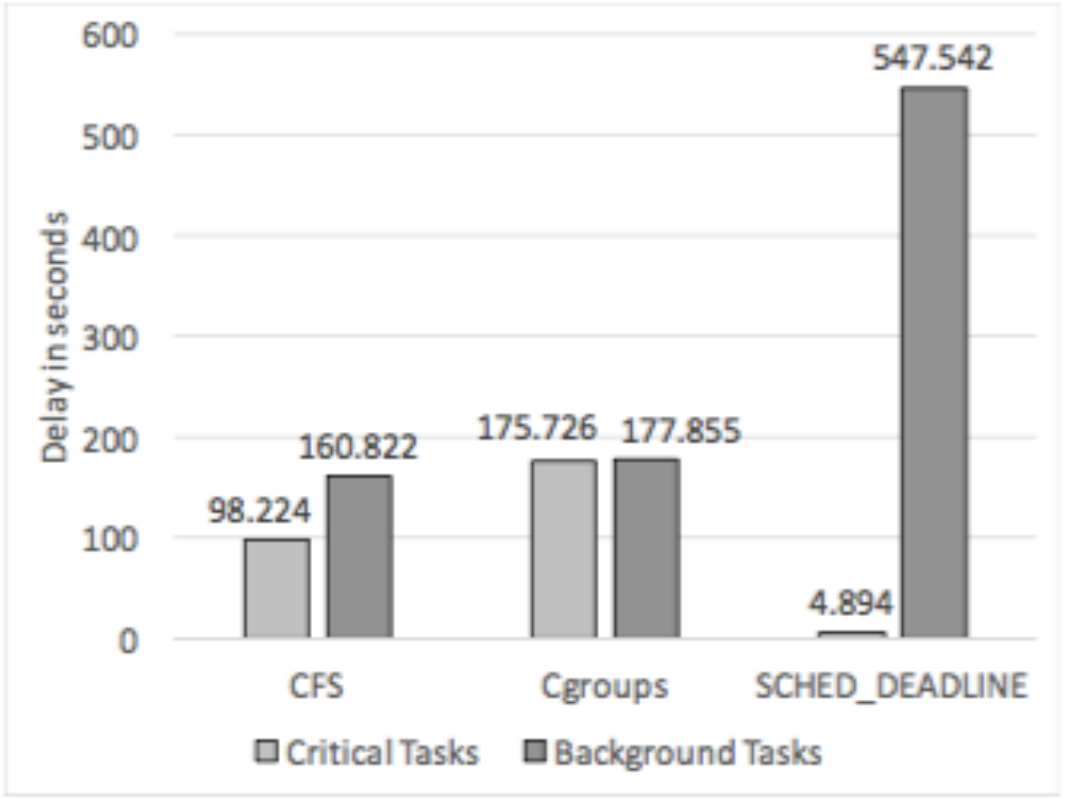
(b) Interval 2

Server	SCHED_DEADLINE	Cgroups
4-core Physical Machine	0.044	1.745
32-core Physical Machine	0.054	0.578
EC2 Instance- c4.2xlarge	0.027	0.805
EC2 Instance- c5n.xlarge	0.028	1.120

