LA-UR-23-28509
Approved for public release; distribution is unlimited.

Title: SYSTEM AND JOB SCHEDULING SIMULATION FOR ENHANCING PRODUCTION HPC

Author(s): Hafener, Vivian Erica
Senator, Steven Terry
Jones, William M.
Walker, Craig S.
Debardeleben, Nathan A.

Intended for: Slurm User Group, 2023-09-11/2023-09-13 (Provo, Utah, United States)

Issued: 2023-07-26

SYSTEM AND JOB SCHEDULING SIMULATION FOR ENHANCING PRODUCTION HPC

Principal Developers: **Craig Walker** (CCU staff), **Vivian Hafener** (LANL staff, RIT student, presenter)

Former Contributors: Nicklaus Przybylski (LANL post-bacc / NGP), Braeden Slade (LANL post-bacc / storage)
Gavin Bailey (Boeing / PhD student)

LANL POCs: **Steve Senator** (HPC-ENV) and **Nathan DeBardeleben** (HPC-DES)

CCU Faculty Mentor: **William M. Jones**

Work funded under LANL contract #C3132/SR489790212

Executed: 11-JAN-2023

September 12-13th, 2023
Slurm User Group
Brigham Young University

COASTAL CAROLINA UNIVERSITY

ROCHESTER INSTITUTE OF TECHNOLOGY R·I·T 1829

Los Alamos NATIONAL LABORATORY EST. 1943

Bottomline Upfront – Takeaways

Bottomline Upfront – Takeaways

- Specific experiments with specific LANL workloads
  - Resilience and reliability
  - Node sharing / packing
  - DST scheduling

LANL CTS-1 GRIZZLY

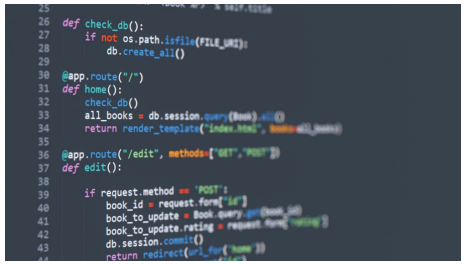
Bottomline Upfront – Takeaways

- Specific experiments with specific LANL workloads
  - Resilience and reliability
  - Node sharing / packing
  - DST scheduling
- Tooling / software
  - Quantify user- and system-centric metrics
  - Applied to systems of arbitrary size
  - Workloads of interest behind the fence

TOP 500 The List.

LANL CTS-1 GRIZZLY

slurm workload manager



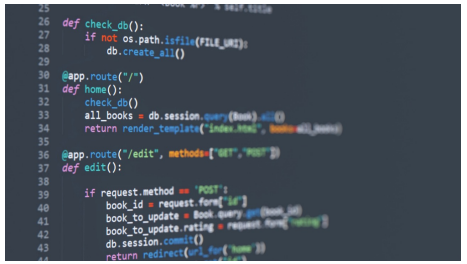
Bottomline Upfront – Takeaways

- Specific experiments with specific LANL workloads
  - Resilience and reliability
  - Node sharing / packing
  - DST scheduling
- Tooling / software
  - Quantify user- and system-centric metrics
  - Applied to systems of arbitrary size
  - Workloads of interest behind the fence
- Interested in feedback on ways to improve / expand our work

TOP 500 The List.

LANL CTS-1 GRIZZLY

slurm workload manager



FEEDBACK

How did this all start?

What is BatSim?

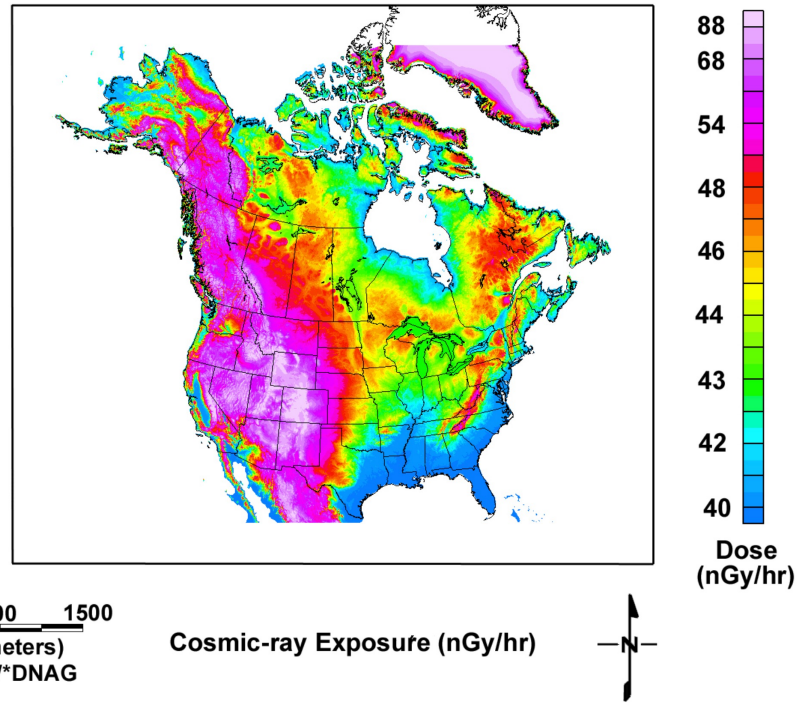
How did we modify BatSim?

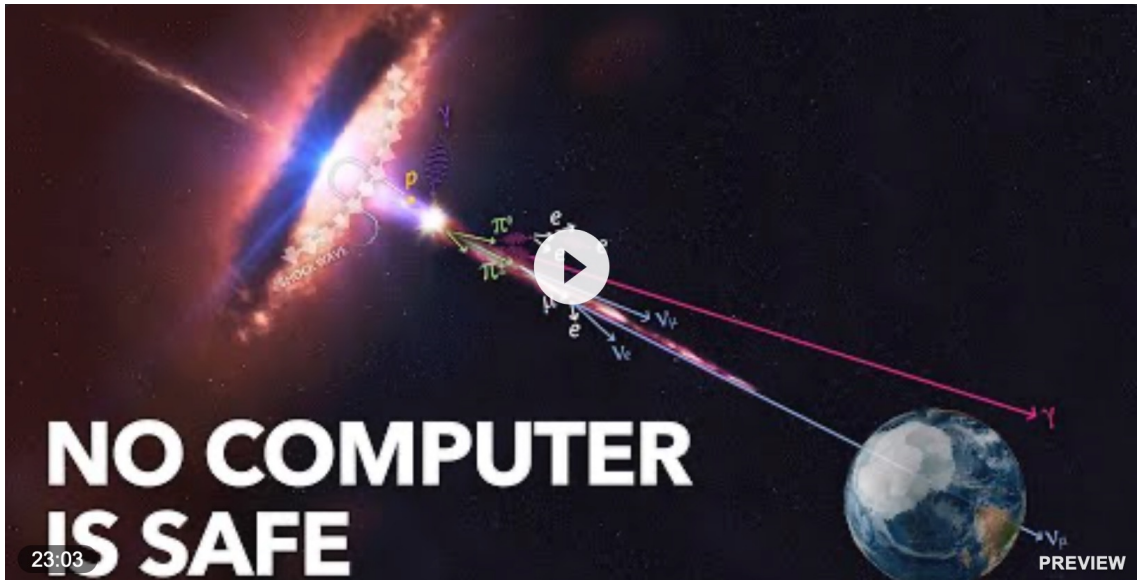
What are some questions that these tools can help answer?

Simulation-based framework to explore impact to performance of large-scale systems due to degraded reliability in DRAM systems

Simulation-based framework to explore impact to performance of large-scale systems due to degraded reliability in DRAM systems

(primarily looking at soft errors)





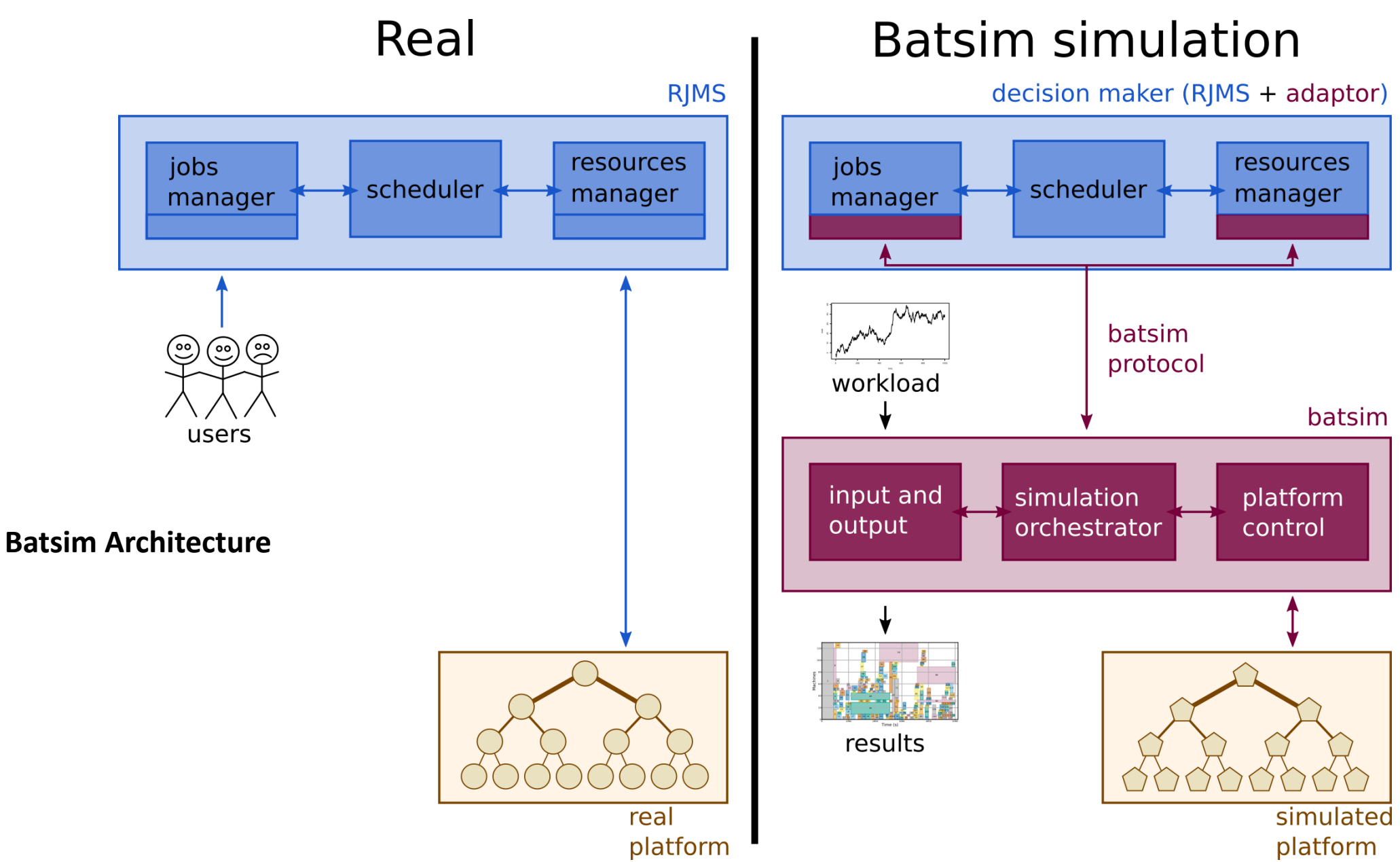
The Universe is Hostile to Computers **by Veritasium**

