Software Resource Disaggregation for HPC with Serverless Computing

Marcin Copik, Alexandru Calotoiu (advisor), Torsten Hoefler (advisor)

SC22
Dallas, TX | hpc accelerates.

Tracking Wasted Money in HPC

FINAL REPORT
WORKLOAD ANALYSIS OF BLUE WATERS
(ACI 1650758)

Matthew D. Jones, Joseph P. White, Martins Innus, Robert L. DeLeon, Nikolay Simakov, Jeffrey T. Palmer, Steven M. Gallo, and Thomas R. Furlani (furlani@buffalo.edu), Center for Computational Research, University at Buffalo, SUNY

Michael Showerman, Robert Brunner, Andry Kot, Gregory Bauer, Brett Bode, Jeremy Enos, and William Kramer (wtkramer@illinois.edu), National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana Champaign

arXiv, 2017

Job Characteristics on Large-Scale Systems: Long-Term Analysis, Quantification, and Implications*

Tirthak Patel
Northeastern University

Zhengchun Liu, Raj Kettimuthu
Argonne National Laboratory

Paul Rich, William Allcock
Argonne National Laboratory

Devesh Tiwari
Northeastern University

SC, 2020

Comprehensive Workload Analysis and Modeling of a Petascale Supercomputer

Haihang You[1] and Hao Zhang[2]

[1] National Institute for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
[2] Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996, USA
{hyou,haozhang}@utk.edu

JSSPP, 2012

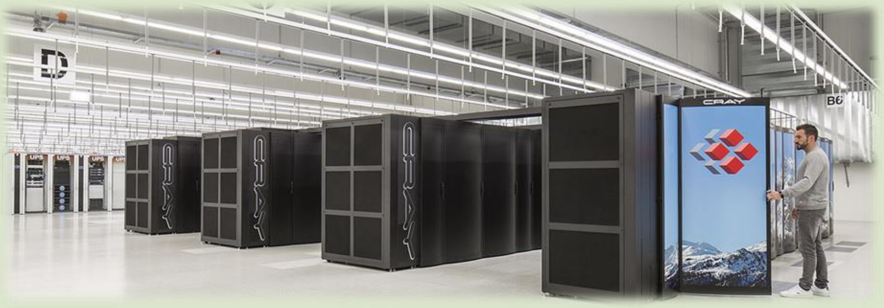
HPC System Utilization - CPU

Piz Daint, April 2022.

- XC50 nodes – CPU + GPU, 64 GB memory.
- XC40 nodes – CPU, 64/128 GB memory.

Query SLURM info every two minutes.

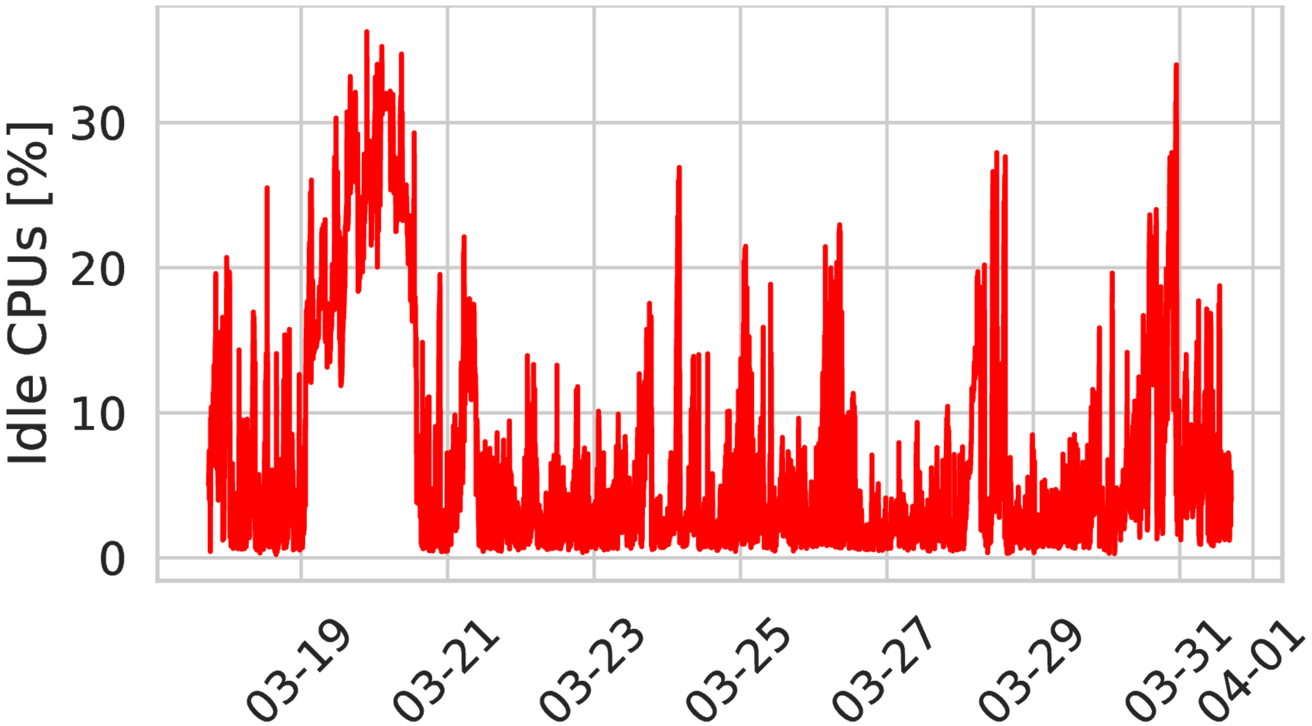
HPC System Utilization - CPU



Piz Daint, April 2022.

- XC50 nodes – CPU + GPU, 64 GB memory.
- XC40 nodes – CPU, 64/128 GB memory.

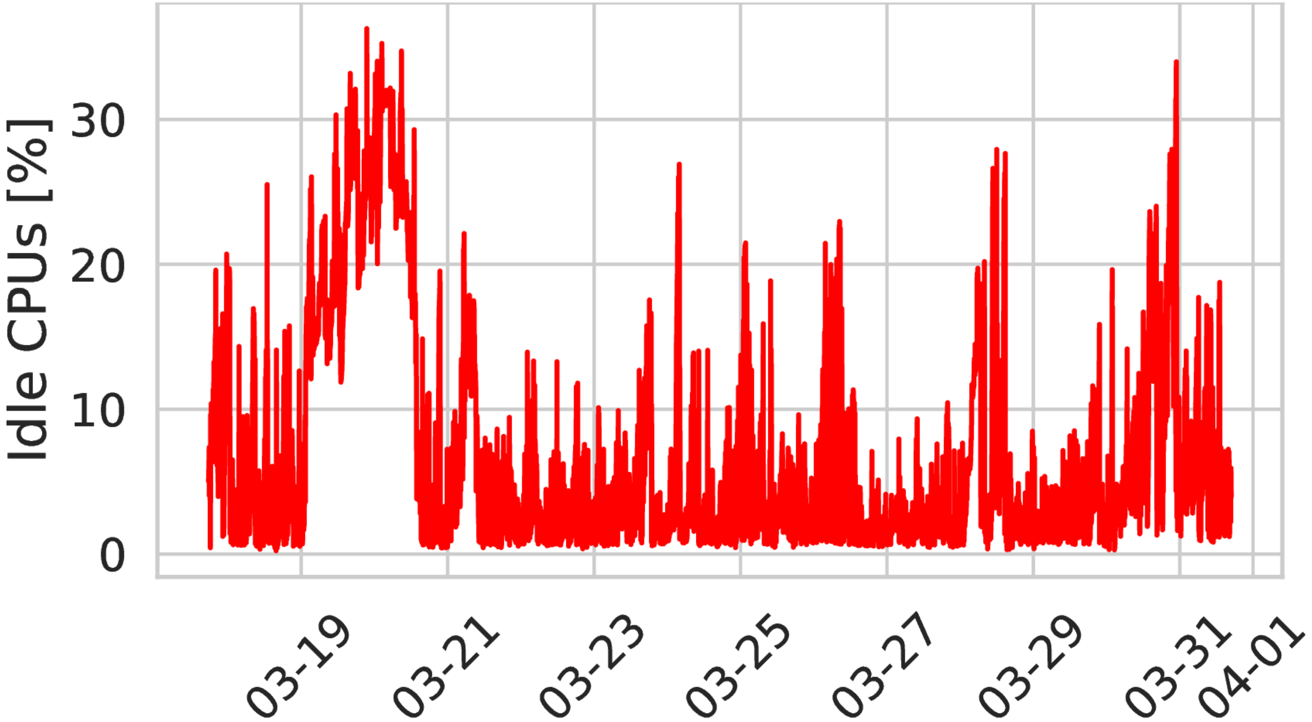
Query SLURM info every two minutes.



HPC System Utilization - CPU

Piz Daint, April 2022.
- XC50 nodes – CPU + GPU, 64 GB memory.
- XC40 nodes – CPU, 64/128 GB memory.

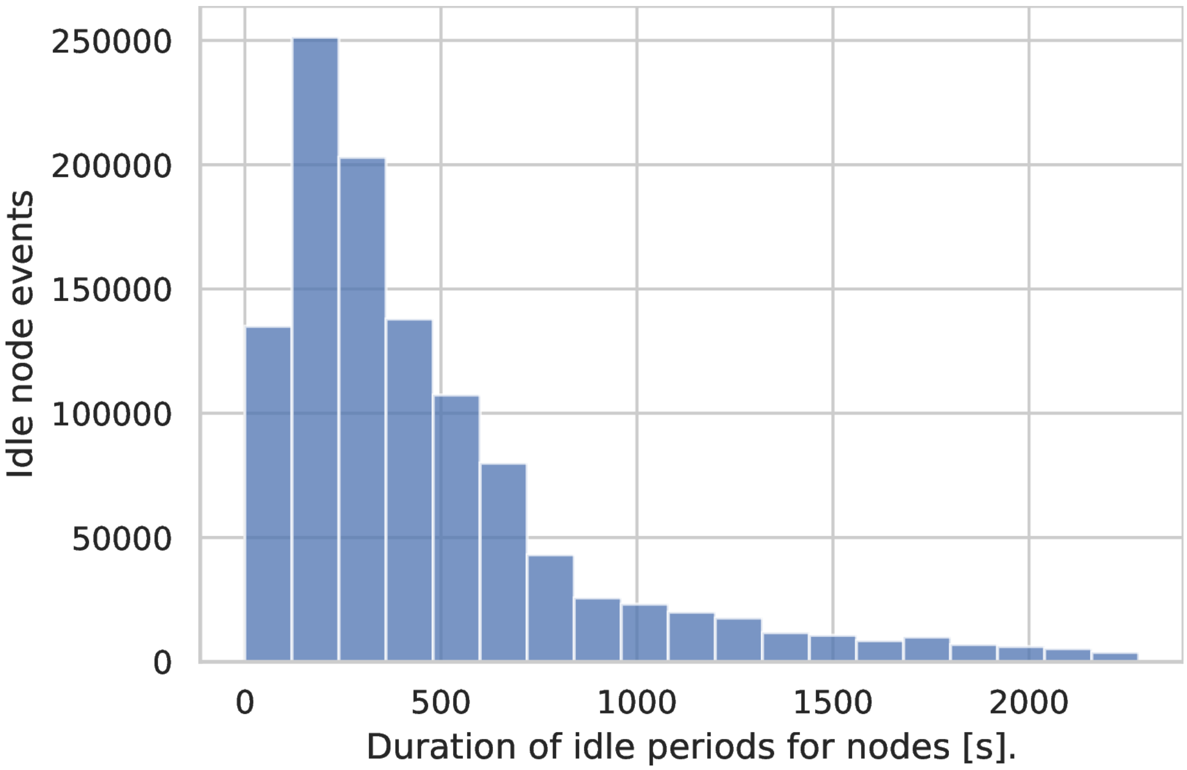
Query SLURM info every two minutes.



How long do nodes stay idle?