3a. Average delay in tasks

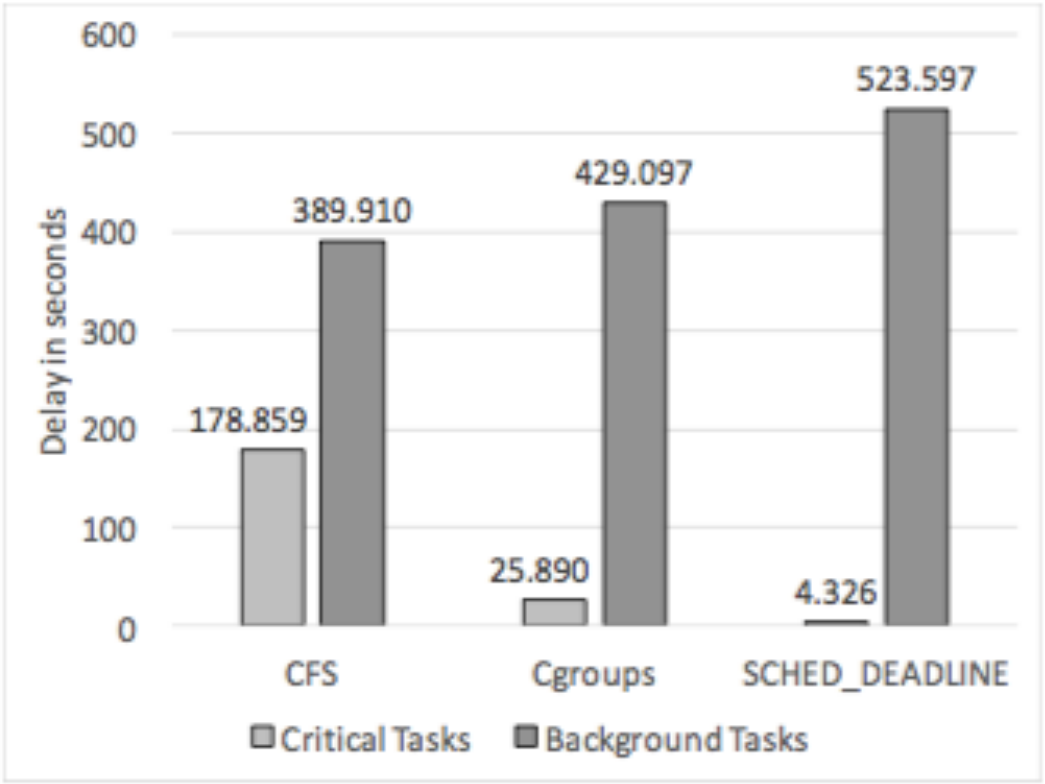


4-core Intel Xeon machine

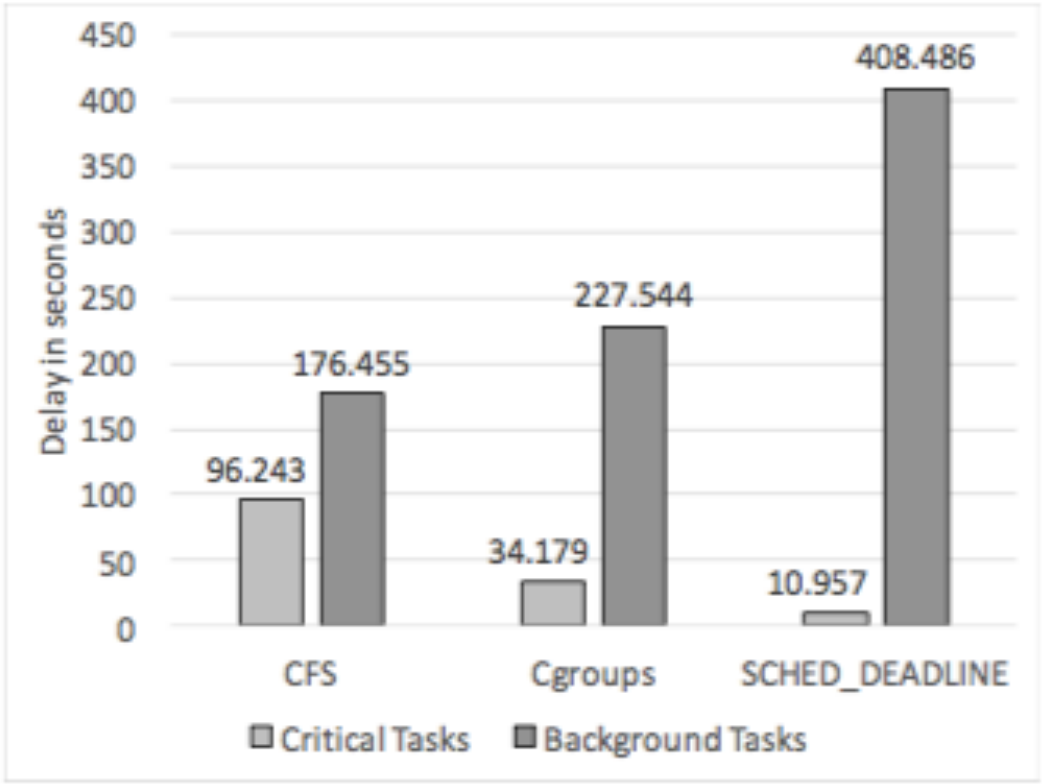


32-core Intel Xeon machine

3b. Average delay in tasks

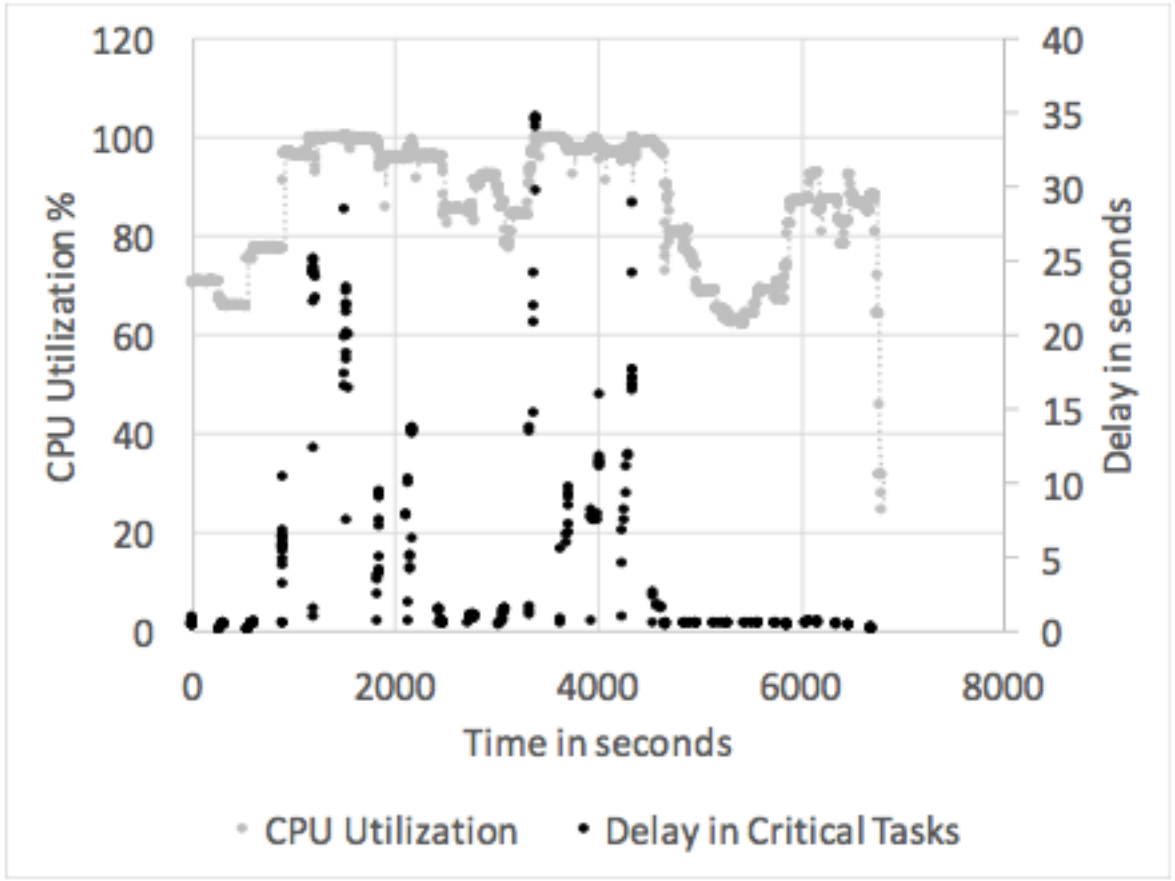


c5n.xlarge instance

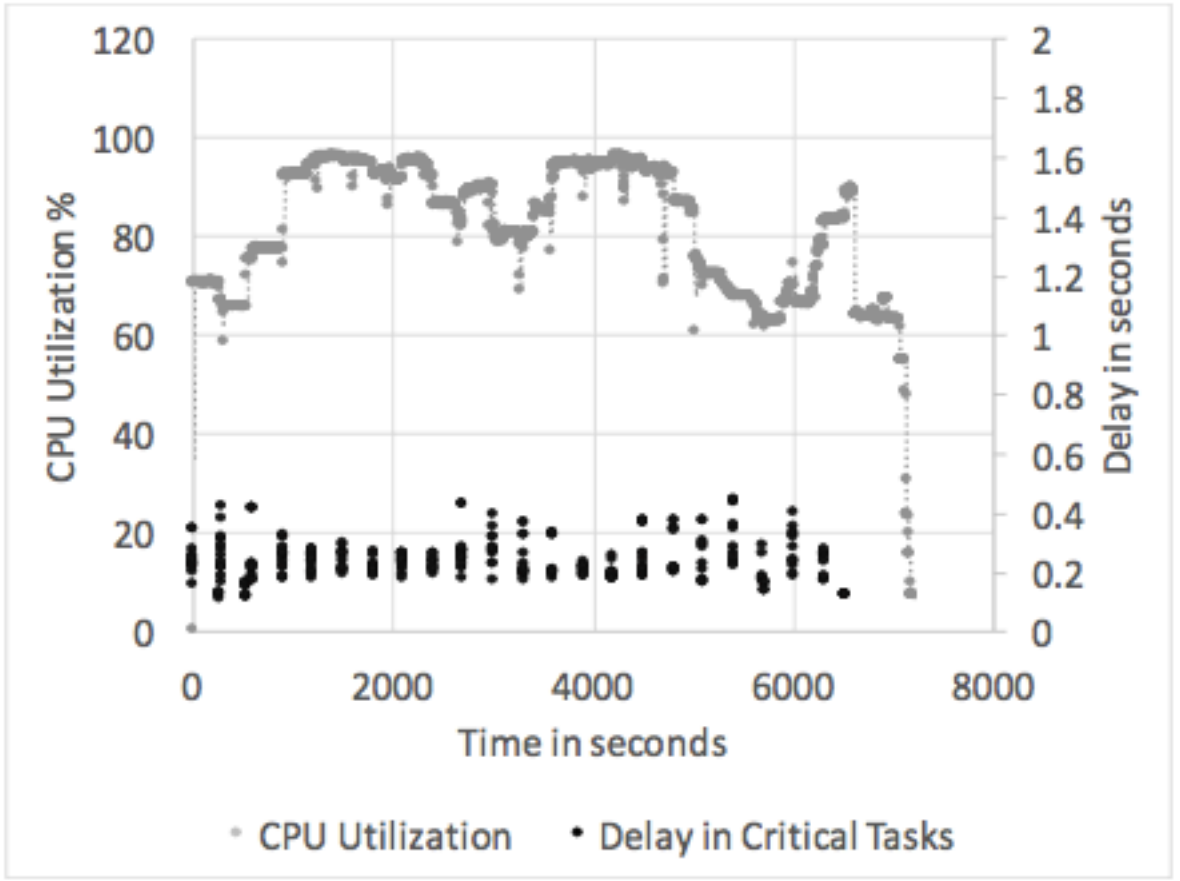


c4.2xlarge instance

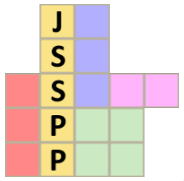
4. Delay versus CPU Utilization



Cgroups

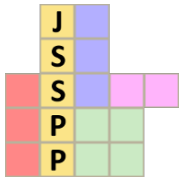


SCHED_DEADLINE



Conclusion

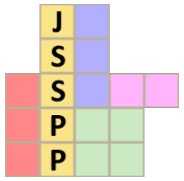
- Resource isolation provided by cgroups – susceptible to interference