What is Batsim ?

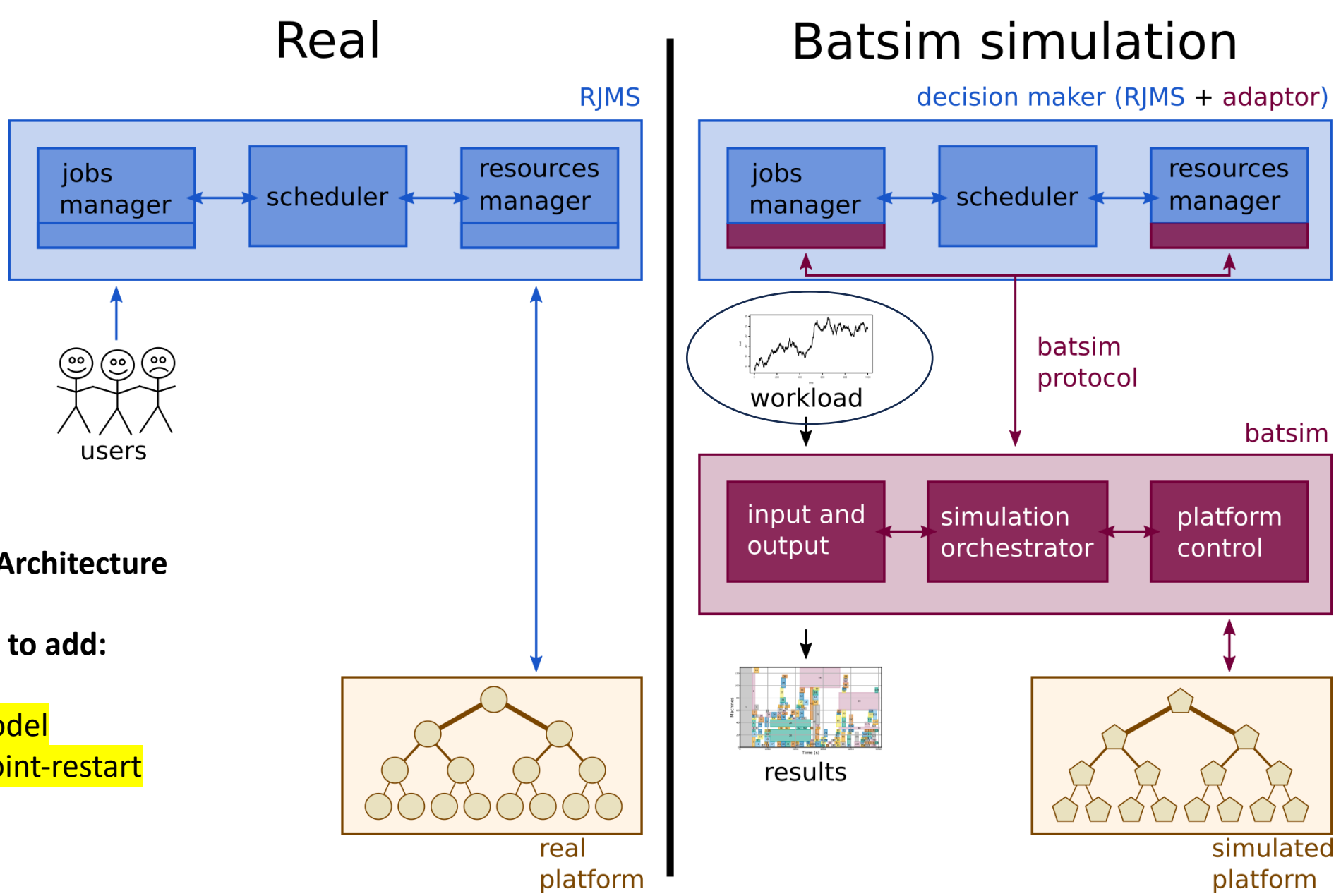
Simulation of Distributed Computer Systems

realistic, Based on SimGrid, multi-processed, **on-going active development**

Pierre-François Dutot, Michael Mercier, Millian Poquet, Olivier Richard. "*Batsim: a Realistic Language-Independent Resources and Jobs Management Systems Simulator*," **20th Workshop on Job Scheduling Strategies for Parallel Processing**, May 2016.