HPC System Utilization - CPU

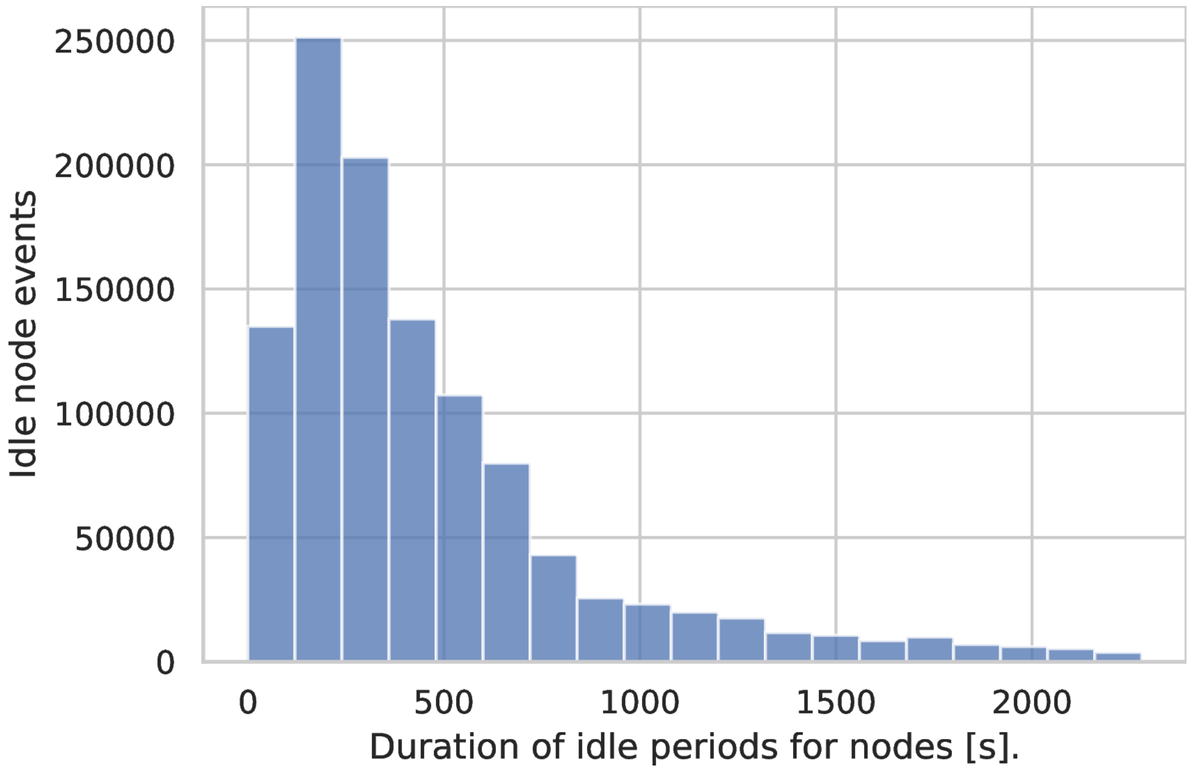
Minimum



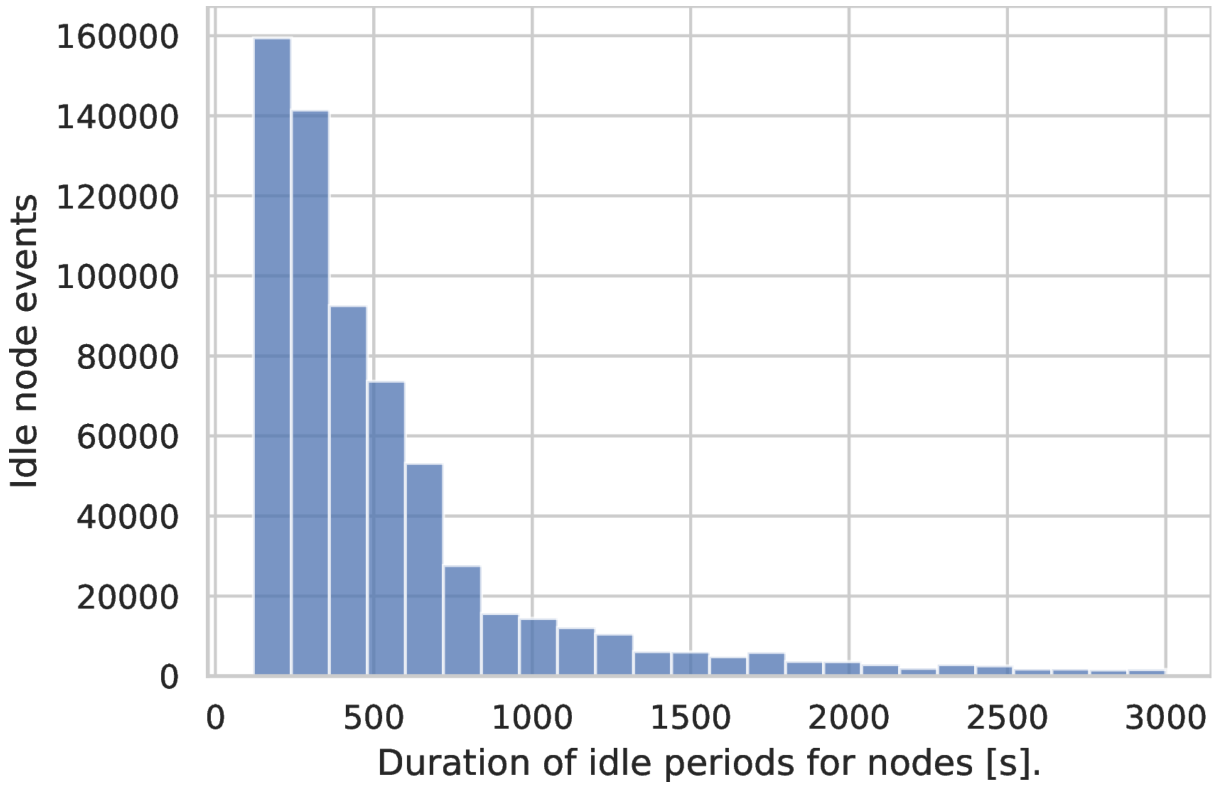
Maximum

HPC System Utilization - CPU

Minimum

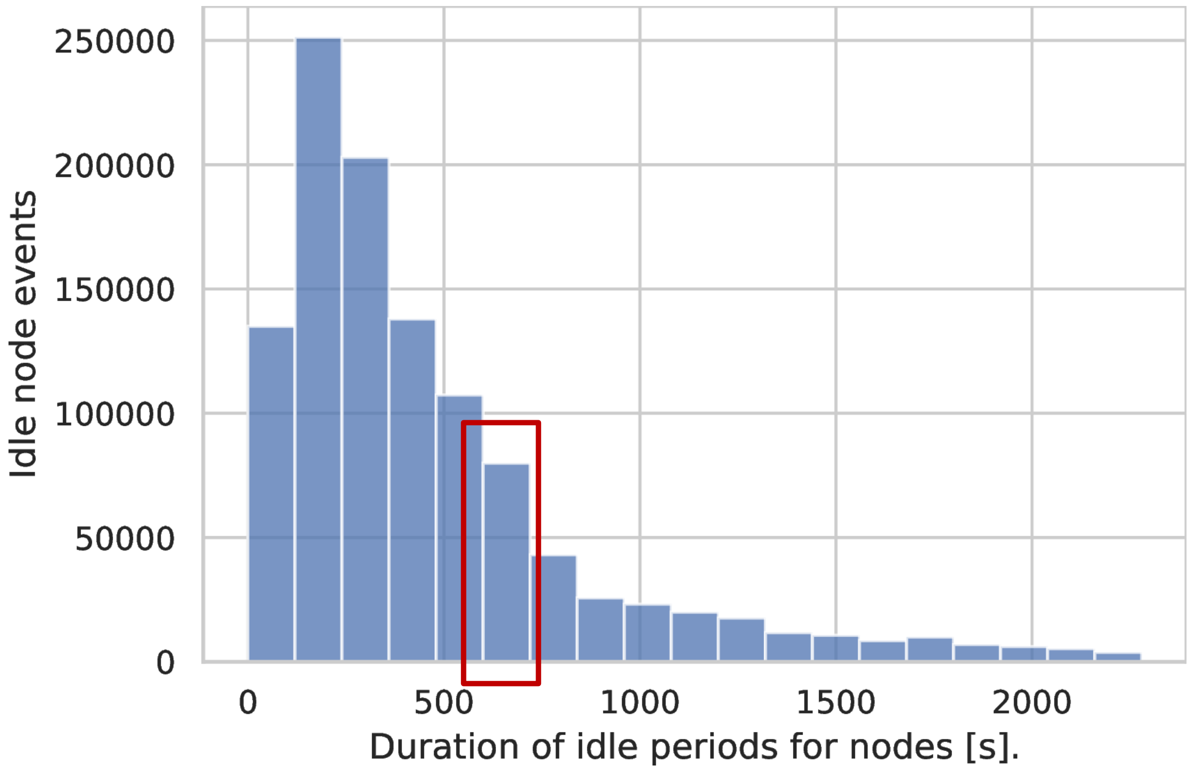


Maximum

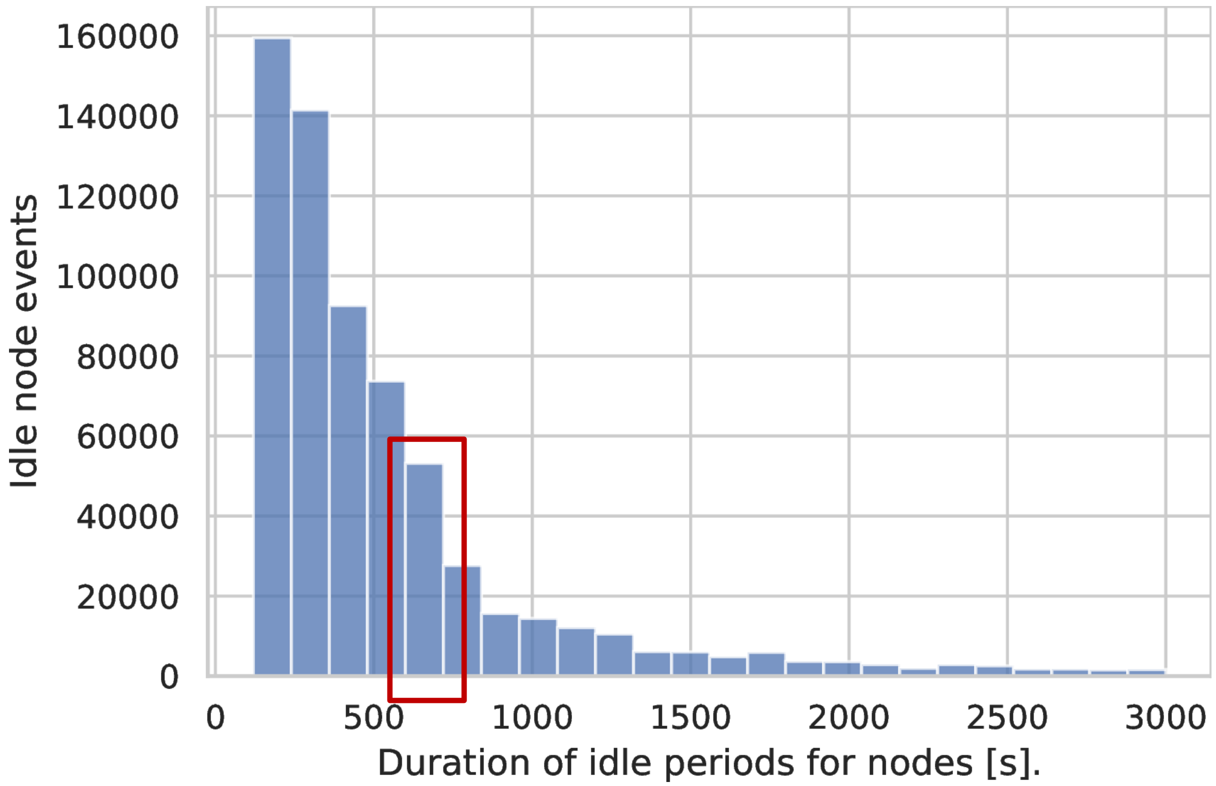


HPC System Utilization - CPU

Minimum

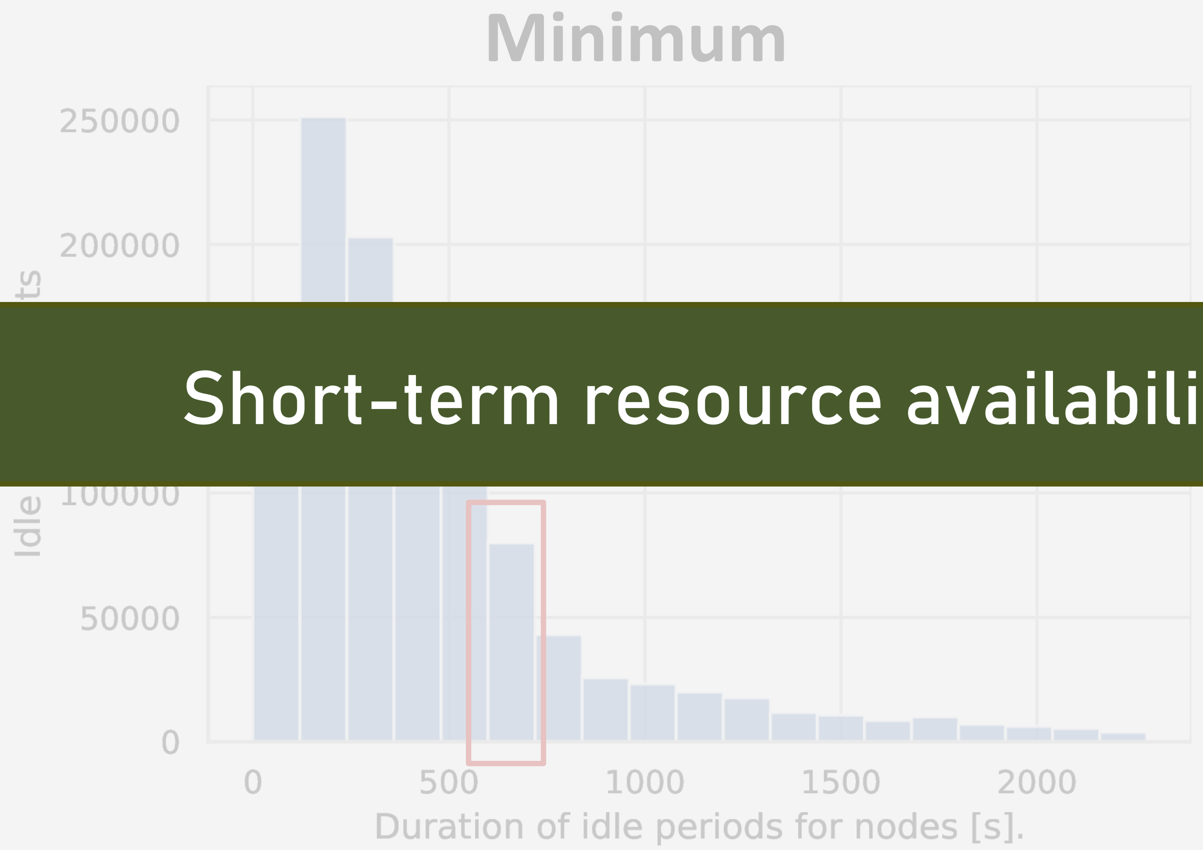


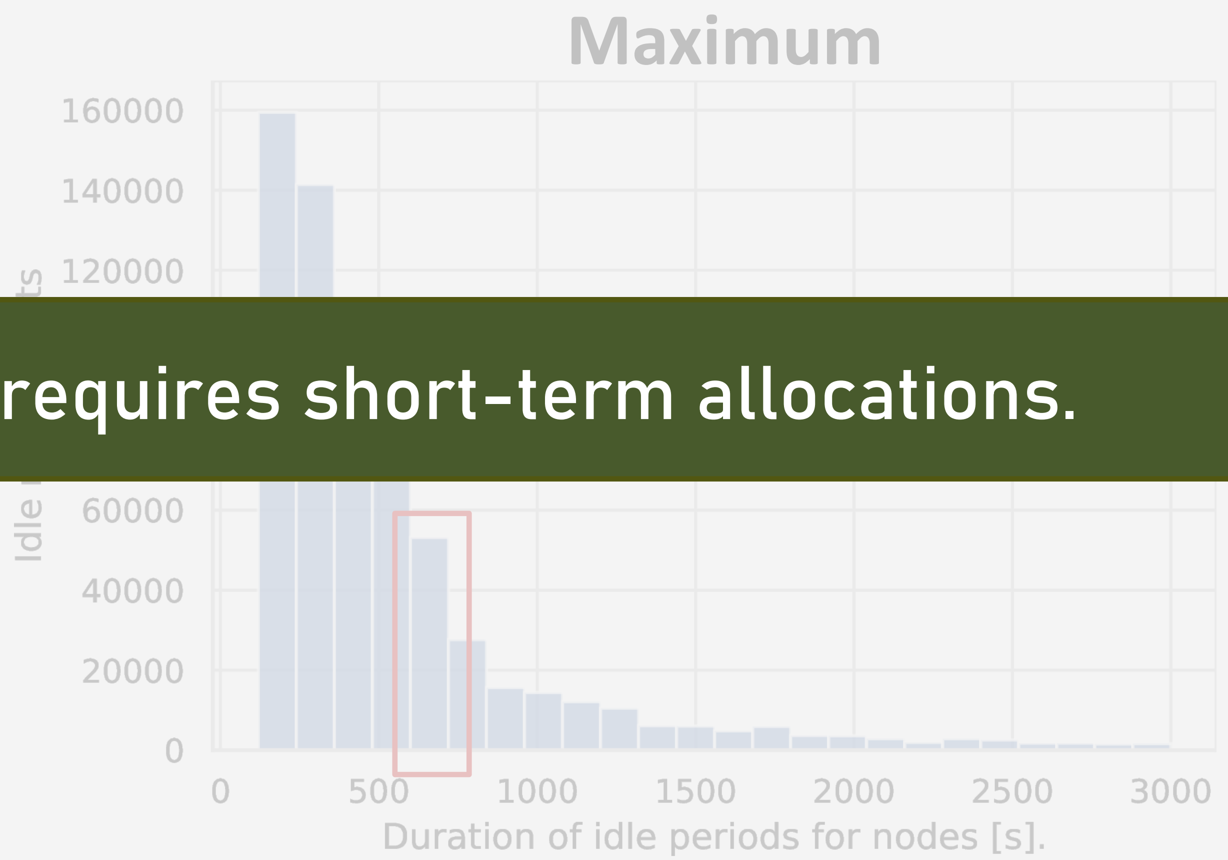
Maximum



80% and 70% of idle node events last less than 10 minutes.