- Unstable bandwidth allocation leads to delays in tasks.
- Use of SCHED_DEADLINE for CPU bandwidth allocation of critical tasks--- offers better resource isolation.
- Average delays under SCHED_DEADLINE are 3x-40x smaller than under cgroups.
- Steady bandwidth allocation – immune to load on the CPU



PES UNIVERSITY

Q&A

Centre for Cloud Computing and Big Data



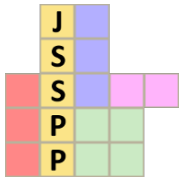
Thank You!

Contact info:

meghana.thiyyakat@pesu.pes.edu

subramaniamkv@pes.edu

dinkars@pes.edu



Additional Slides

Contact info:

meghana.thiyyakat@pesu.pes.edu

subramaniamkv@pes.edu

dinkars@pes.edu

Workload Generation

- Assume chosen machine A with 2.1GHz clock speed, 4 cores.

$$
\begin{aligned}
ClockCycles &= CPU\,RuntimeA \times ClockSpeedA \\
&= CPU\,RuntimeB \times ClockSpeedB \\
\therefore CPU\,RuntimeB &= \frac{CPU\,RuntimeA \times ClockSpeedA}{ClockSpeedB}
\end{aligned}
$$

- Task thread executed a few mathematical statements for *CPUSlice* seconds and was then put to sleep for *SleepSlice* seconds.
- The task thread repeated these steps until the total CPU Time consumed by it equalled CPURuntimeB

Table 1: Description of the HCU workload tasks on the 4-core machine.