Batsim Architecture

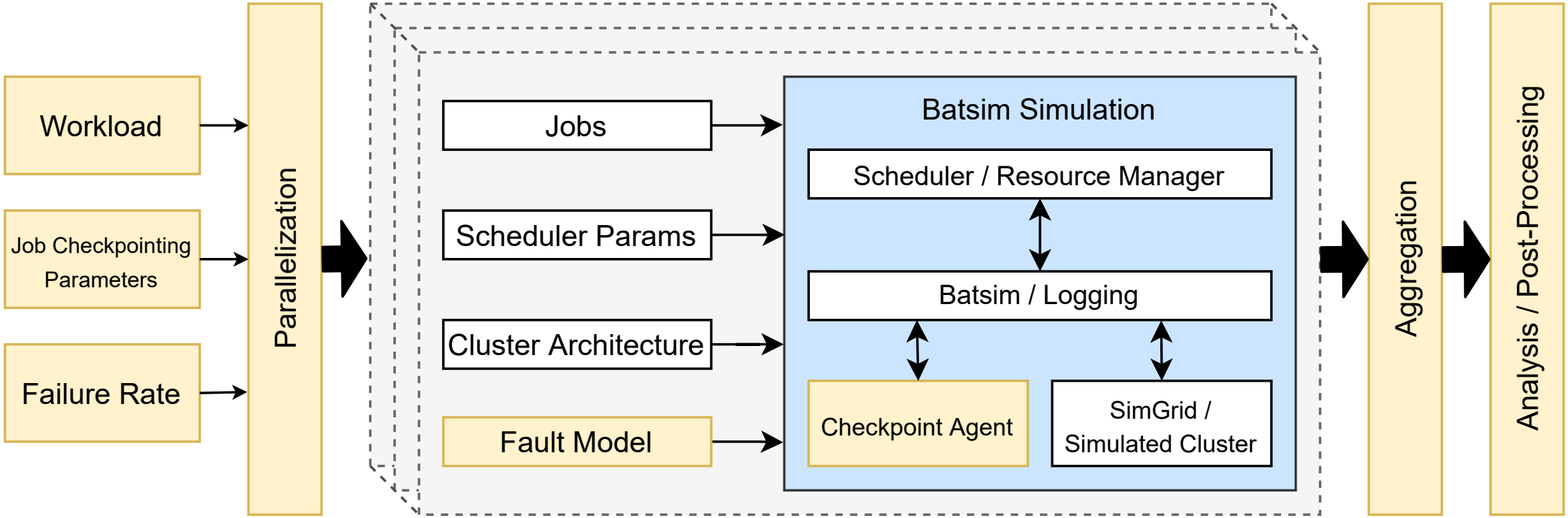


Batsim Architecture

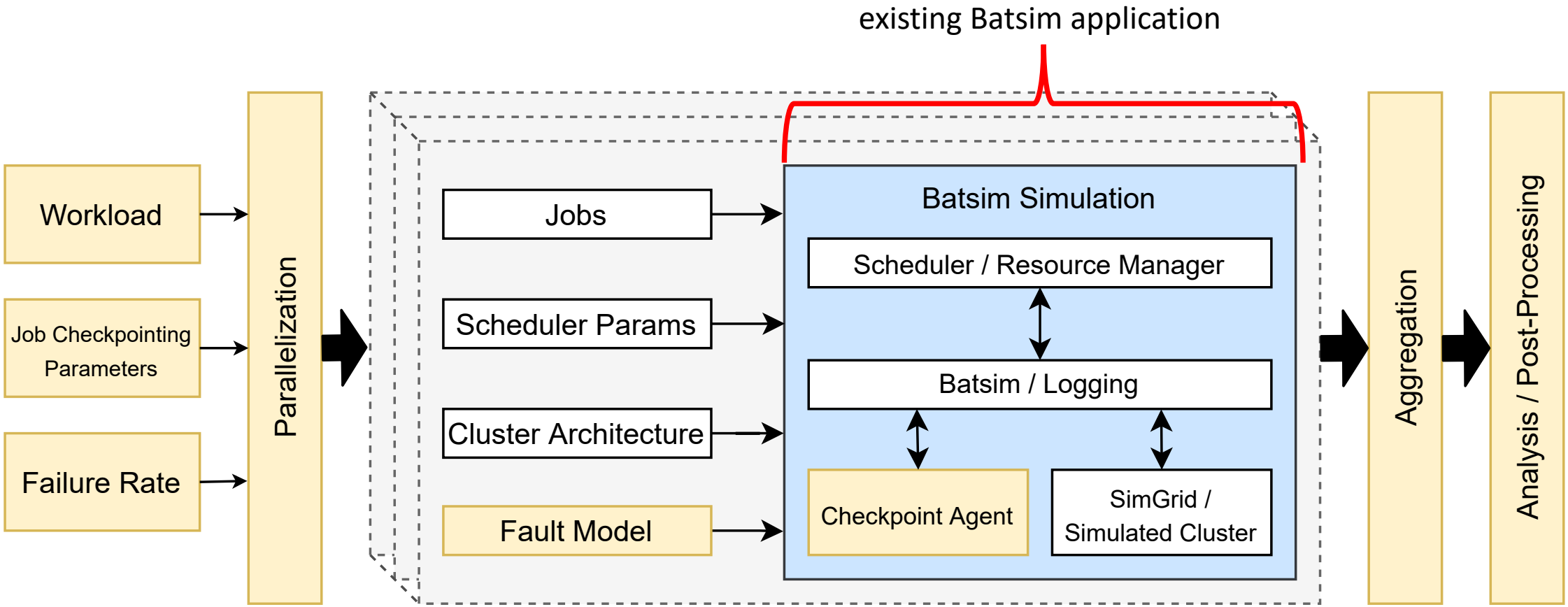
Needed to add:

fault model
checkpoint-restart

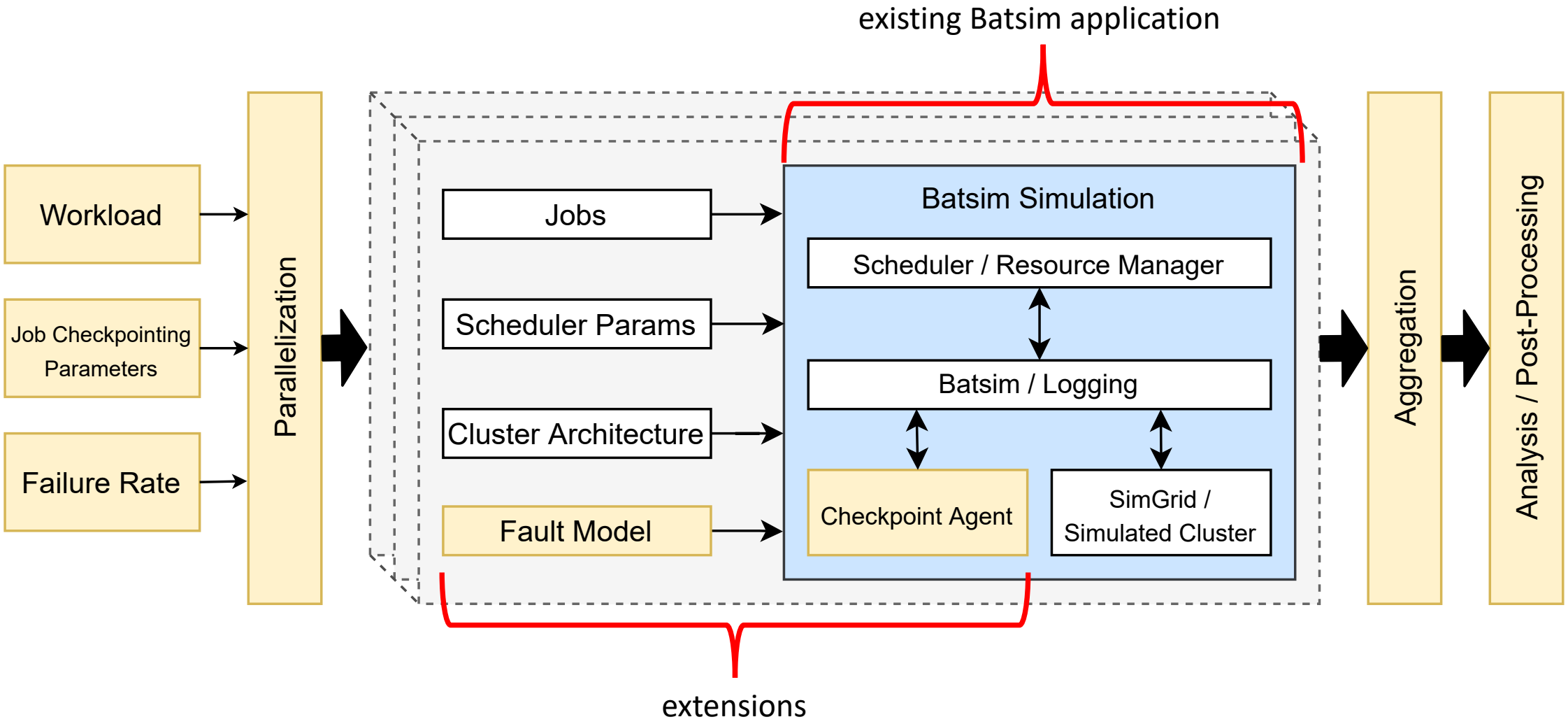
Modifications to Batsim



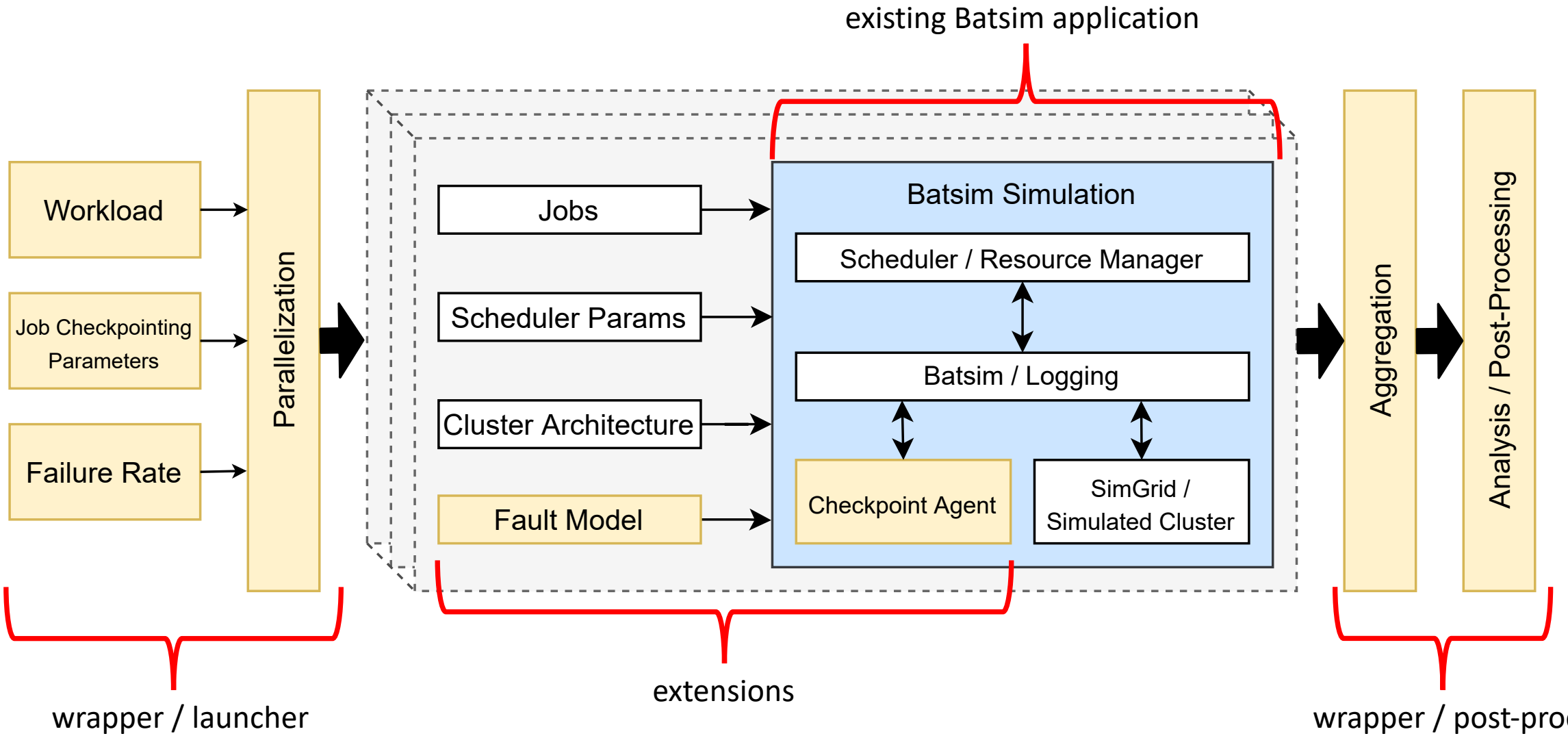
Modifications to Batsim



Modifications to Batsim



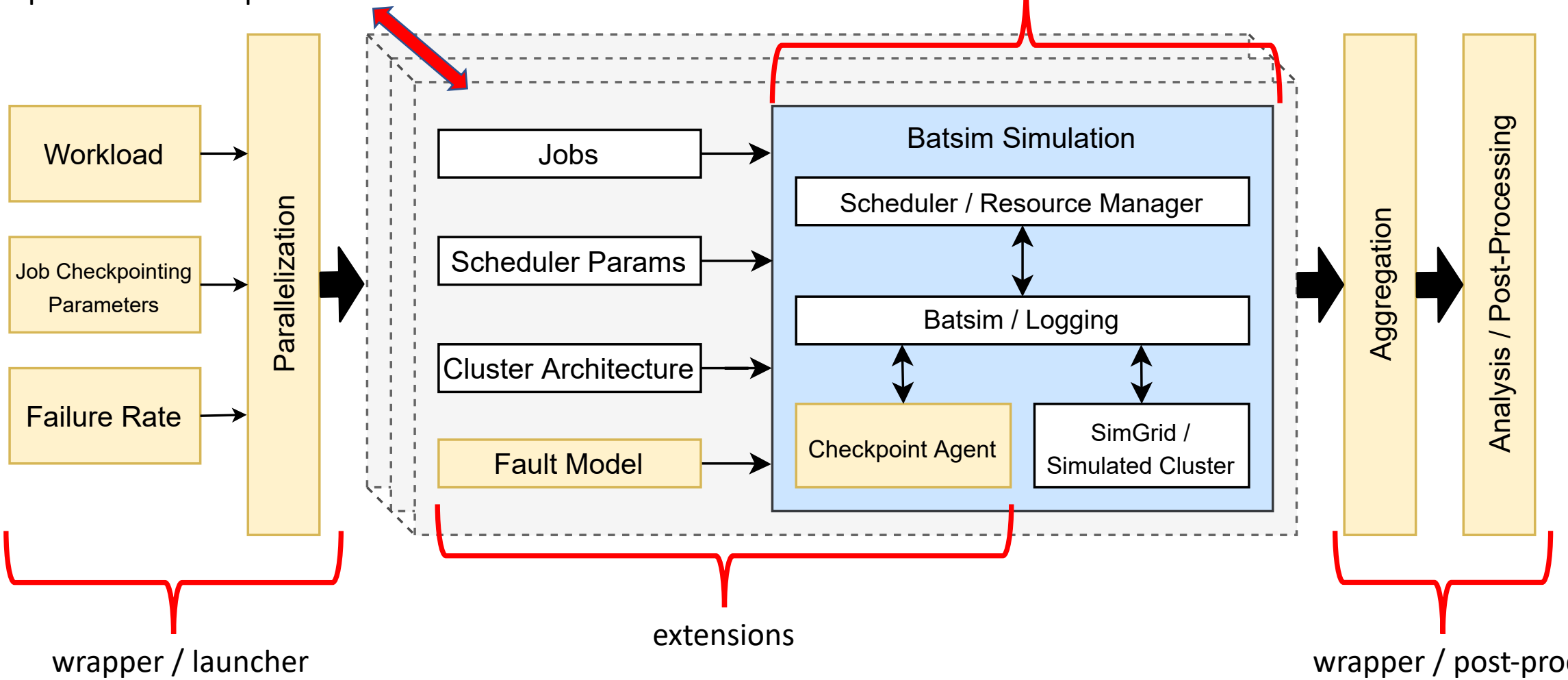
Modifications to Batsim



Modifications to Batsim

parallelize for large-scale Monte Carlo parameter sweeps

existing Batsim application



Experimentation

Experimentation

LANL Cluster Grizzly

Experimentation

LANL Cluster Grizzly

11 months of real job log data from 2018
180K jobs

Experimentation

LANL Cluster Grizzly

11 months of real job log data from 2018
180K jobs

DESCRIPTION OF INPUT WORKLOADS USED IN PERFORMANCE ANALYSIS

workload	description
WL1	All jobs are one node wide and 24 hours long. This represents the analog of WL6, where all jobs span the entire cluster.
WL2	Jobs are based on 11 months of data from a capacity cluster at with 1490 nodes. This workload contained very few large jobs, with roughly 48% of the jobs requiring one node, and less than one hour of wall time, with the remaining jobs distributed in duration and width.
WL3	Job widths are uniformly distributed from one to size of cluster, with durations also uniformly distributed from one to 24 hours.
WL4	Job widths are uniformly distributed from 32 to size of cluster, with durations from 6 to 24 hours.
WL5	Job widths are divided into two bins: 512 or the entire cluster, 1490, with durations ranging from one to 24 hours, according to the distributions of durations present in the original WL2 workload.
WL6	Jobs are all 1490 nodes wide with durations of 24 hours. This represents a "worst-case" scenario from a reliability point of view, as any node failure will result in a job failure that spans the entire cluster.

Experimentation

LANL Cluster Grizzly

11 months of real job log data from 2018
180K jobs
very few “large jobs”

explore a range of scenarios
capability to capacity

DESCRIPTION OF INPUT WORKLOADS USED IN PERFORMANCE ANALYSIS

workload	description
WL1	All jobs are one node wide and 24 hours long. This represents the analog of WL6, where all jobs span the entire cluster.
WL2	Jobs are based on 11 months of data from a capacity cluster at with 1490 nodes. This workload contained very few large jobs, with roughly 48% of the jobs requiring one node, and less than one hour of wall time, with the remaining jobs distributed in duration and width.
WL3	Job widths are uniformly distributed from one to size of cluster, with durations also uniformly distributed from one to 24 hours.
WL4	Job widths are uniformly distributed from 32 to size of cluster, with durations from 6 to 24 hours.
WL5	Job widths are divided into two bins: 512 or the entire cluster, 1490, with durations ranging from one to 24 hours, according to the distributions of durations present in the original WL2 workload.
WL6	Jobs are all 1490 nodes wide with durations of 24 hours. This represents a “worst-case” scenario from a reliability point of view, as any node failure will result in a job failure that spans the entire cluster.

Experimentation

LANL Cluster Grizzly

11 months of real job log data from 2018
180K jobs
very few “large jobs”

explore a range of scenarios
capability to capacity

WL1 – purely capacity

DESCRIPTION OF INPUT WORKLOADS USED IN PERFORMANCE ANALYSIS

workload	description
WL1	All jobs are one node wide and 24 hours long. This represents the analog of WL6, where all jobs span the entire cluster.
WL2	Jobs are based on 11 months of data from a capacity cluster at with 1490 nodes. This workload contained very few large jobs, with roughly 48% of the jobs requiring one node, and less than one hour of wall time, with the remaining jobs distributed in duration and width.
WL3	Job widths are uniformly distributed from one to size of cluster, with durations also uniformly distributed from one to 24 hours.
WL4	Job widths are uniformly distributed from 32 to size of cluster, with durations from 6 to 24 hours.
WL5	Job widths are divided into two bins: 512 or the entire cluster, 1490, with durations ranging from one to 24 hours, according to the distributions of durations present in the original WL2 workload.
WL6	Jobs are all 1490 nodes wide with durations of 24 hours. This represents a “worst-case” scenario from a reliability point of view, as any node failure will result in a job failure that spans the entire cluster.

Experimentation

LANL Cluster Grizzly

11 months of real job log data from 2018
180K jobs
very few “large jobs”

explore a range of scenarios
capability to capacity

WL1 – purely capacity
WL6 – purely capability

DESCRIPTION OF INPUT WORKLOADS USED IN PERFORMANCE ANALYSIS

workload	description
WL1	All jobs are one node wide and 24 hours long. This represents the analog of WL6, where all jobs span the entire cluster.
WL2	Jobs are based on 11 months of data from a capacity cluster at with 1490 nodes. This workload contained very few large jobs, with roughly 48% of the jobs requiring one node, and less than one hour of wall time, with the remaining jobs distributed in duration and width.
WL3	Job widths are uniformly distributed from one to size of cluster, with durations also uniformly distributed from one to 24 hours.
WL4	Job widths are uniformly distributed from 32 to size of cluster, with durations from 6 to 24 hours.
WL5	Job widths are divided into two bins: 512 or the entire cluster, 1490, with durations ranging from one to 24 hours, according to the distributions of durations present in the original WL2 workload.
WL6	Jobs are all 1490 nodes wide with durations of 24 hours. This represents a “worst-case” scenario from a reliability point of view, as any node failure will result in a job failure that spans the entire cluster.

Experimentation

LANL Cluster Grizzly

11 months of real job log data from 2018
180K jobs
very few “large jobs”

explore a range of scenarios
capability to capacity

WL1 – purely capacity
WL6 – purely capability
WL{3-5} -- mixture

DESCRIPTION OF INPUT WORKLOADS USED IN PERFORMANCE ANALYSIS

workload	description
WL1	All jobs are one node wide and 24 hours long. This represents the analog of WL6, where all jobs span the entire cluster.
WL2	Jobs are based on 11 months of data from a capacity cluster at with 1490 nodes. This workload contained very few large jobs, with roughly 48% of the jobs requiring one node, and less than one hour of wall time, with the remaining jobs distributed in duration and width.
WL3	Job widths are uniformly distributed from one to size of cluster, with durations also uniformly distributed from one to 24 hours.
WL4	Job widths are uniformly distributed from 32 to size of cluster, with durations from 6 to 24 hours.
WL5	Job widths are divided into two bins: 512 or the entire cluster, 1490, with durations ranging from one to 24 hours, according to the distributions of durations present in the original WL2 workload.
WL6	Jobs are all 1490 nodes wide with durations of 24 hours. This represents a “worst-case” scenario from a reliability point of view, as any node failure will result in a job failure that spans the entire cluster.