HPC System Utilization - CPU

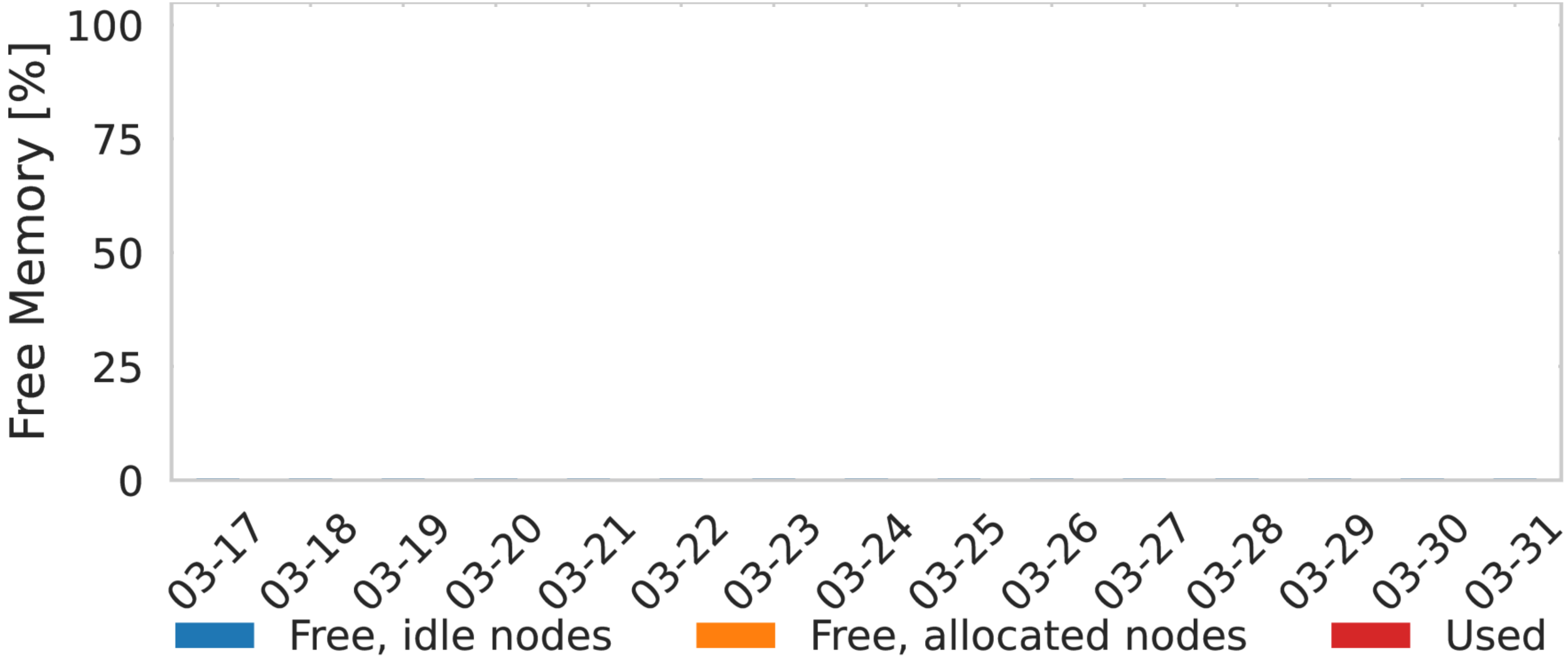




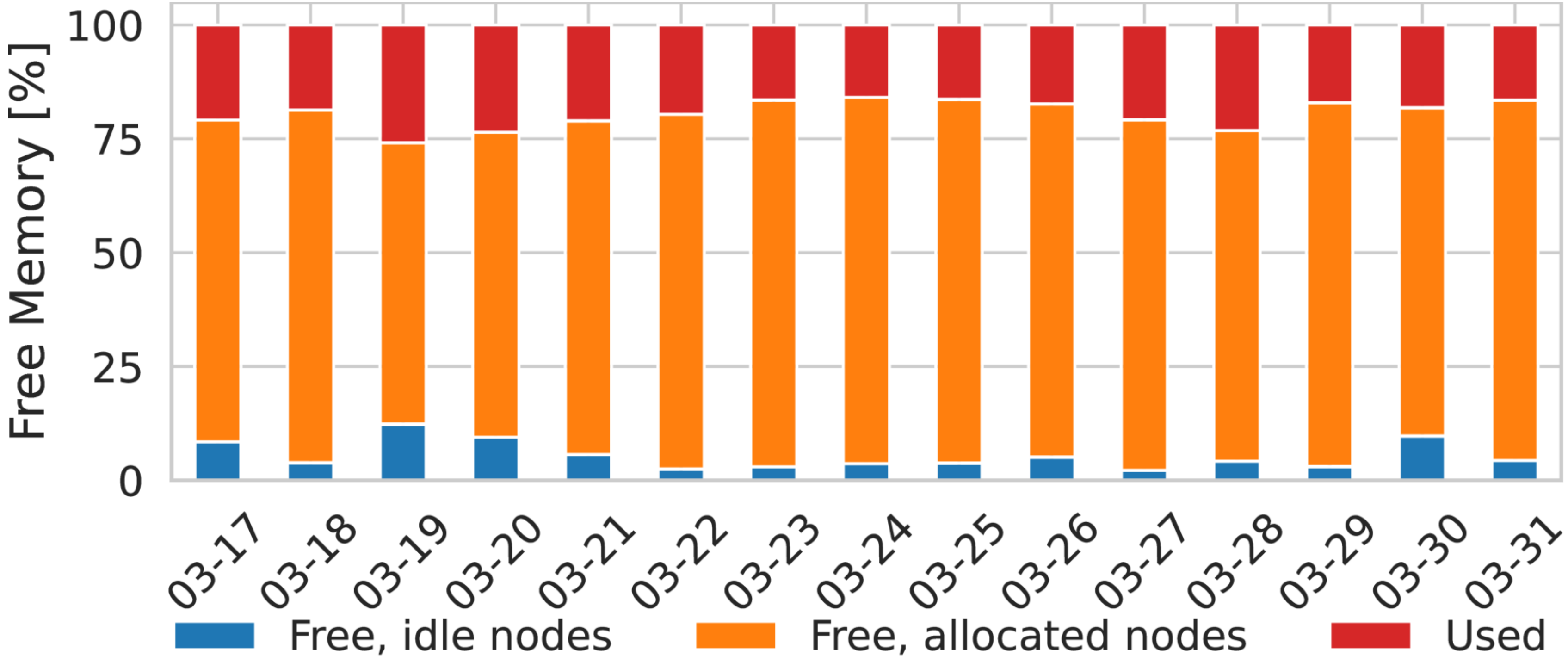
Short-term resource availability requires short-term allocations.

80% and 70% of idle node events last less than 10 minutes.

HPC System Utilization - Memory



HPC System Utilization - Memory



HPC System Utilization - Memory

A Case For Intra-rack Resource Disaggregation in HPC

GEORGE MICHELOGIANNAKIS, Lawrence Berkeley National Laboratory, USA
BENJAMIN KLENK, NVIDIA, USA
BRANDON COOK, Lawrence Berkeley National Laboratory, USA
MIN YEE TEH and MADELEINE GLICK, Columbia University, USA
LARRY DENNISON, NVIDIA, USA
KEREN BERGMAN, Columbia University, USA
JOHN SHALF, Lawrence Berkeley National Laboratory, USA

TACO, 2022

Quantifying Memory Underutilization in HPC Systems and Using it to Improve Performance via Architecture Support

Gagandeep Panwar*, Virginia Tech, Blacksburg, USA, gpanwar@vt.edu
Da Zhang*, Virginia Tech, Blacksburg, USA, daz3@vt.edu
Yihan Pang*, Virginia Tech, Blacksburg, USA, pyihan1@vt.edu
Mai Dahshan, Virginia Tech, Blacksburg, USA, mdahshan@vt.edu
Nathan DeBardeleben, Los Alamos National Laboratory, Los Alamos, USA, ndebard@lanl.gov
Binoy Ravindran, Virginia Tech, Blacksburg, USA, binoy@vt.edu
Xun Jian, Virginia Tech, Blacksburg, USA, xunj@vt.edu

MICRO, 2019

FINAL REPORT
WORKLOAD ANALYSIS OF BLUE WATERS
(ACI 1650758)

Matthew D. Jones, Joseph P. White, Martins Innus, Robert L. DeLeon, Nikolay Simakov, Jeffrey T. Palmer, Steven M. Gallo, and Thomas R. Furlani (furlani@buffalo.edu), Center for Computational Research, University at Buffalo, SUNY

Michael Showerman, Robert Brunner, Andry Kot, Gregory Bauer, Brett Bode, Jeremy Enos, and William Kramer (wtkramer@illinois.edu), National Center for Supercomputing Applications (NCSA), University of Illinois at Urbana Champaign

arXiv, 2017

A Holistic View of Memory Utilization on HPC Systems: Current and Future Trends

Ivy B. Peng*, peng8@llnl.gov, Lawrence Livermore National Laboratory, USA
Ian Karlin, karlin1@llnl.gov, Lawrence Livermore National Laboratory, USA
Maya B. Gokhale, gokhale2@llnl.gov, Lawrence Livermore National Laboratory, USA
Kathleen Shoga, Shoga1@llnl.gov, Lawrence Livermore National Laboratory, USA
Matthew Legendre, legendre1@llnl.gov, Lawrence Livermore National Laboratory, USA
Todd Gamblin, gamblin2@llnl.gov, Lawrence Livermore National Laboratory, USA

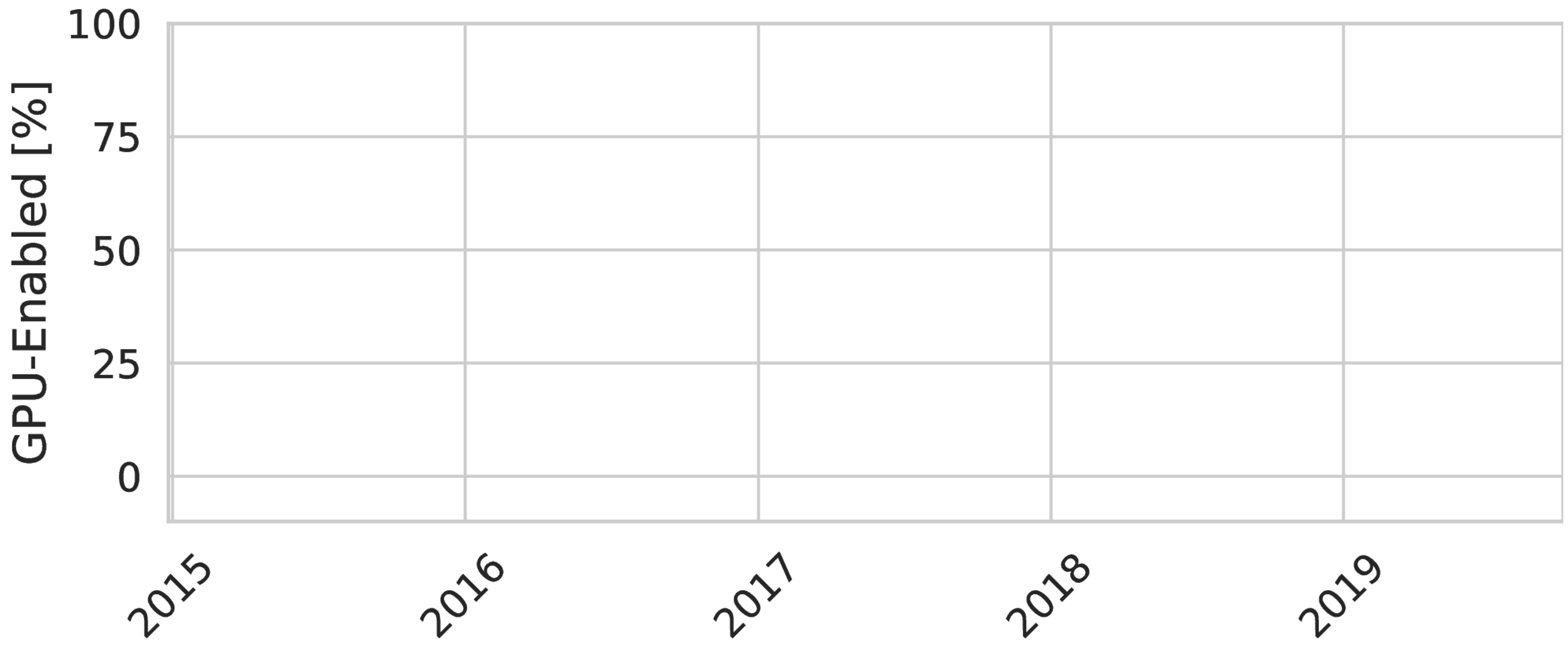
MEMSYS, 2021

HPC System Utilization - GPU

Learning from Five-year Resource-Utilization Data of Titan System

Feiyi Wang*, Sarp Oral†, Satyabrata Sen ‡ and Neena Imam§
Oak Ridge National Laboratory

CLUSTER, 2019



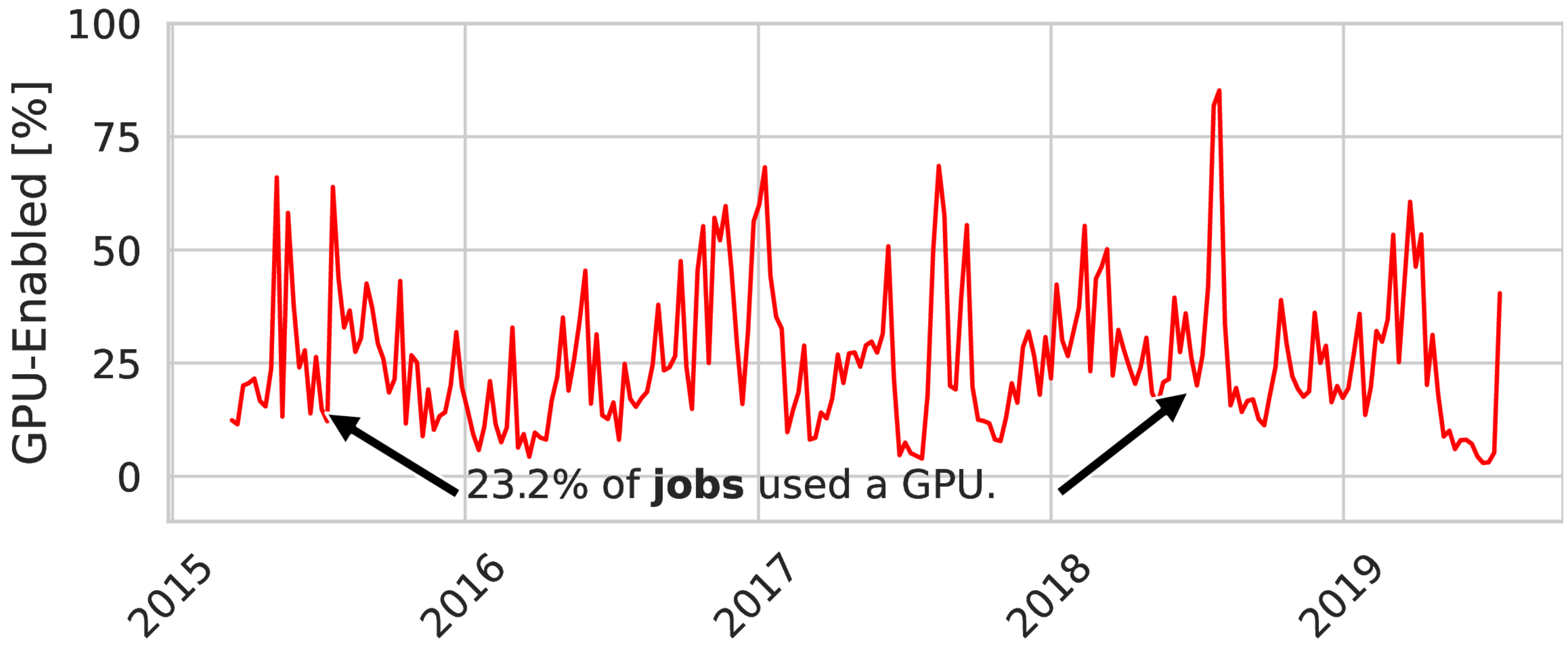
HPC System Utilization - GPU

Learning from Five-year Resource-Utilization Data of Titan System

Feiyi Wang*, Sarp Oral†, Satyabrata Sen ‡ and Neena Imam§

Oak Ridge National Laboratory

CLUSTER, 2019

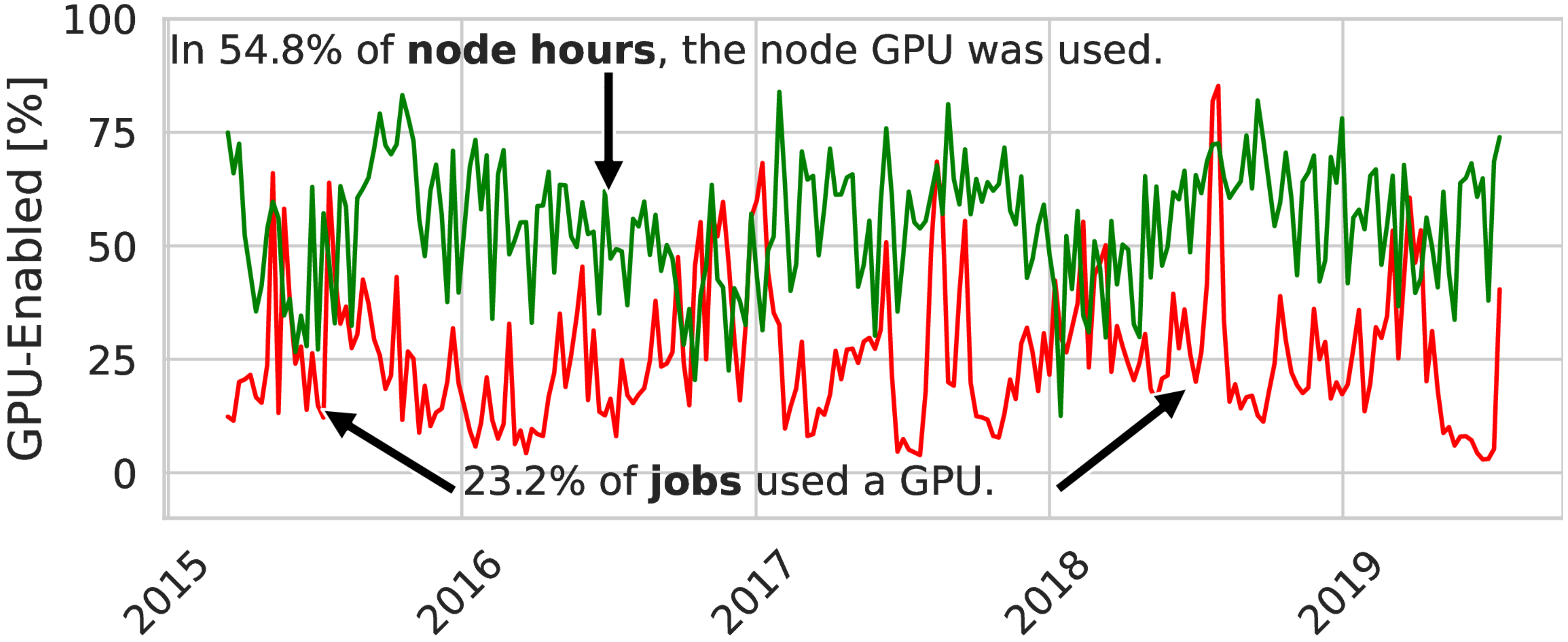


HPC System Utilization - GPU

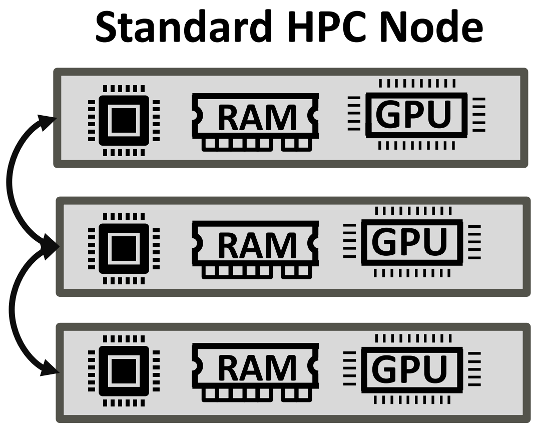
Learning from Five-year Resource-Utilization Data of Titan System

Feiyi Wang*, Sarp Oral†, Satyabrata Sen ‡ and Neena Imam§
Oak Ridge National Laboratory