Task No	Type	Mean CPU Usage Rate [Std Dev] (in core-second/second)	Duration (in s)
1	critical	0.186 [0.024]	5700
2	critical	0.179 [0.018]	5700
3	critical	0.162 [0.019]	6300
4	critical	0.039 [0.042]	6600
5	critical	0.030 [0.025]	6600
6	critical	0.042 [0.040]	6600
7	critical	0.705 [0.154]	6600
8	background	1.398 [0.266]	6600
9	background	0.247 [0.085]	6600
10	background	0.239 [0.090]	6600
11	background	0.231 [0.120]	6600
12	background	0.261 [0.124]	5438
13	background	0.247 [0.094]	6600
14	background	0.001 [0.0]	246

Table 3: Mean Absolute Percentage Error and Standard Error of the estimated `sched_runtime` parameter in the HCU workload

Server	Mean Absolute Percentage Error	Standard Error (in core-second/second)
EC2 Instance- c5n.xlarge	4.694%	0.043
EC2 Instance- c4.2xlarge	3.726	0.052
Intel Xeon E3-1220	5.349%	0.052
Intel Xeon E5-2683 v4	1.249%	0.030

$$\mathrm{M} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

$$\sigma_{est} = \sqrt{\frac{\sum (Y - Y')^2}{N}}$$

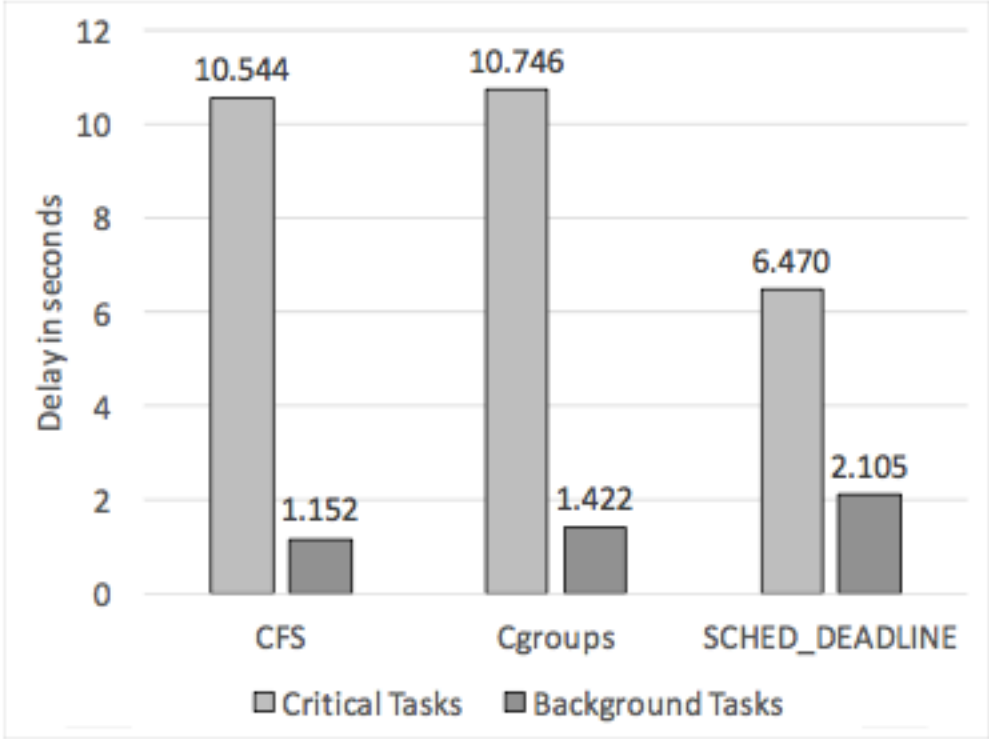


Fig. 9: Average delay in LCU workload tasks on the 32-core physical machine.