Tradeoff between reliability and performance

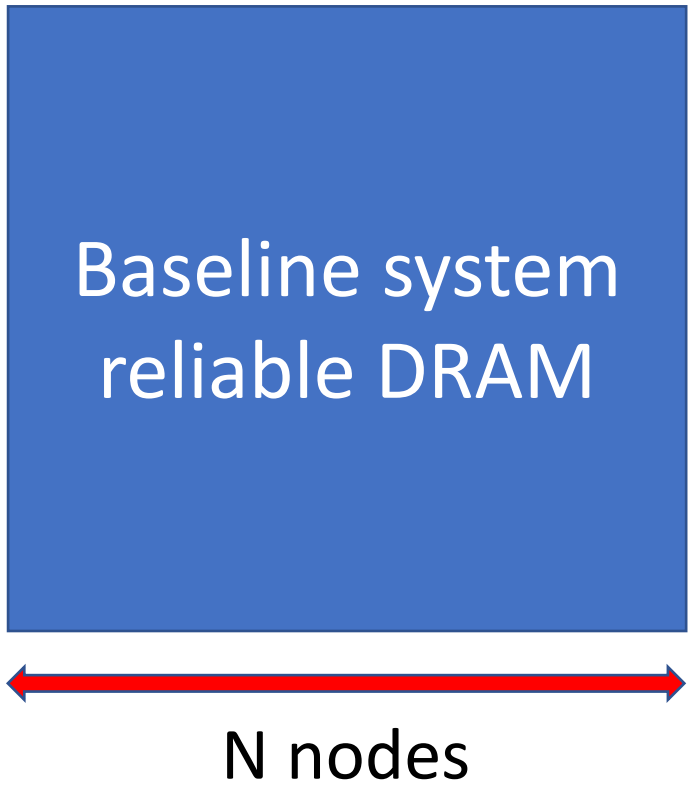
Tradeoff between reliability and performance

1

Baseline system reliable DRAM

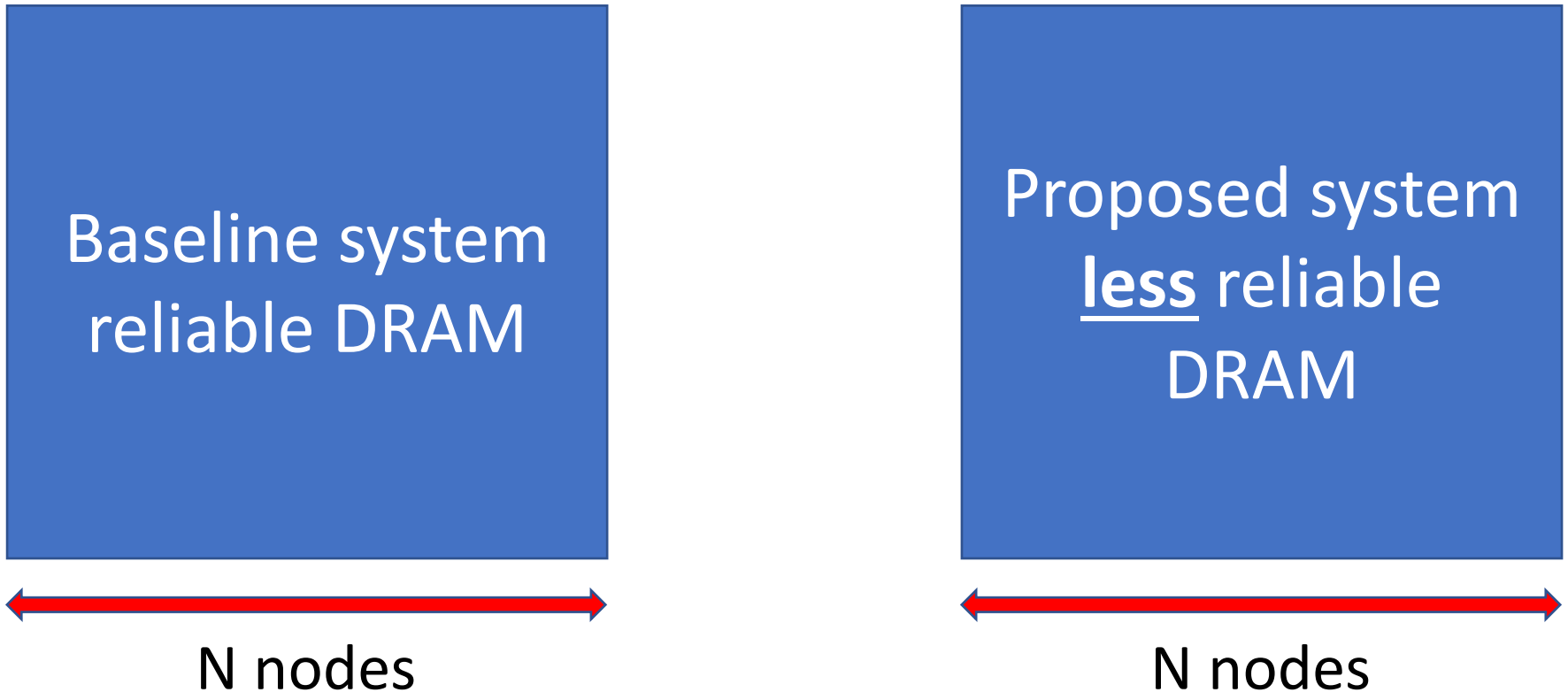
Tradeoff between reliability and performance

1



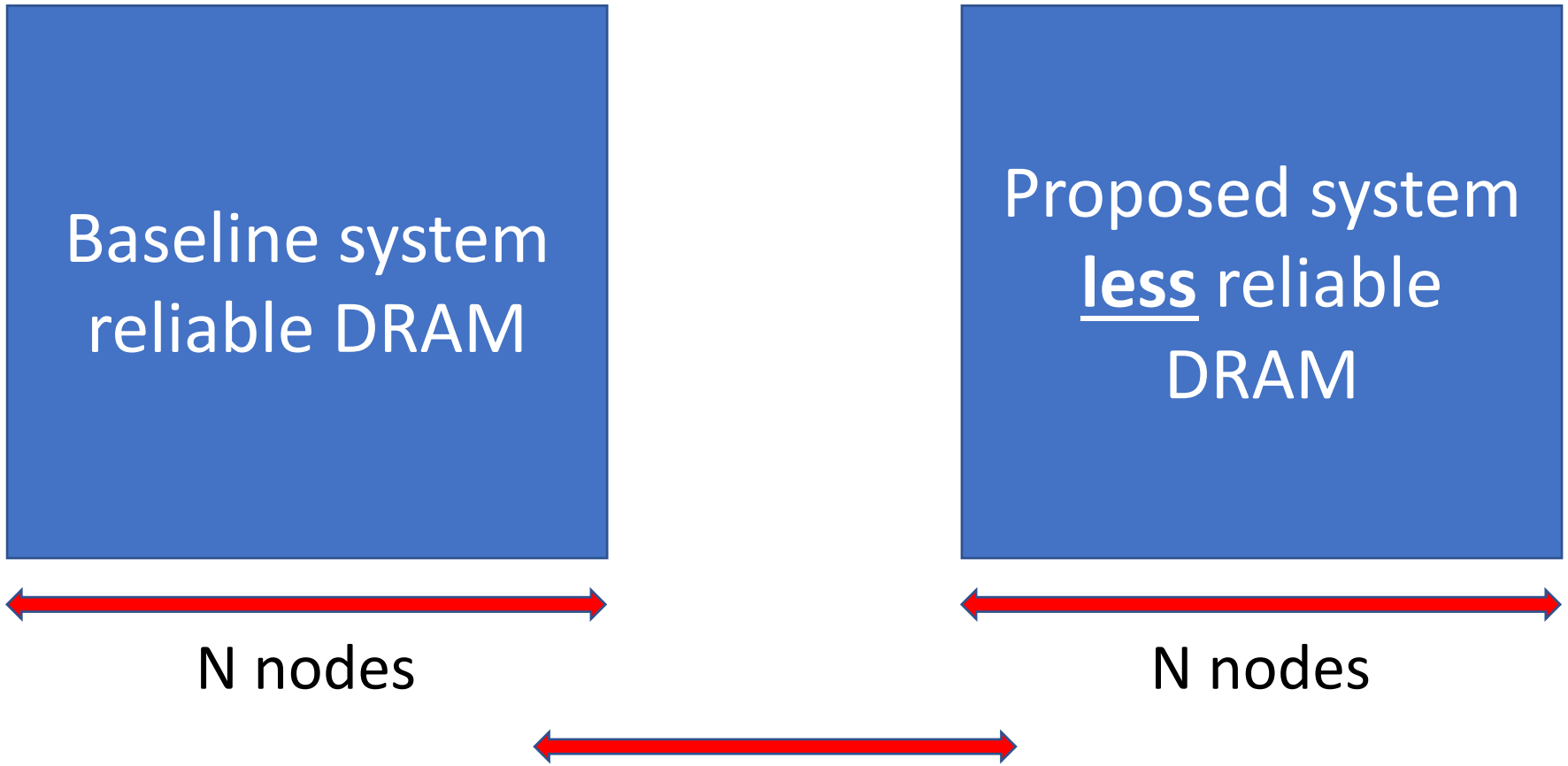
Tradeoff between reliability and performance