CLUSTER, 2019



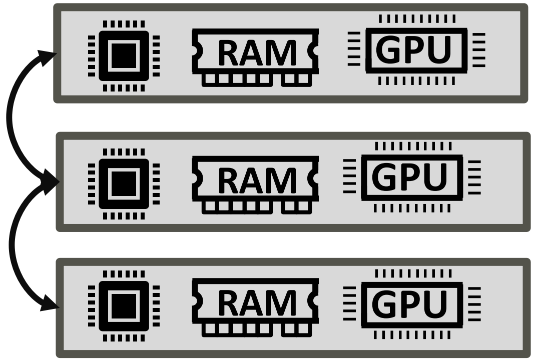
Software Solution



- ✔ High performance
- ✘ Inflexible architecture

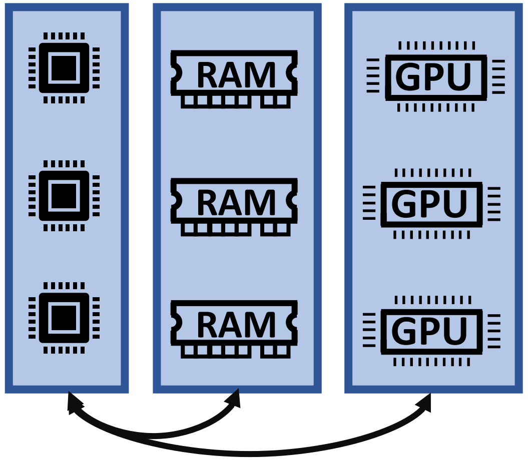
Software Solution

Standard HPC Node



- High performance
- Inflexible architecture

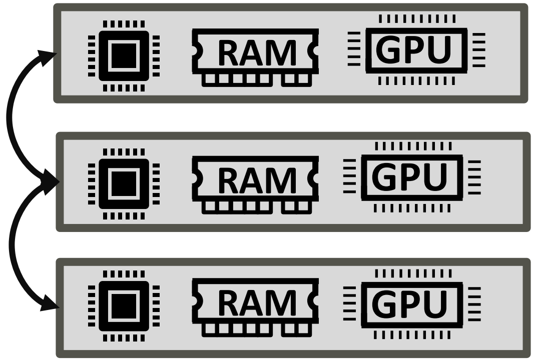
Hardware Disaggregation



- High efficiency
- Cost, performance penalty

Software Solution

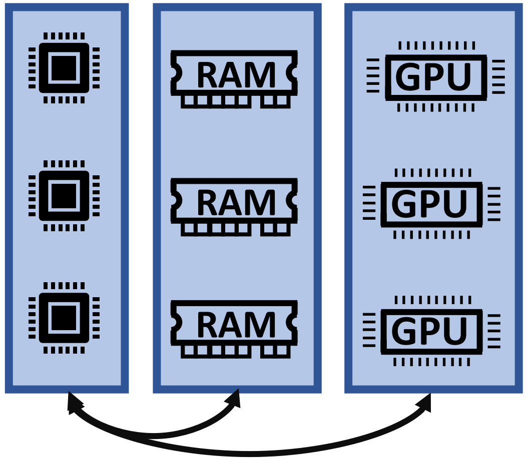
Standard HPC Node



High performance

Inflexible architecture

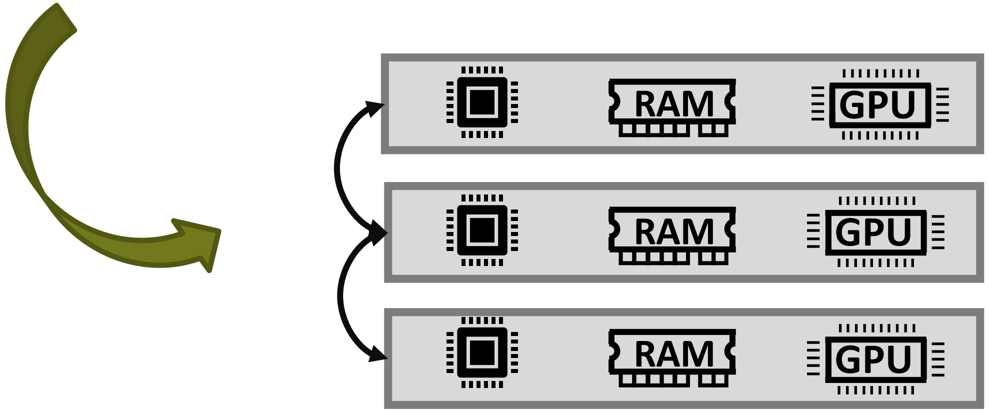
Hardware Disaggregation



High efficiency

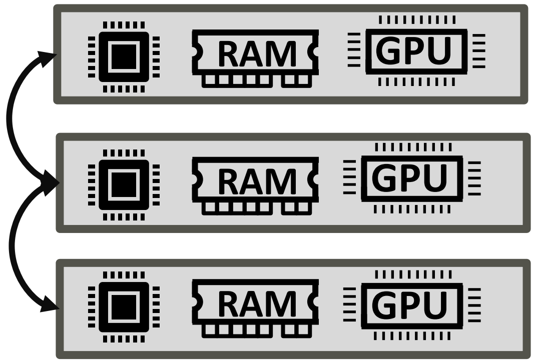
Cost, performance penalty

Existing Coupled Hardware Systems



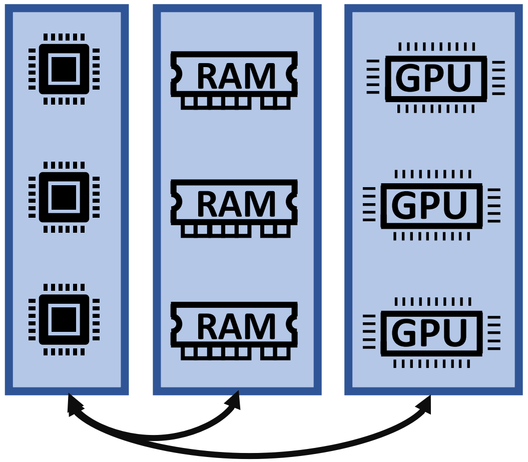
Software Solution

Standard HPC Node



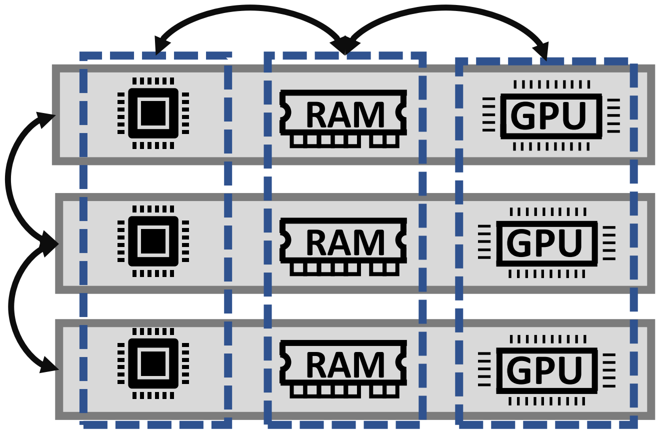
- High performance
- Inflexible architecture

Hardware Disaggregation



- High efficiency
- Cost, performance penalty

Existing Coupled Hardware Systems



Software Abstraction for Disaggregation

We propose a software disaggregation approach to share node resources

We propose a software disaggregation approach to share node resources between
coarse-grained, long-running, and static batch jobs

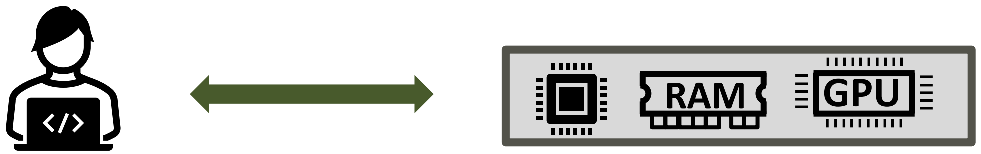
We propose a software disaggregation approach to share node resources between
coarse-grained, long-running, and static batch jobs
and
fine-grained, short-term, and dynamically allocated serverless functions.