1



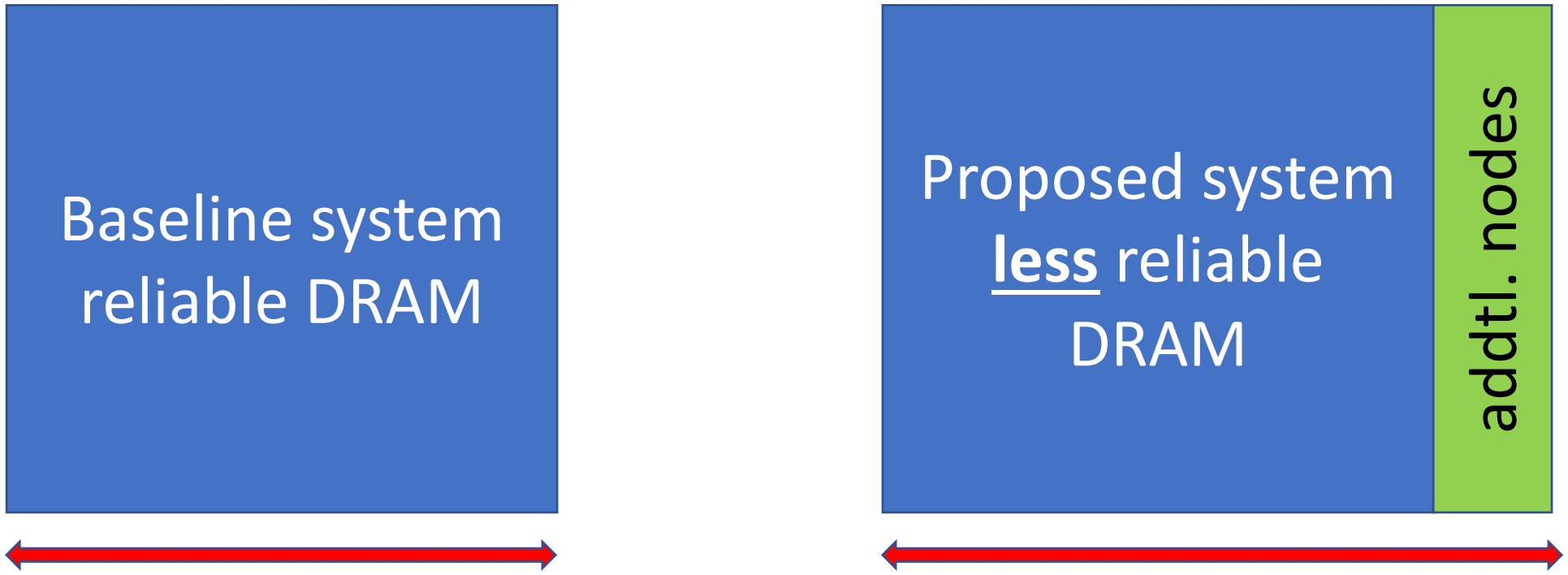
Tradeoff between reliability and performance

1



Tradeoff between reliability and performance

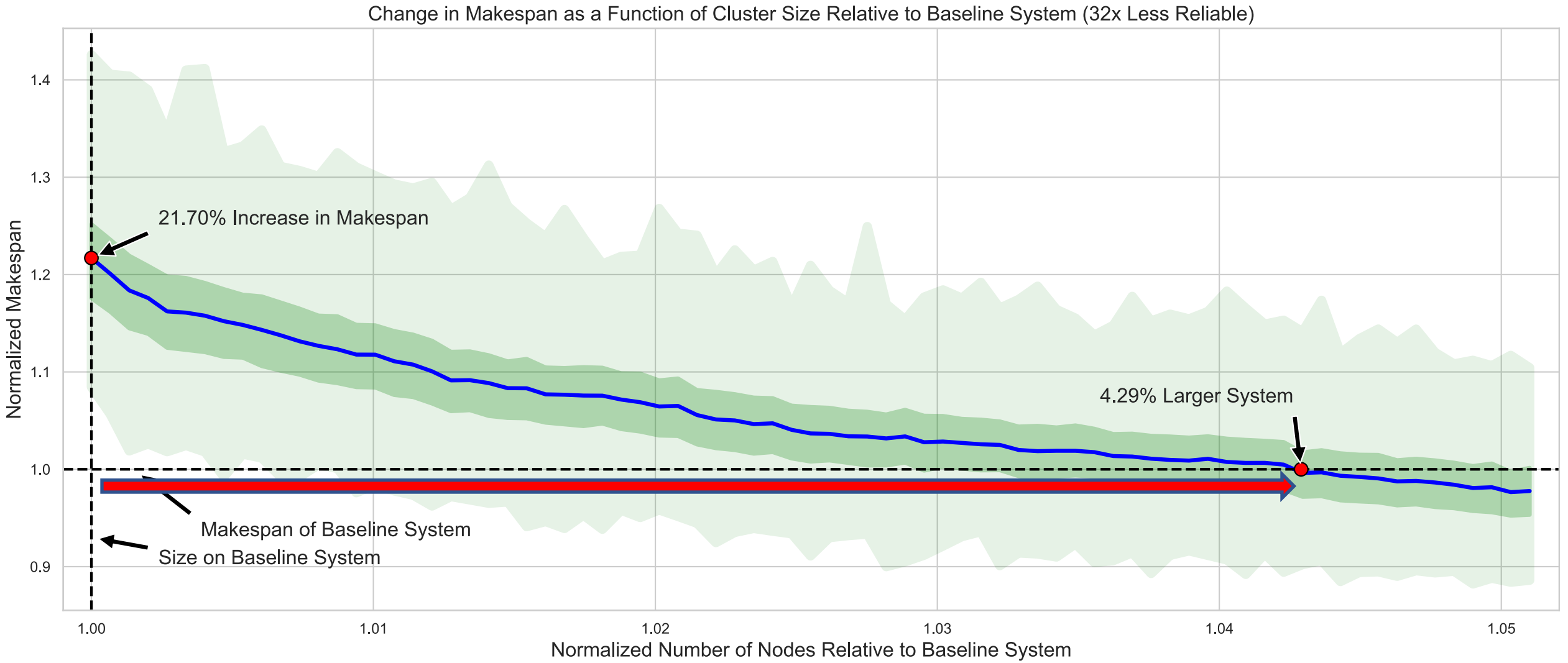
2



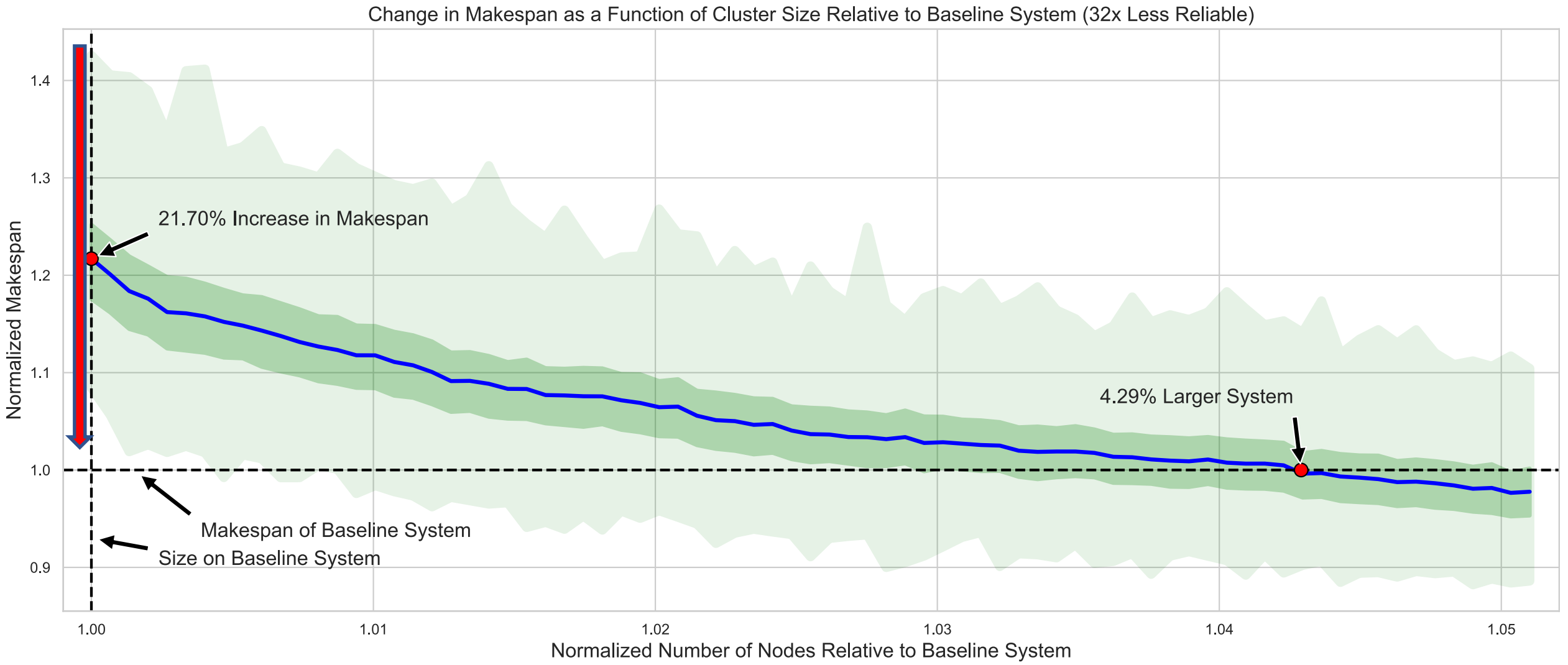
First experimental campaign

- Grizzly WL2 (original) – 1490 nodes
- Assuming 13 day overall MTBF
  - **comparable to Trinity (20K nodes) 1 day MTBF**
- Assume proposed system w/ 32x less reliable DRAM
- Assume entire workload arrives at t = 0
  - evaluate makespan
- Random, treat as Monte Carlo
  - → extremely large number of overall simulations conducted

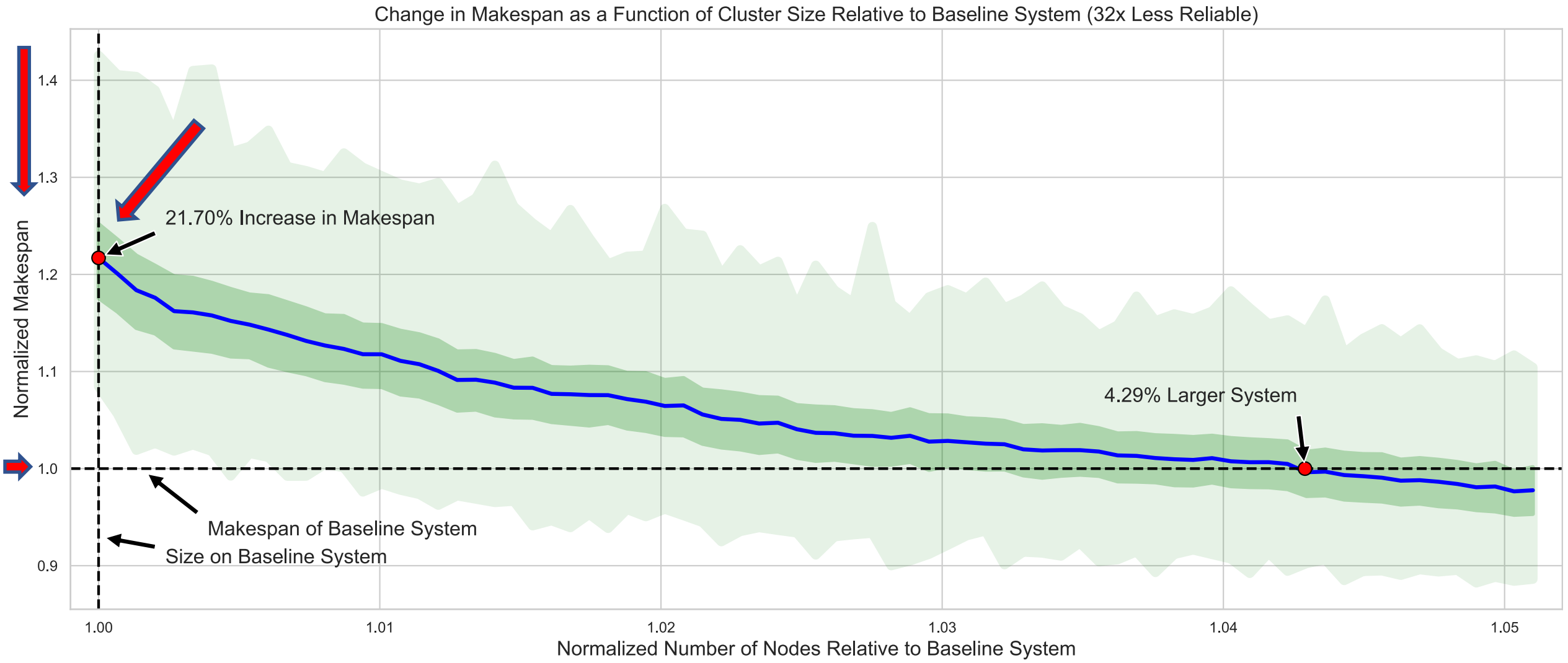
Results


Change in Makespan as a Function of Cluster Size Relative to Baseline System (32x Less Reliable)
Normalized Makespan
1.4
1.3
1.2
1.1
1.0
0.9
21.70% Increase in Makespan
4.29% Larger System
Makespan of Baseline System
Size on Baseline System
1.00
1.01
1.02
1.03
1.04
1.05
Normalized Number of Nodes Relative to Baseline System

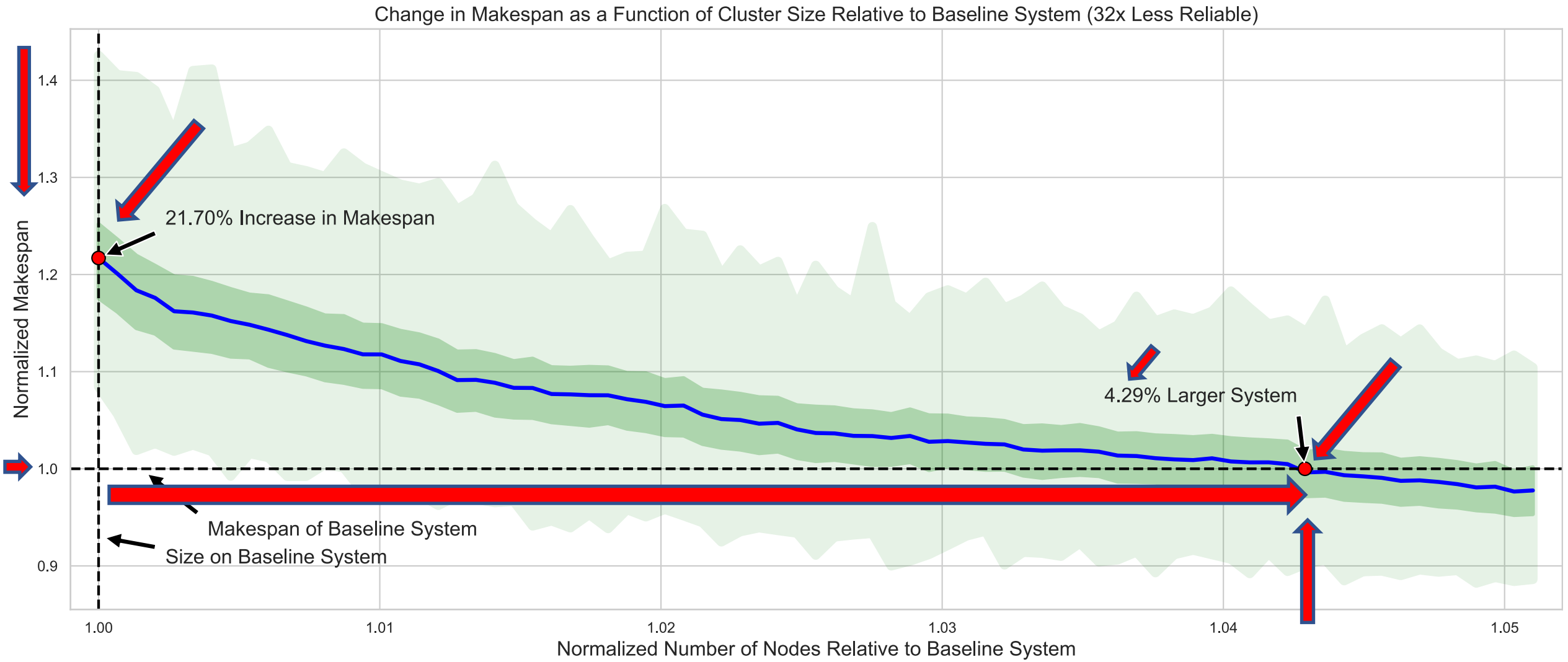
Results


Change in Makespan as a Function of Cluster Size Relative to Baseline System (32x Less Reliable)
Normalized Makespan
1.4
1.3
1.2
1.1
1.0
0.9
21.70% Increase in Makespan
4.29% Larger System
Makespan of Baseline System
Size on Baseline System
1.00
1.01
1.02
1.03
1.04
1.05
Normalized Number of Nodes Relative to Baseline System

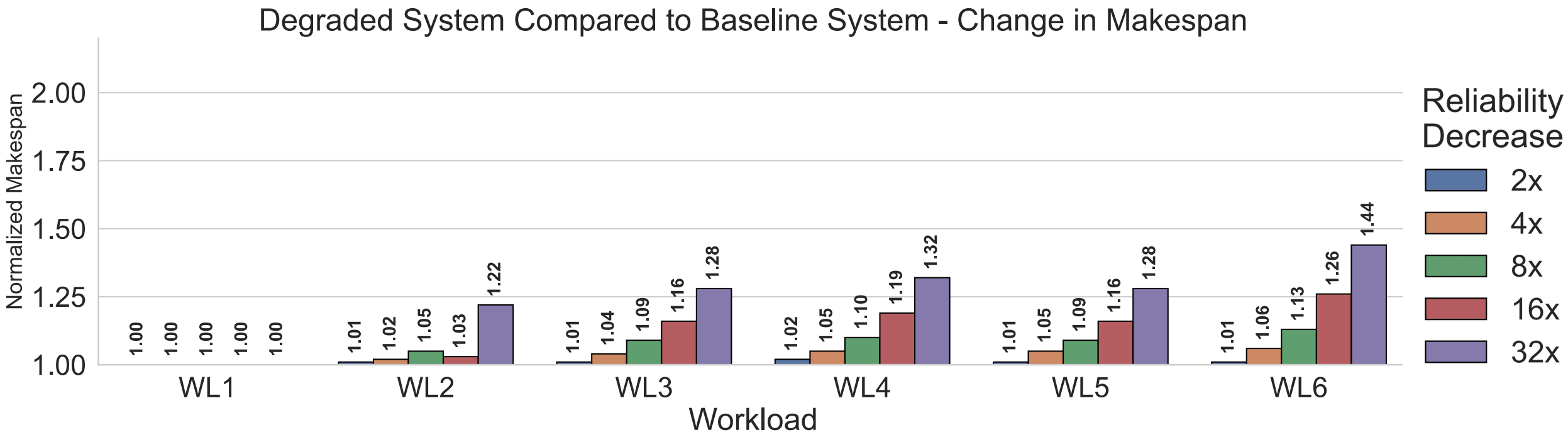
Results


Change in Makespan as a Function of Cluster Size Relative to Baseline System (32x Less Reliable)
Normalized Makespan
1.4
1.3
1.2
1.1
1.0
0.9
21.70% Increase in Makespan
4.29% Larger System
Makespan of Baseline System
Size on Baseline System
1.00
1.01
1.02
1.03
1.04
1.05
Normalized Number of Nodes Relative to Baseline System