Serverless as an Answer

Serverless as an Answer

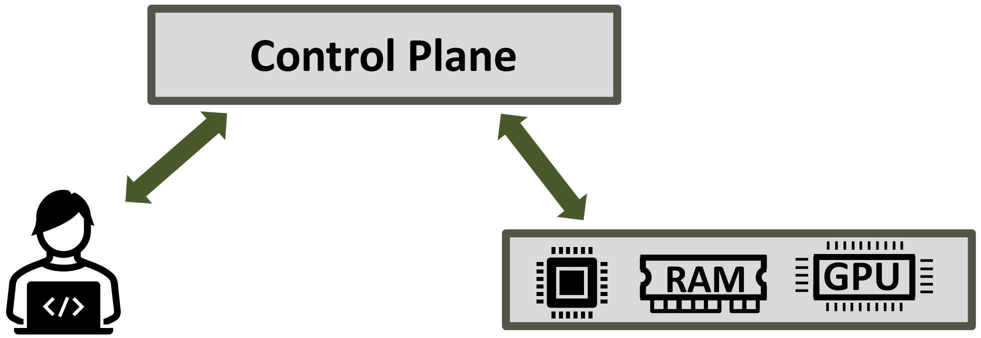
Serverless as an Answer

Hardware Abstraction



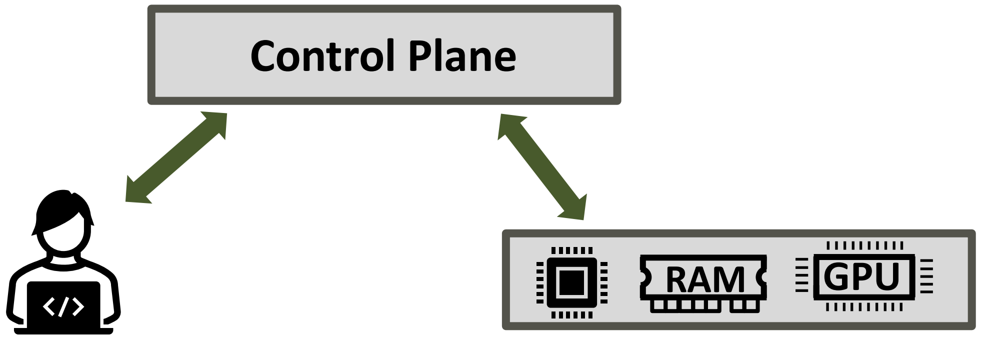
Serverless as an Answer

Hardware Abstraction

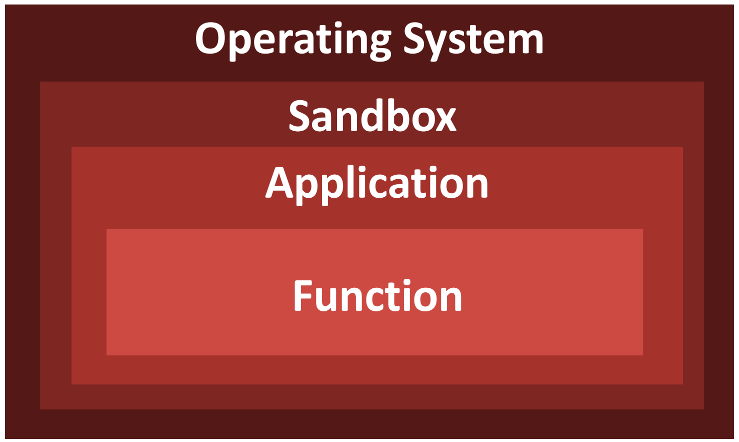


Serverless as an Answer

Hardware Abstraction

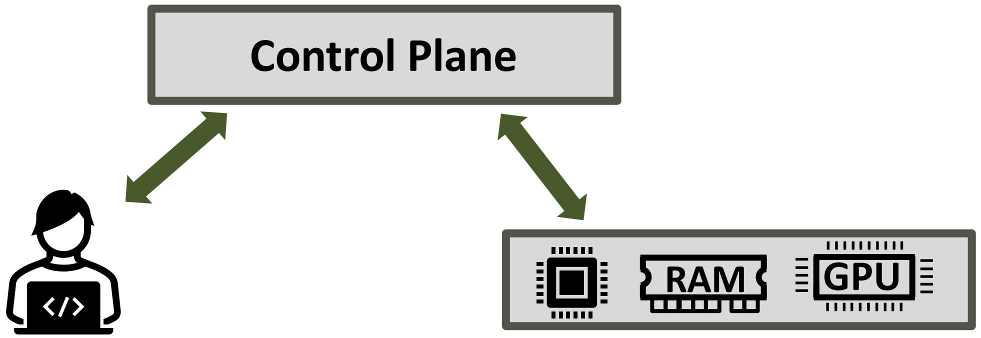


Software Abstraction

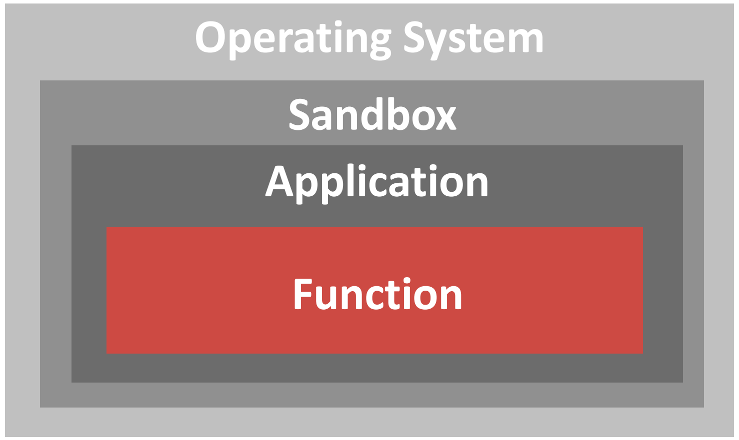


Serverless as an Answer

Hardware Abstraction

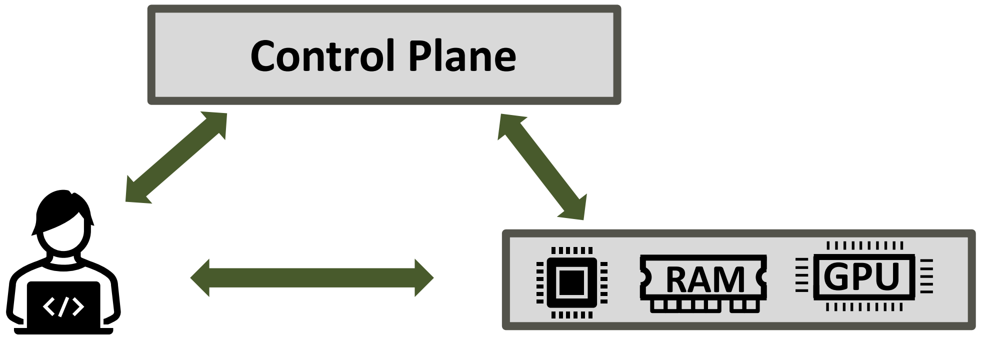


Software Abstraction



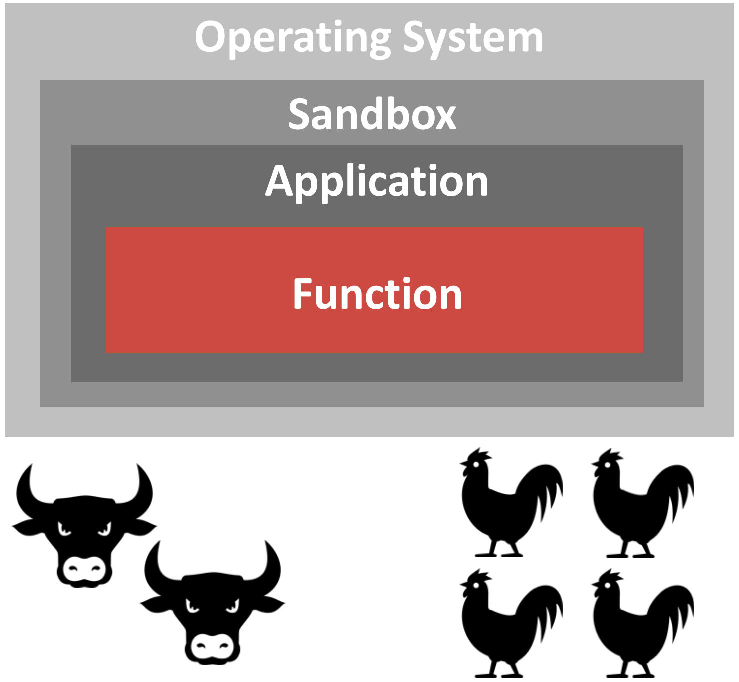
Serverless as an Answer

Hardware Abstraction



Pay-as-you-go billing

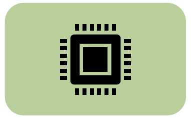
Software Abstraction

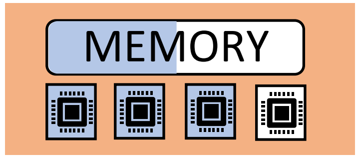


Granular computing

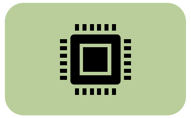
Serverless Disaggregation

Batch jobs

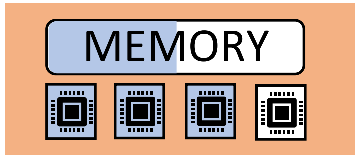




Serverless Disaggregation



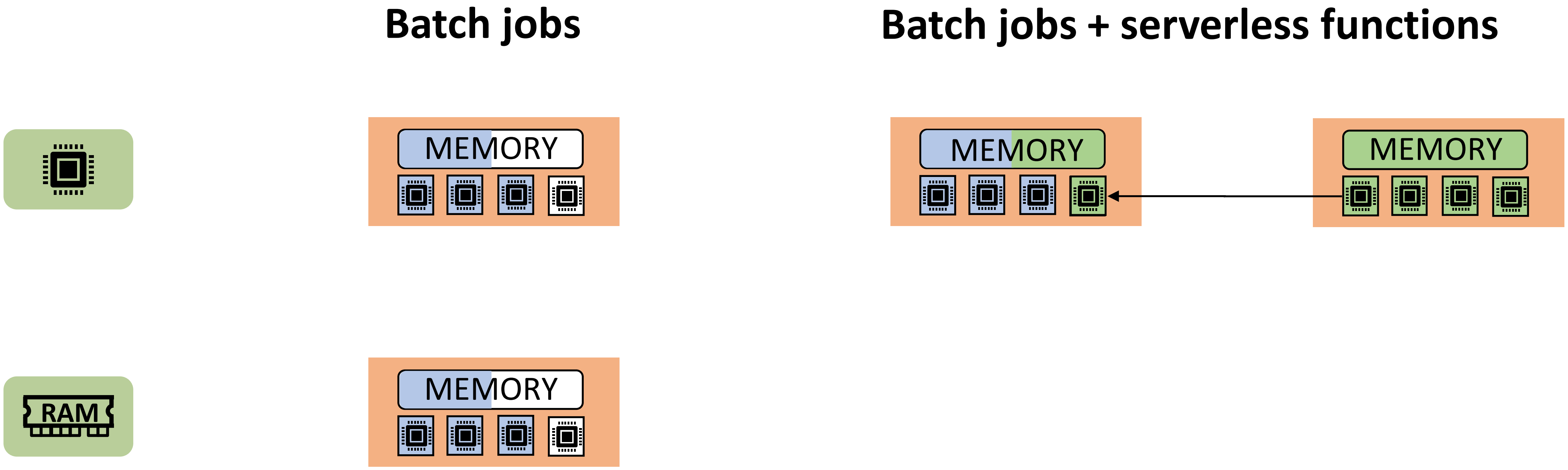
Batch jobs



Batch jobs + serverless functions

Serverless Disaggregation

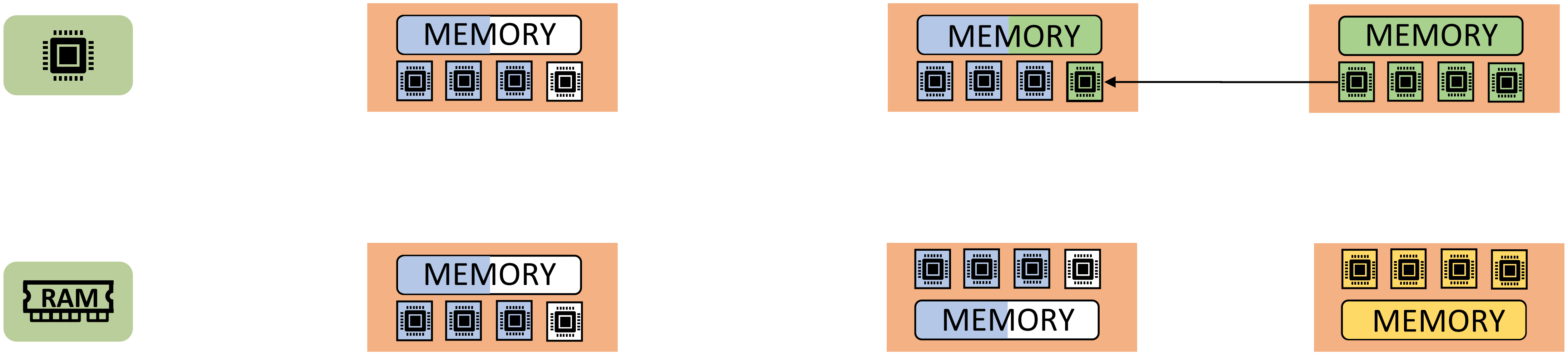
Serverless Disaggregation



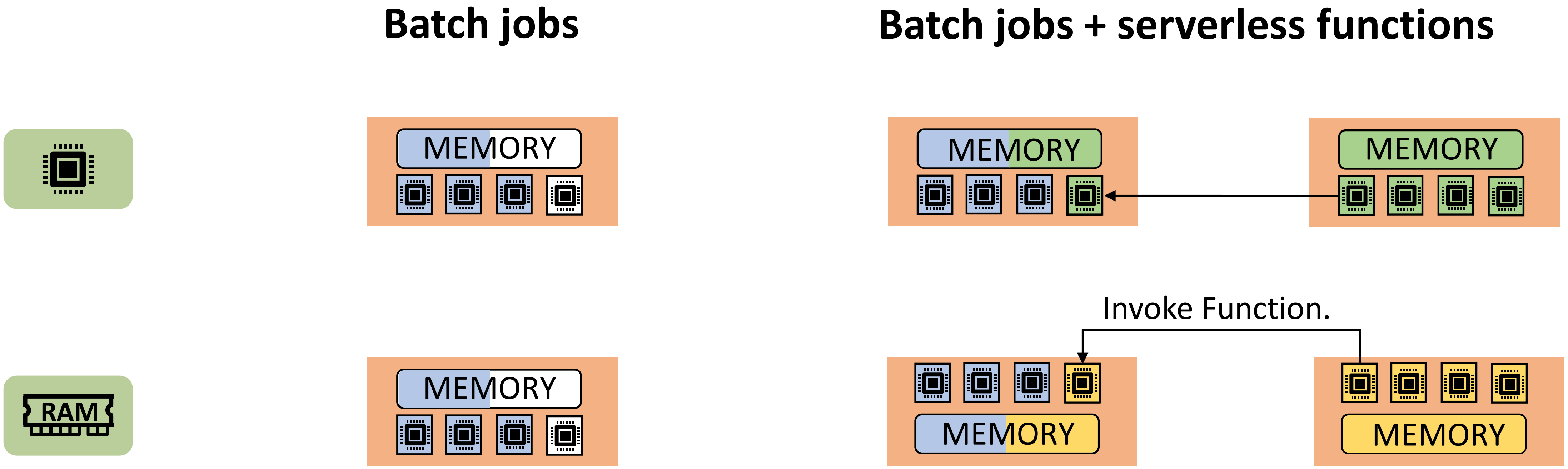
Serverless Disaggregation

Batch jobs

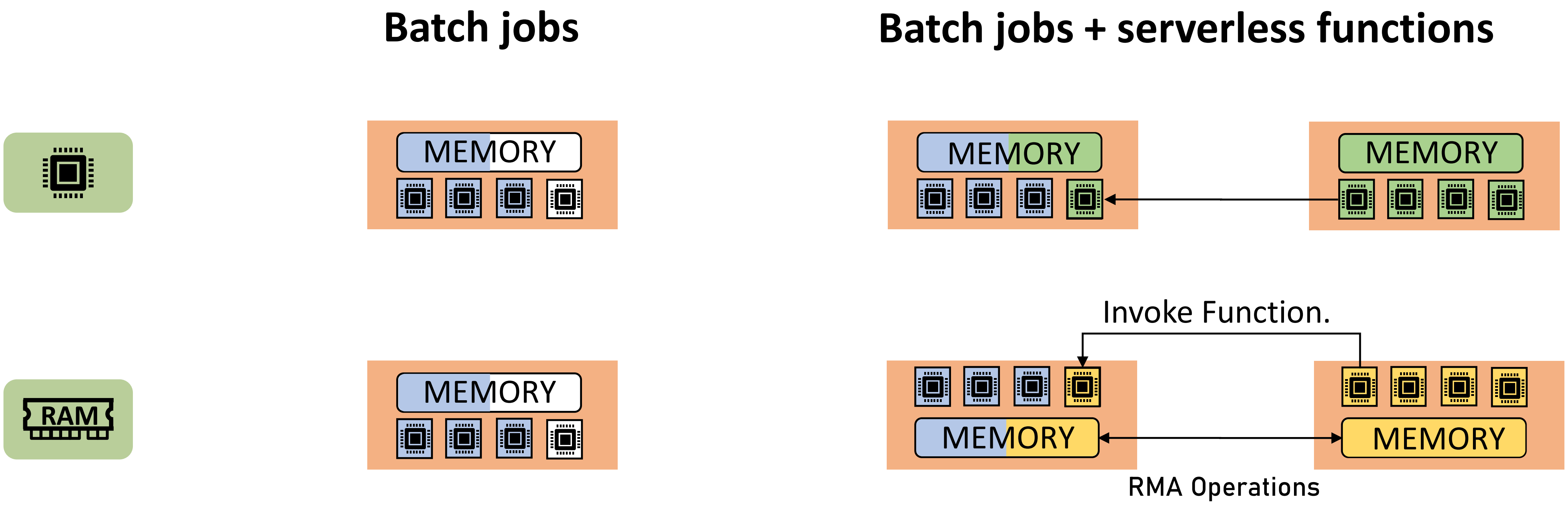
Batch jobs + serverless functions



Serverless Disaggregation

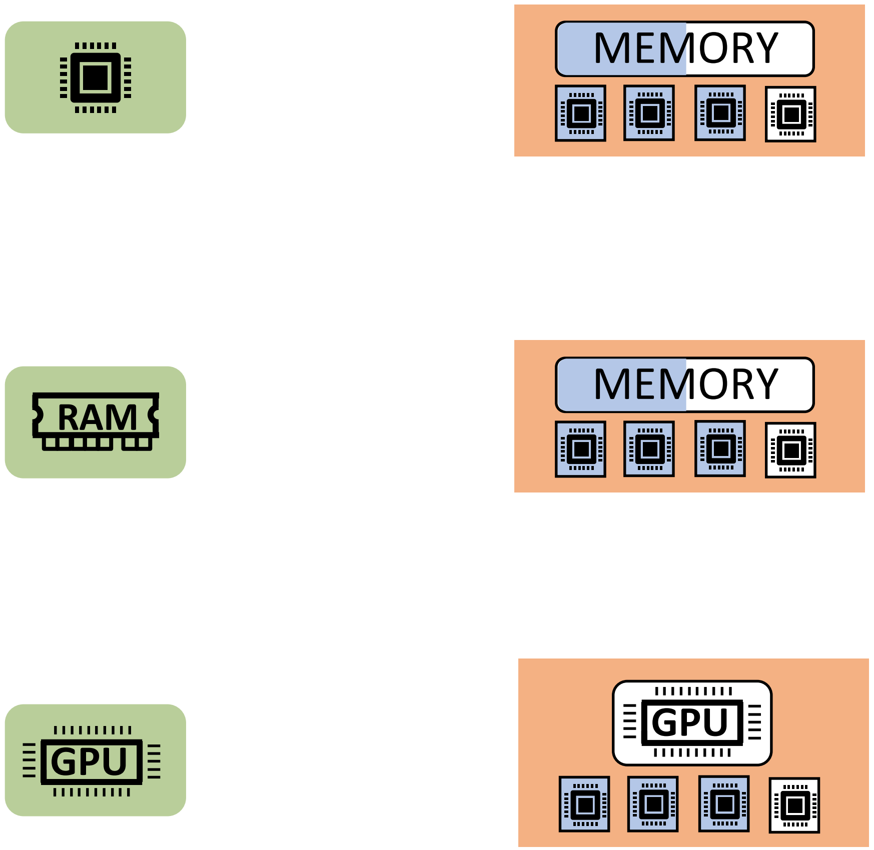


Serverless Disaggregation

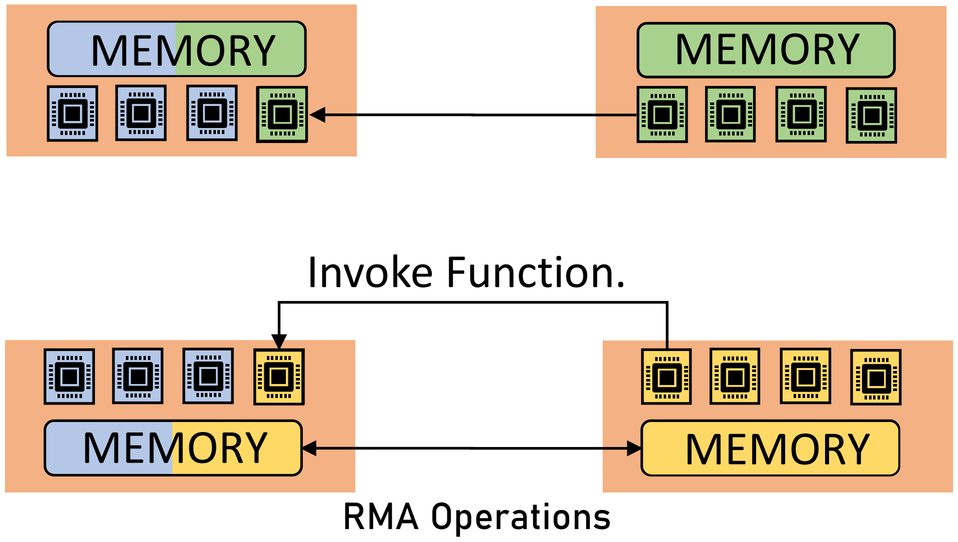


Serverless Disaggregation

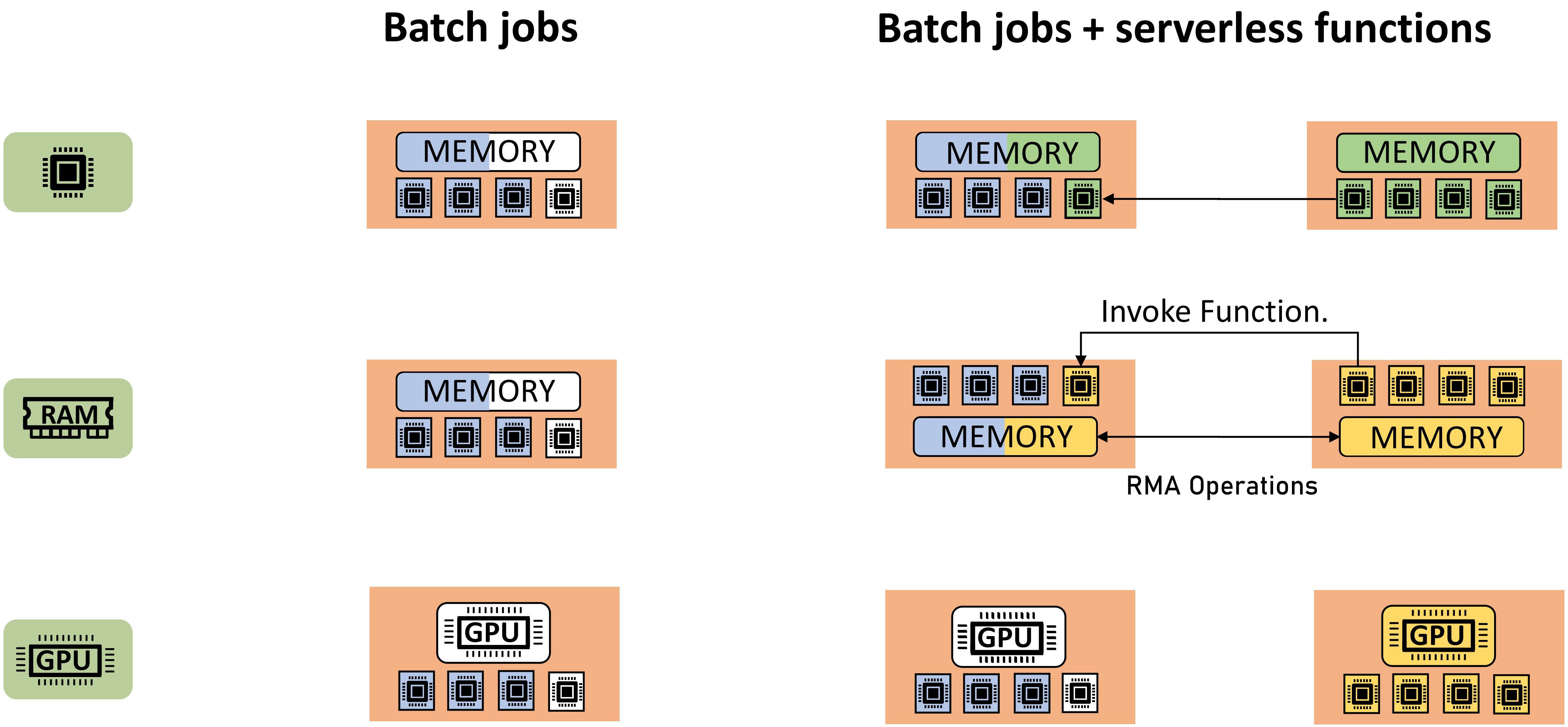
Batch jobs



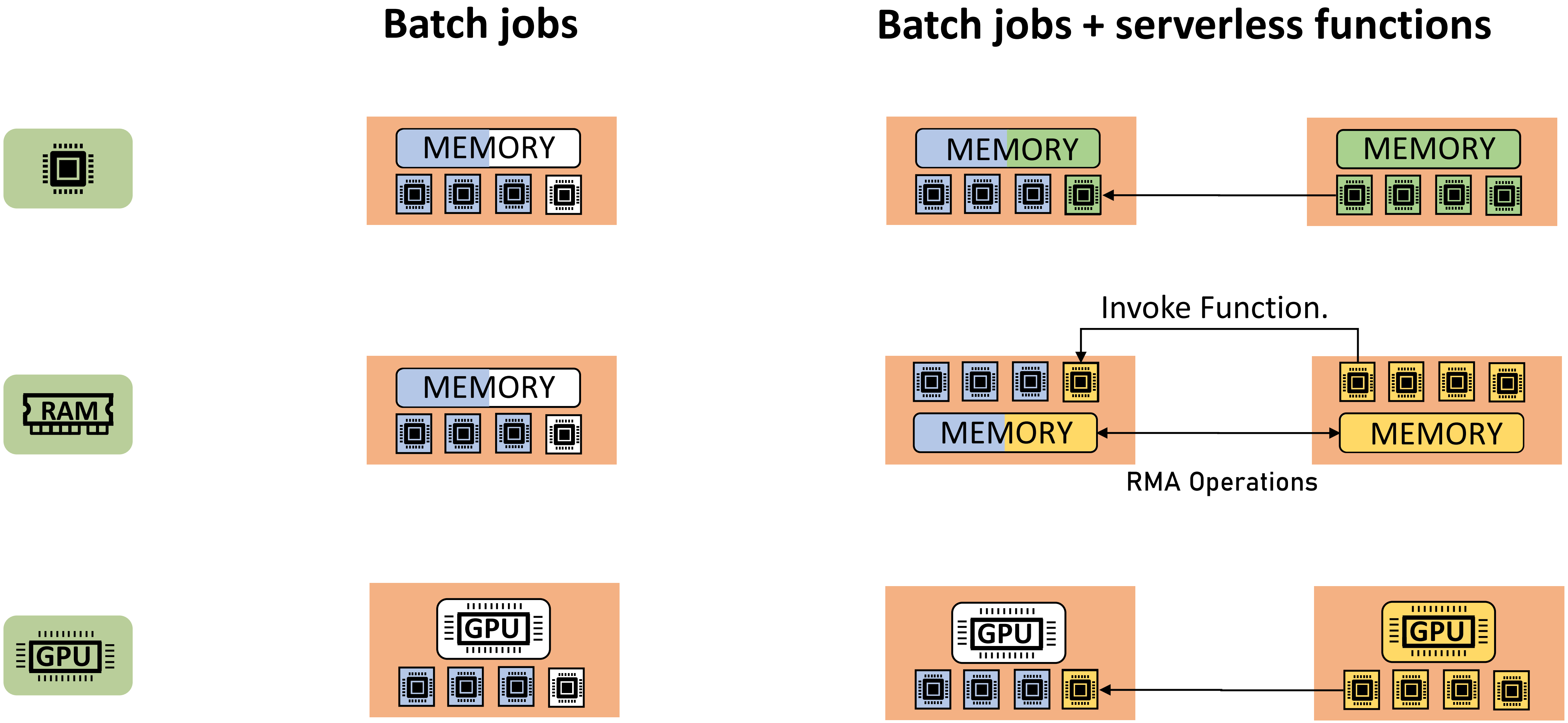
Batch jobs + serverless functions



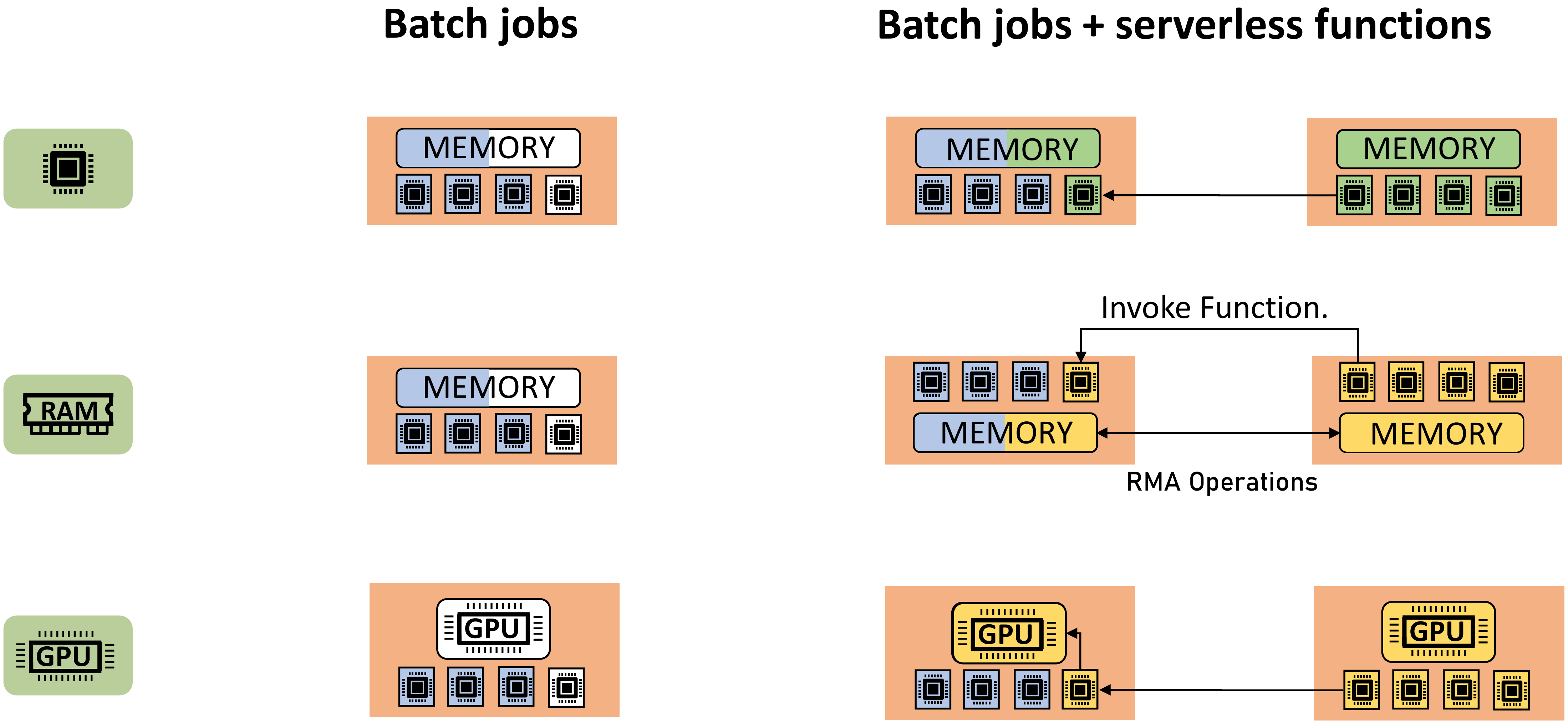
Serverless Disaggregation



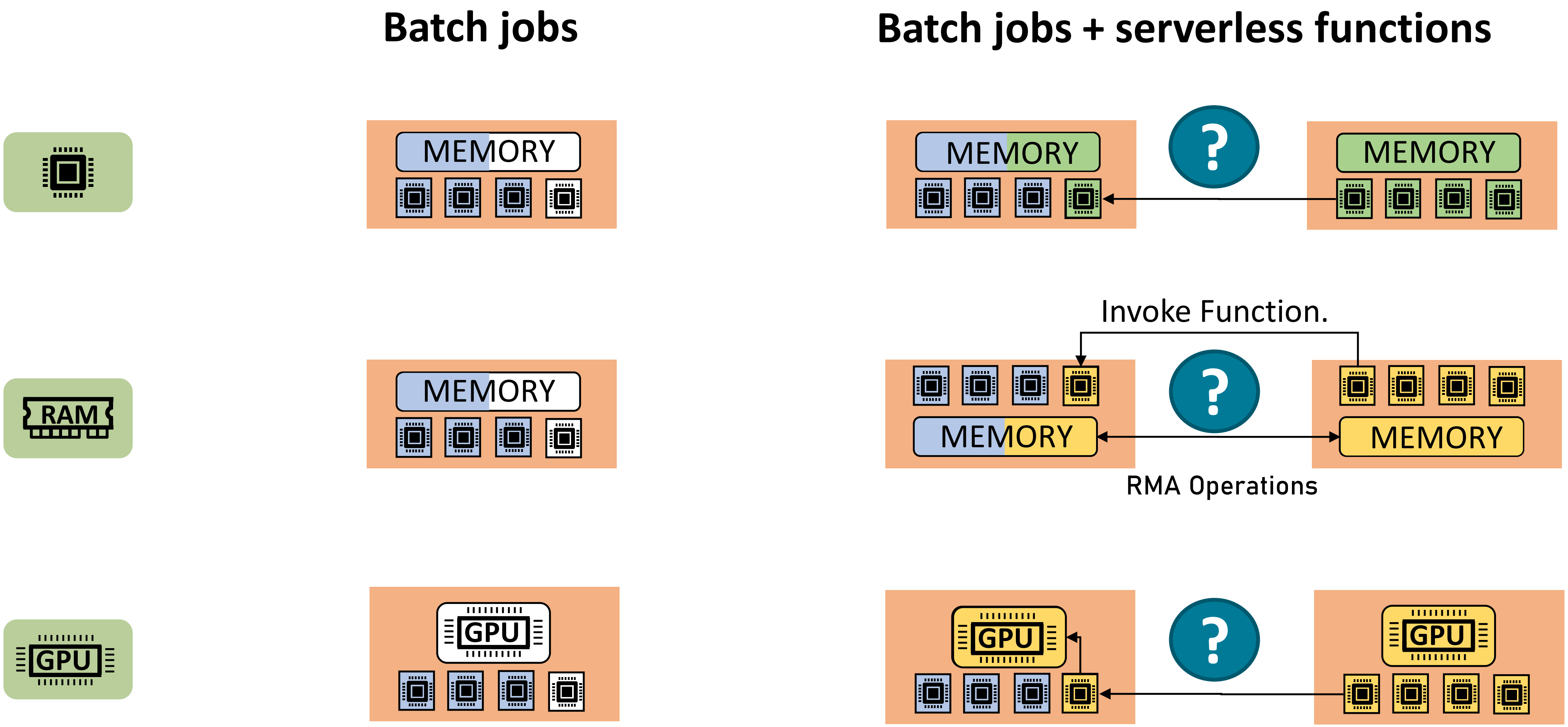
Serverless Disaggregation



Serverless Disaggregation



Serverless Disaggregation



Evaluation

XC50 nodes - 12 CPU cores, GPU, 64 GB memory.

XC40 nodes - 36 CPU cores, 64/128 GB memory.

Cray Aries interconnect.

36 CPU cores, 377 GB memory.
Ethernet with RoCEv2 support.

#1 CPU Sharing

Mean utilisation

collocated benchmark type	Disaggregation	Ideal Non-sharing	Realistic
BT, A	0.937	0.893	0.693
BT, W	0.903	0.89	0.64
CG, B	0.992	0.901	0.65
EP, B	0.915	0.891	0.661
LU, A	0.941	0.893	0.674
MG, A	0.903	0.89	0.625
MG, W	0.903	0.89	0.638

LULESH

64 ranks, 2 nodes

32 out of 36 cores allocated.

NAS

1 – 4 ranks

Distributed across nodes.

#1 CPU Sharing

collocated benchmark type	Mean utilisation: Disaggregation	Mean utilisation: Ideal Non-sharing	Mean utilisation: Realistic	Total time: Disaggregation	Total time: Realistic	Core hours: Disaggregation	Core hours: Ideal Non-sharing	Core hours: Realistic
BT, A	0.937	0.893	0.693	0.877	1.0	0.968	1.0	1.29
BT, W	0.903	0.89	0.64	0.981	1.0	0.994	1.0	1.39
CG, B	0.992	0.901	0.65	0.94	1.0	0.908	1.0	1.39
EP, B	0.915	0.891	0.661	0.901	1.0	0.98	1.0	1.35
LU, A	0.941	0.893	0.674	0.929	1.0	0.964	1.0	1.33
MG, A	0.903	0.89	0.625	1.01	1.0	1.01	1.0	1.42
MG, W	0.903	0.89	0.638	1.01	1.0	1.0	1.0	1.39

LULESH

64 ranks, 2 nodes

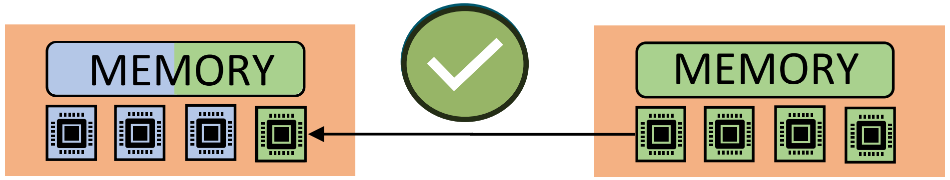
32 out of 36 cores allocated.

NAS

1 – 4 ranks

Distributed across nodes.

#1 CPU Sharing



collocated benchmark type	Mean utilisation			Total time		Core hours		
	Disaggregation	Ideal Non-sharing	Realistic	Disaggregation	Realistic	Disaggregation	Ideal Non-sharing	Realistic
BT, A	0.937	0.893	0.693	0.877	1.0	0.968	1.0	1.29
BT, W	0.903	0.89	0.64	0.981	1.0	0.994	1.0	1.39
CG, B	0.992	0.901	0.65	0.94	1.0	0.908	1.0	1.39
EP, B	0.915	0.891	0.661	0.901	1.0	0.98	1.0	1.35
LU, A	0.941	0.893	0.674	0.929	1.0	0.964	1.0	1.33
MG, A	0.903	0.89	0.625	1.01	1.0	1.01	1.0	1.42
MG, W	0.903	0.89	0.638	1.01	1.0	1.0	1.0	1.39

LULESH

64 ranks, 2 nodes

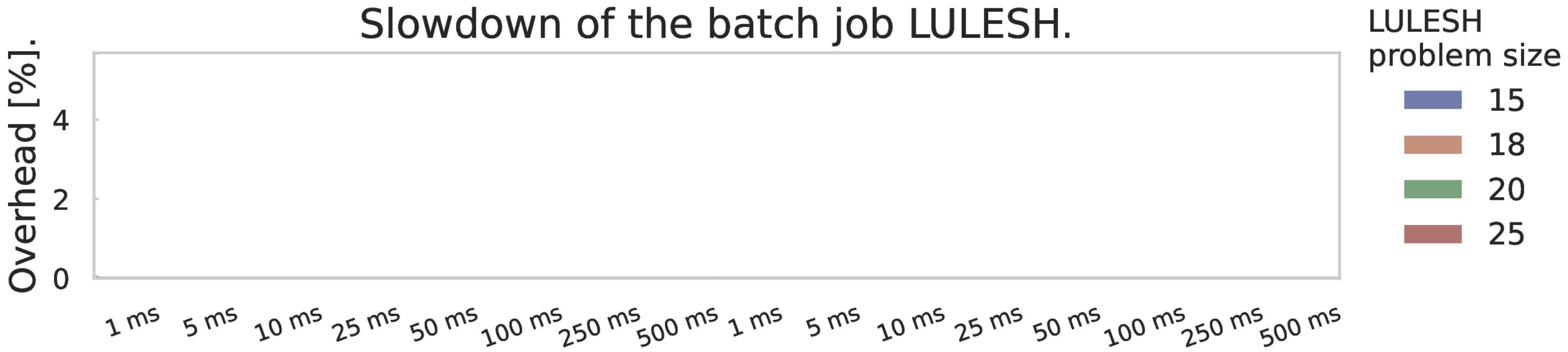
32 out of 36 cores allocated.

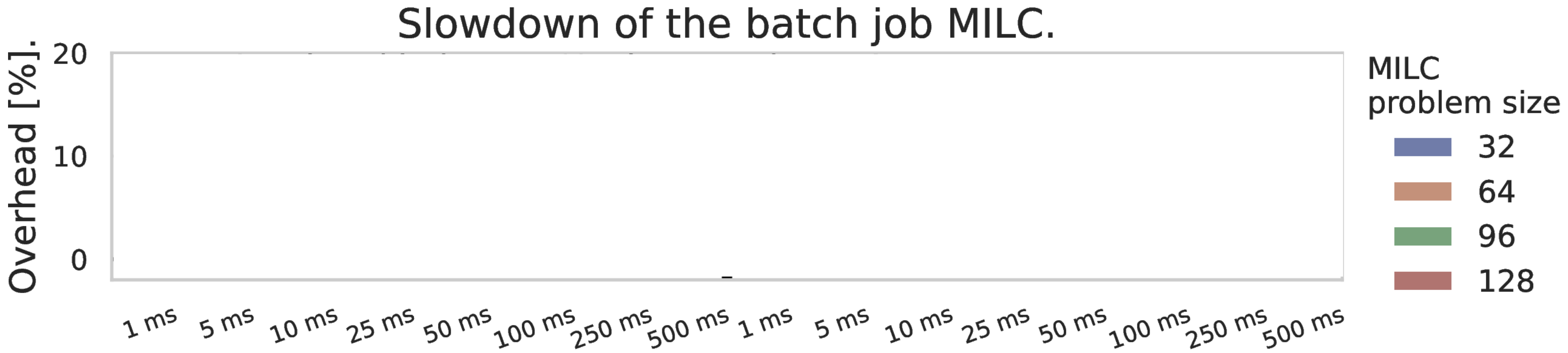
NAS

1 – 4 ranks

Distributed across nodes.

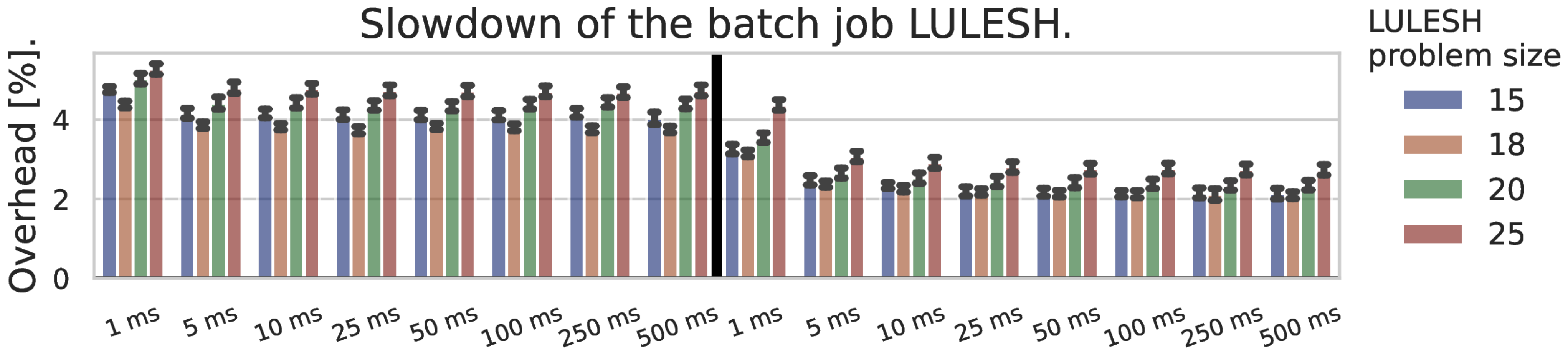
#2 Serving Remote Memory

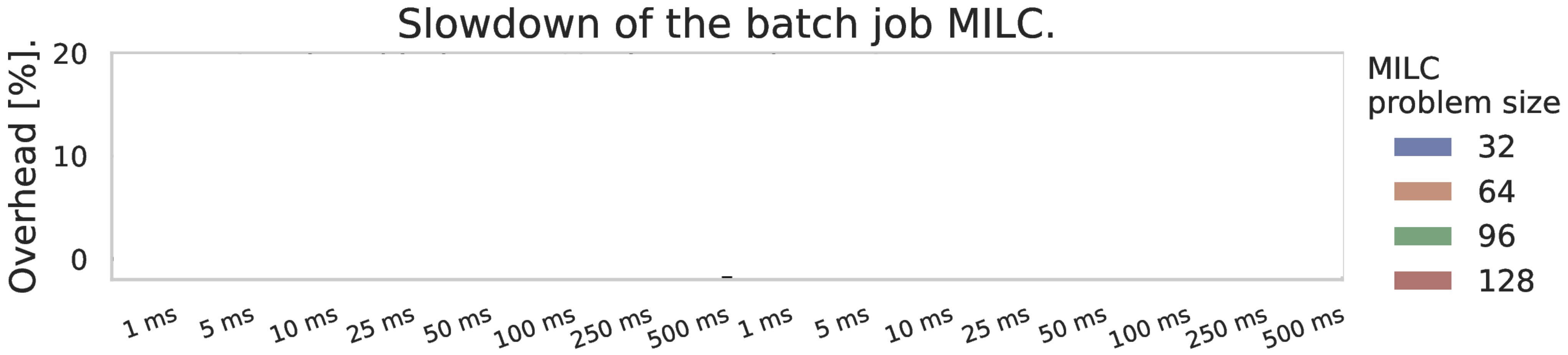




LULESH, MILC – 32 ranks, 1 node, 32 out of 36 cores allocated.

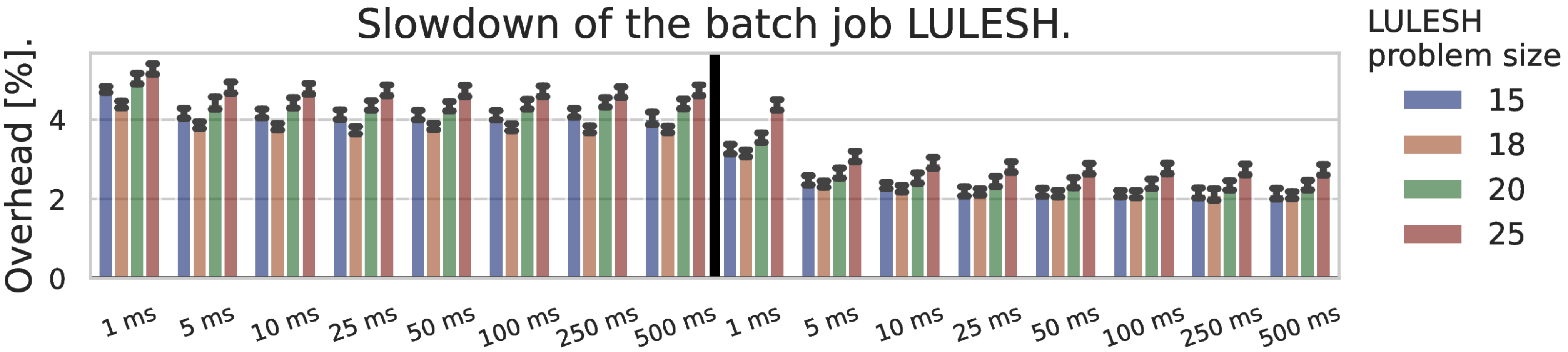
#2 Serving Remote Memory

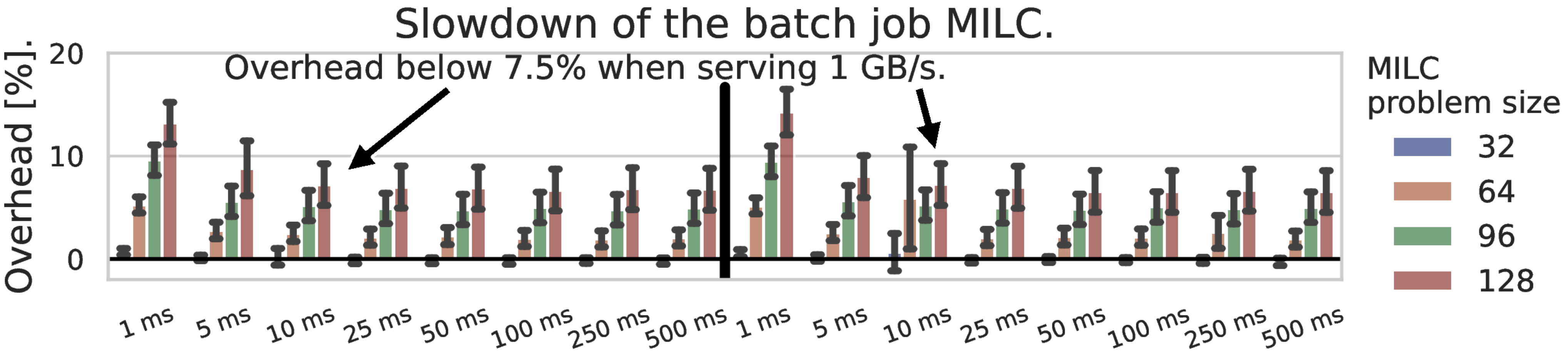




LULESH, MILC – 32 ranks, 1 node, 32 out of 36 cores allocated.

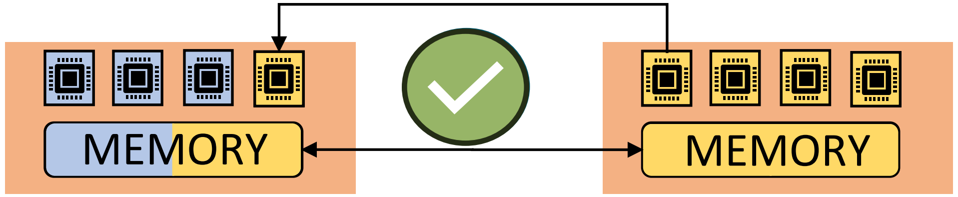
#2 Serving Remote Memory

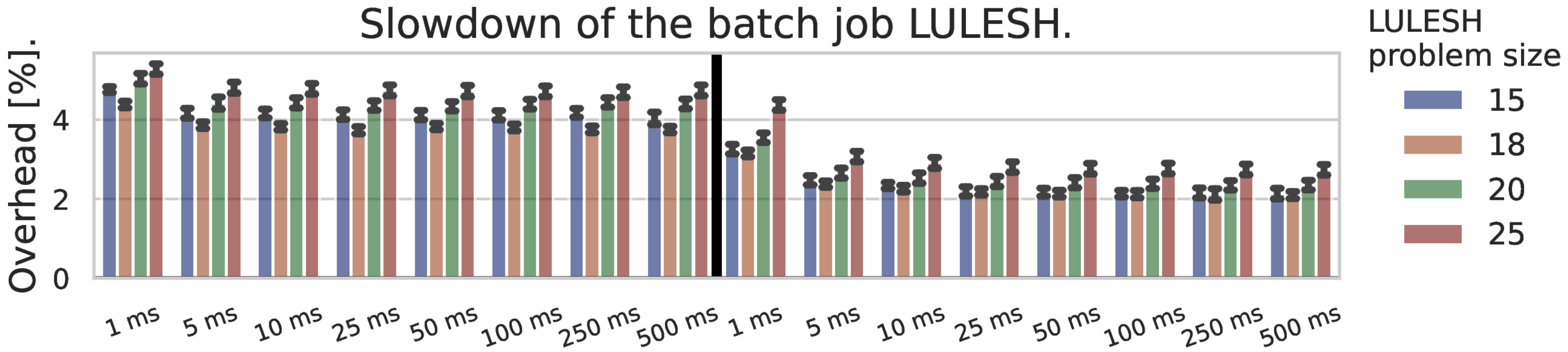


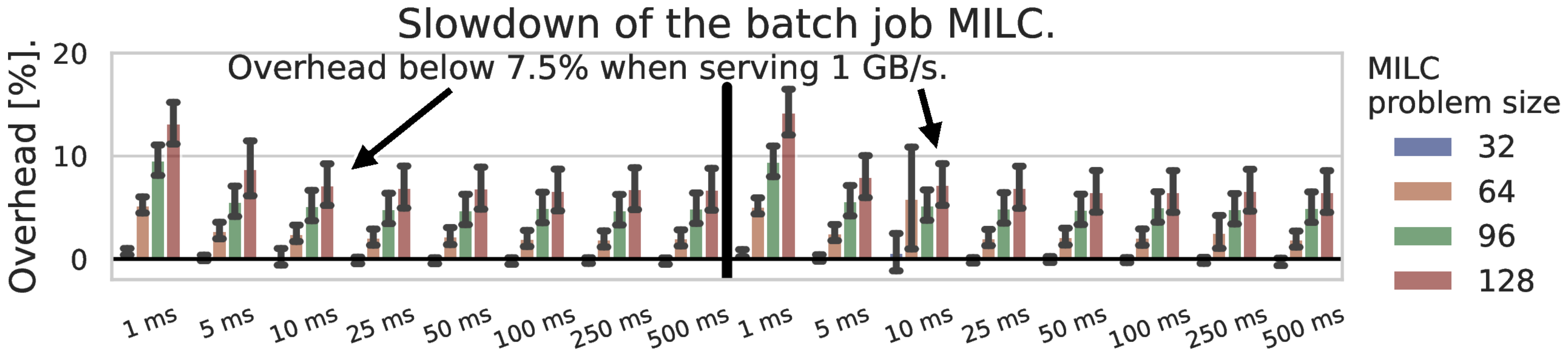


LULESH, MILC – 32 ranks, 1 node, 32 out of 36 cores allocated.

#2 Serving Remote Memory

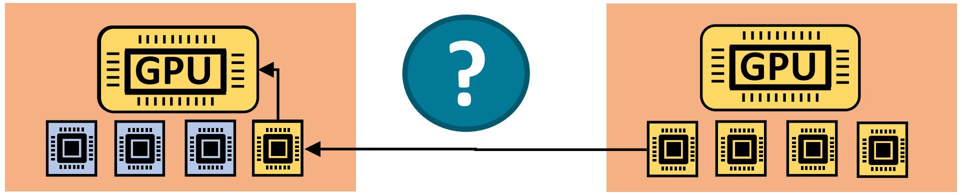


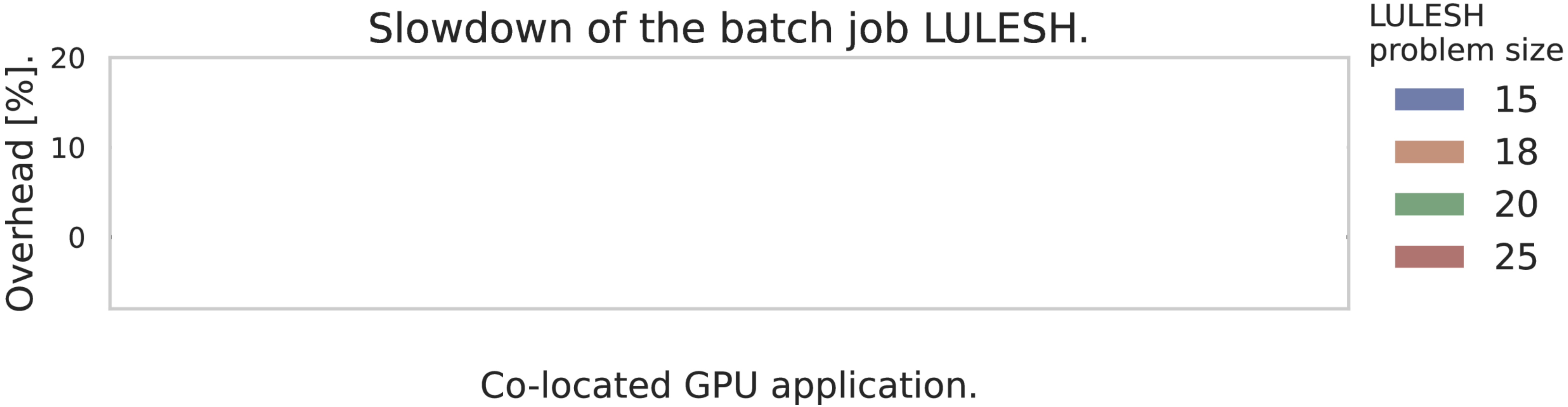




LULESH, MILC – 32 ranks, 1 node, 32 out of 36 cores allocated.

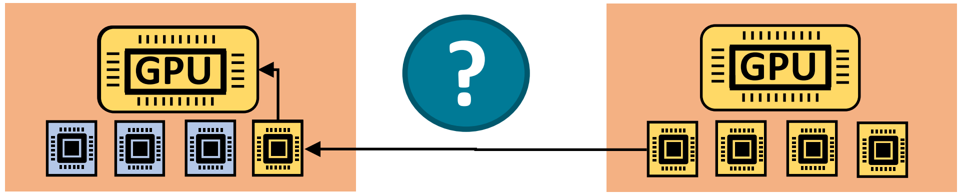
#3 Co-locating GPU and CPU workloads

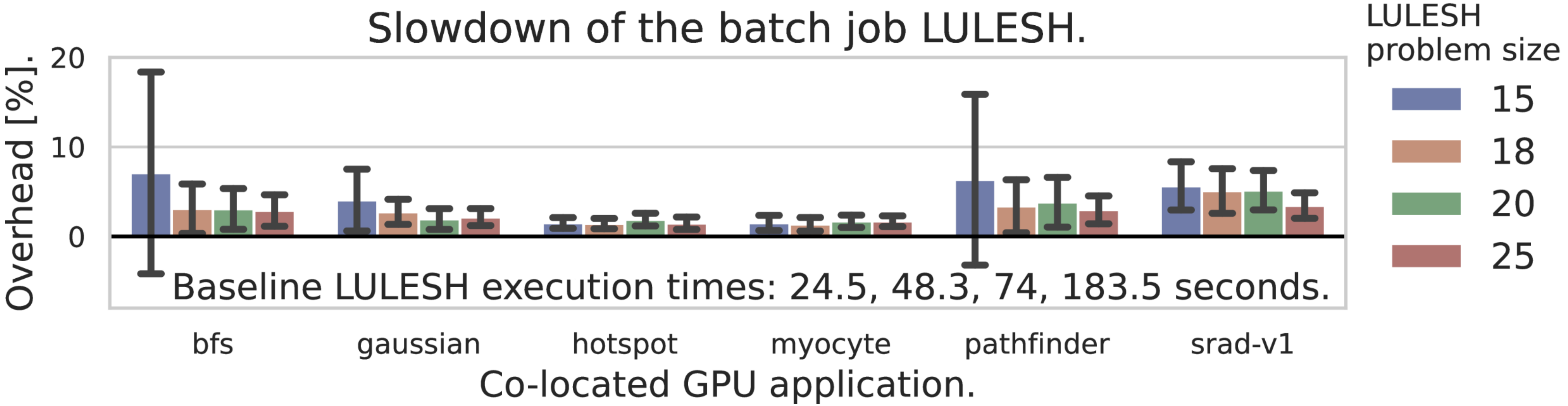




LULESH – 27 ranks, 3 nodes, 9 out of 12 cores allocated.
Rodinia – 1 MPI rank, 1 GPU.

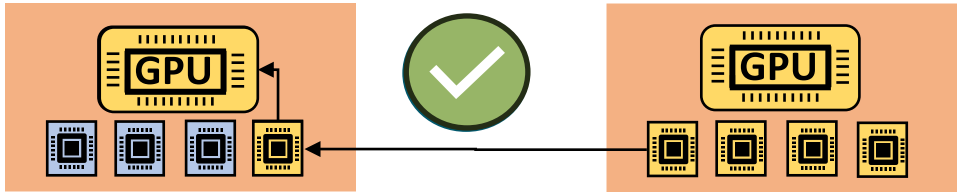
#3 Co-locating GPU and CPU workloads

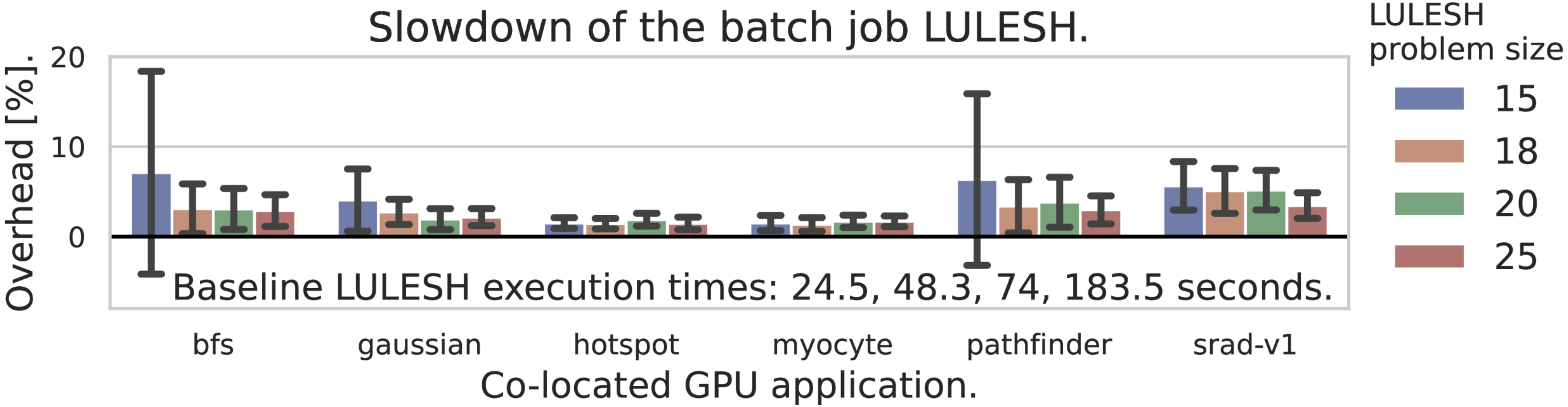




LULESH – 27 ranks, 3 nodes, 9 out of 12 cores allocated.
Rodinia – 1 MPI rank, 1 GPU.

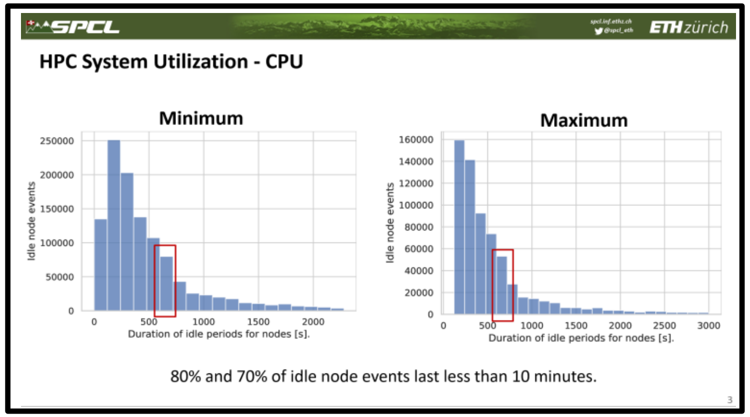
#3 Co-locating GPU and CPU workloads



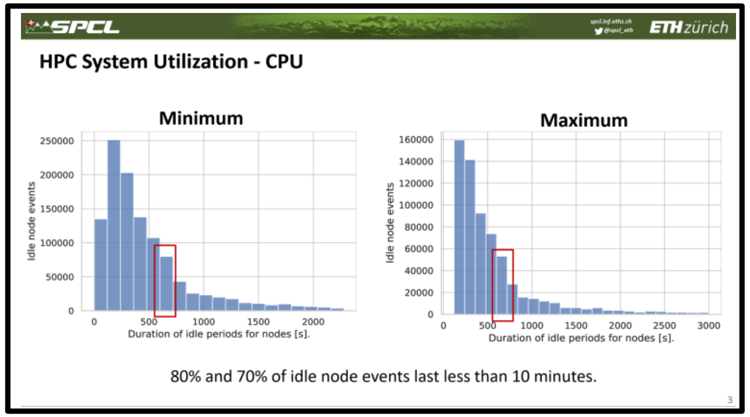


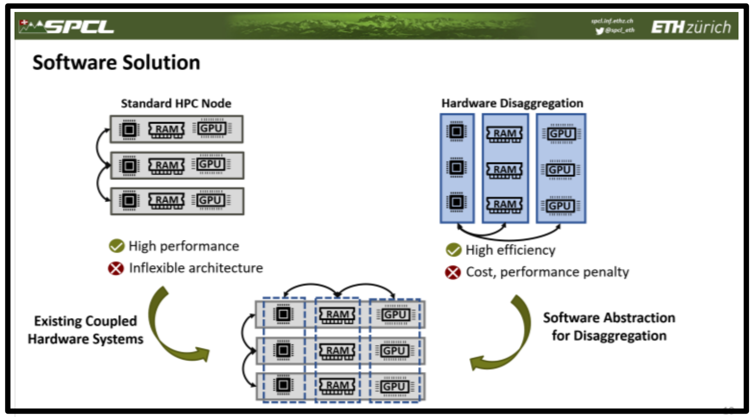
LULESH – 27 ranks, 3 nodes, 9 out of 12 cores allocated.
Rodinia – 1 MPI rank, 1 GPU.

Summary

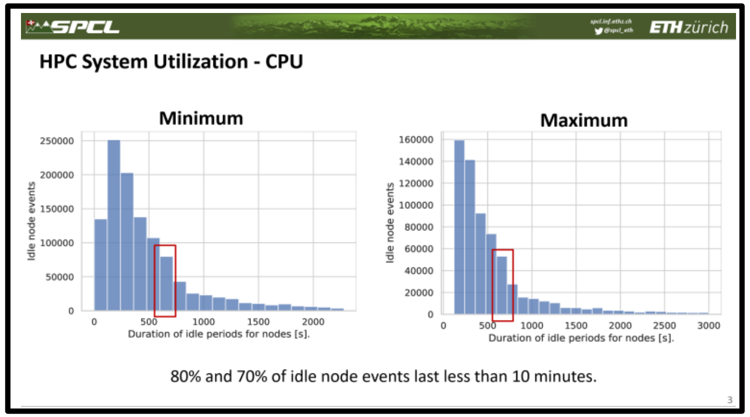

HPC System Utilization - CPU
Minimum
Maximum
Idle node events
Duration of idle periods for nodes [s].
80% and 70% of idle node events last less than 10 minutes.

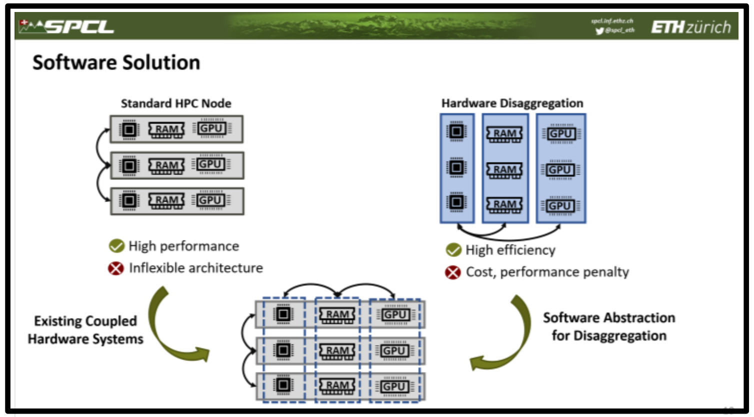
Summary

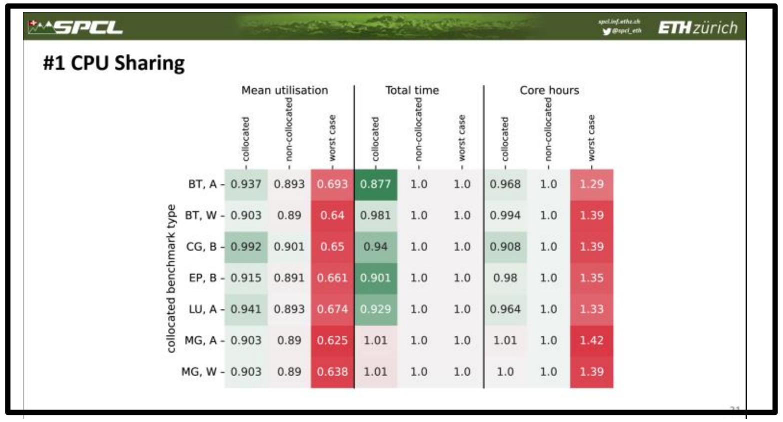

HPC System Utilization - CPU
Minimum
Maximum
Idle node events
Duration of idle periods for nodes [s].
80% and 70% of idle node events last less than 10 minutes.


Software Solution
Standard HPC Node
Hardware Disaggregation
RAM
GPU
High performance
Inflexible architecture
High efficiency
Cost, performance penalty
Existing Coupled Hardware Systems
Software Abstraction for Disaggregation

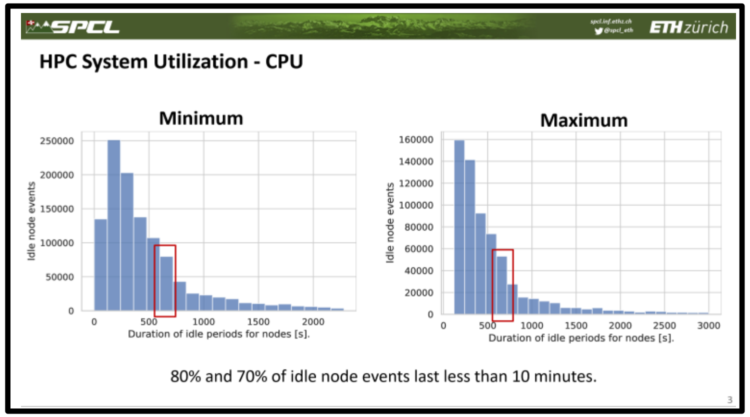
Summary

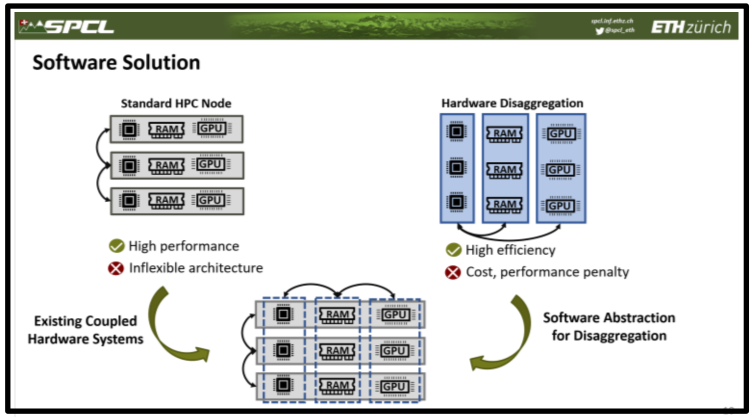

HPC System Utilization - CPU
Minimum
Maximum
Idle node events
Duration of idle periods for nodes [s].
80% and 70% of idle node events last less than 10 minutes.

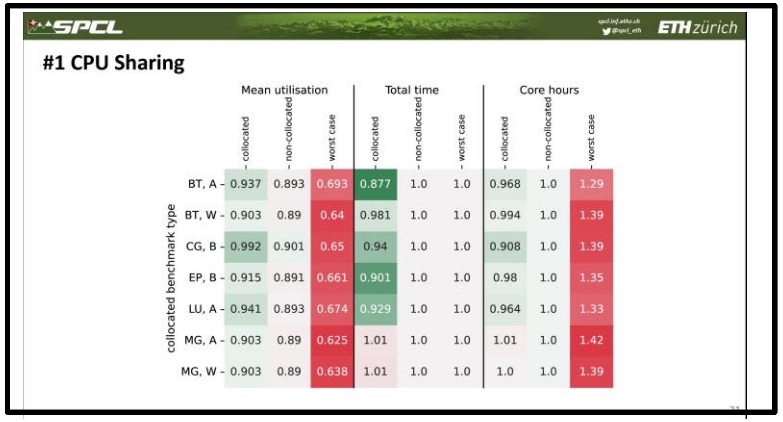

Software Solution
Standard HPC Node
Hardware Disaggregation
RAM
GPU
High performance
Inflexible architecture
High efficiency
Cost, performance penalty
Existing Coupled Hardware Systems
Software Abstraction for Disaggregation


#1 CPU Sharing
Mean utilisation
Total time
Core hours
collocated
non-collocated
worst case
collocated benchmark type
BT, A - 0.937 0.893 0.693 0.877 1.0 1.0 0.968 1.0 1.29
BT, W - 0.903 0.89 0.64 0.981 1.0 1.0 0.994 1.0 1.39
CG, B - 0.992 0.901 0.65 0.94 1.0 1.0 0.908 1.0 1.39
EP, B - 0.915 0.891 0.661 0.901 1.0 1.0 0.98 1.0 1.35
LU, A - 0.941 0.893 0.674 0.929 1.0 1.0 0.964 1.0 1.33
MG, A - 0.903 0.89 0.625 1.01 1.0 1.0 1.01 1.0 1.42
MG, W - 0.903 0.89 0.638 1.01 1.0 1.0 1.0 1.0 1.39

Summary



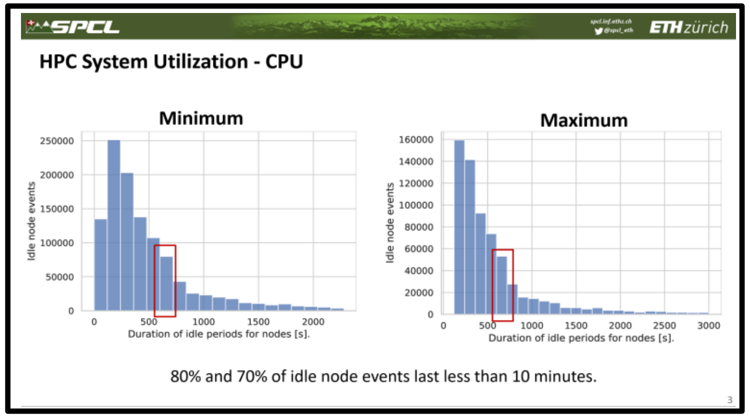


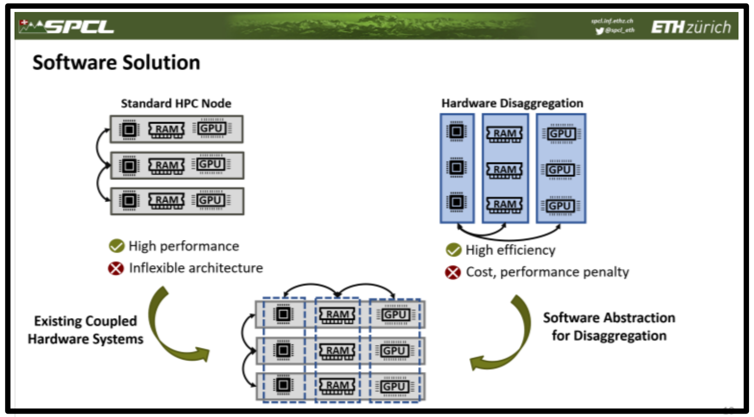


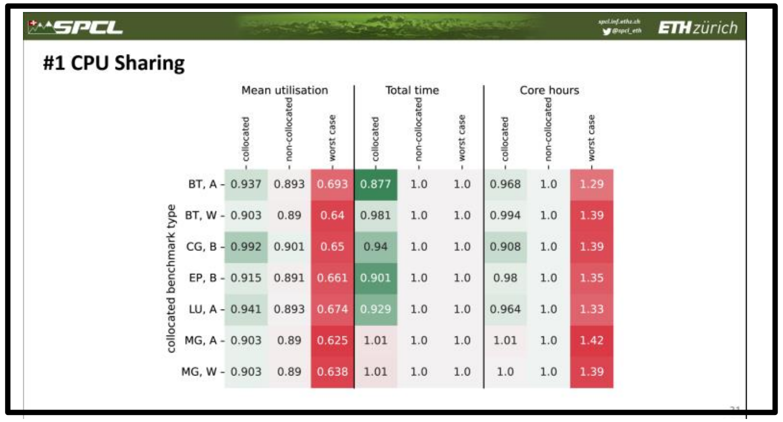
spcl/rFaaS

youtube.com/@spcl twitter.com/spcl_eth spcl.inf.ethz.ch github.com/spcl

Summary







spcl/rFaaS

"the goal of achieving near 100% utilization while supporting a real parallel supercomputing workload is unrealistic"

Scheduling for Parallel Supercomputing: A Historical Perspective of Achievable Utilization

James Patton Jones[1] and Bill Nitzberg[1]

MRJ Technology Solutions
NASA Ames Research Center, M/S 258-6
Moffett Field, CA 94035-1000

jjones@nas.nasa.gov

1999

youtube.com/@spcl twitter.com/spcl_eth spcl.inf.ethz.ch github.com/spcl