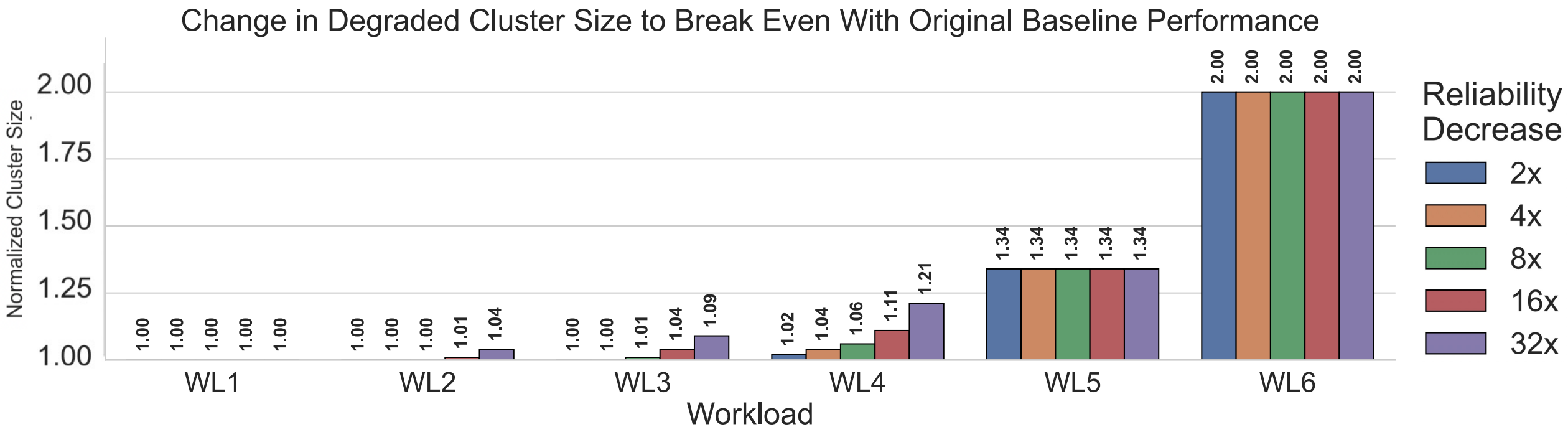
Results


Change in Makespan as a Function of Cluster Size Relative to Baseline System (32x Less Reliable)
Normalized Makespan
1.4
1.3
1.2
1.1
1.0
0.9
21.70% Increase in Makespan
4.29% Larger System
Makespan of Baseline System
Size on Baseline System
1.00
1.01
1.02
1.03
1.04
1.05
Normalized Number of Nodes Relative to Baseline System

Impact to **Makespan**
for different workloads and reliability factors



Impact to size (break-even point) for different workloads and reliability factors

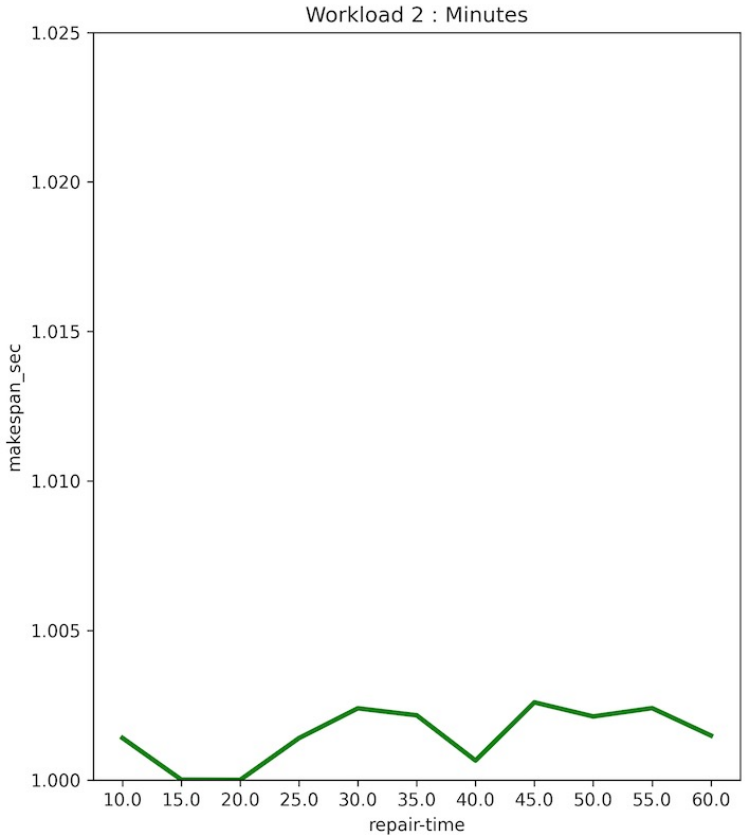


Experimentation

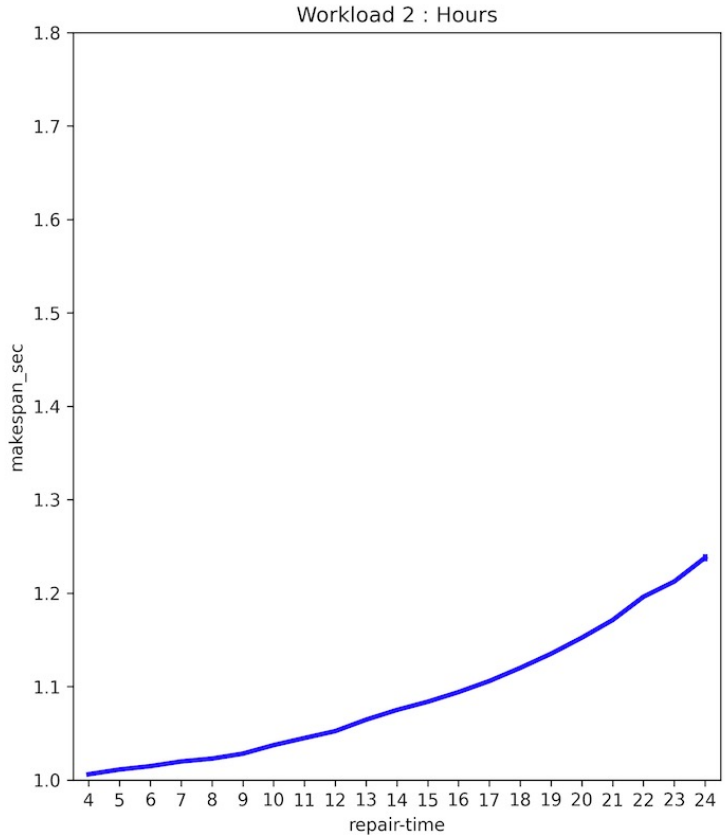
#nodes	1490
SMTTF	~1.6 days
#WLs	6
#jobs / WL	30K *(varied)*
#trials	400

Results (WL2 – Grizzly)

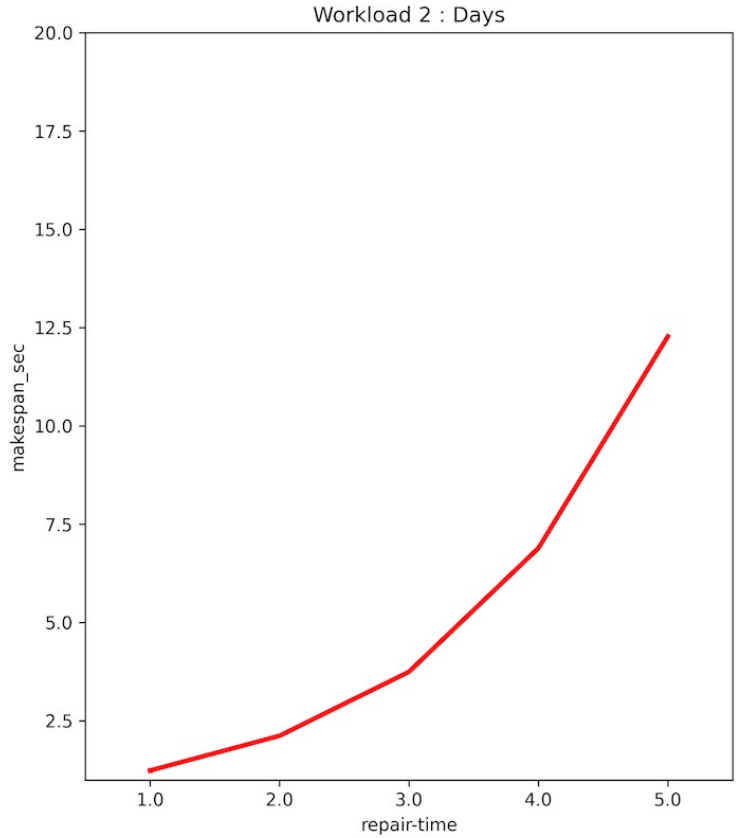
Jobs are based on 11 months of data from a capacity cluster at with 1490 nodes. This workload contained very few large jobs, with roughly 48% of the jobs requiring one node, and less than one hour of wall time, with the remaining jobs distributed in duration and width.



O(minutes)



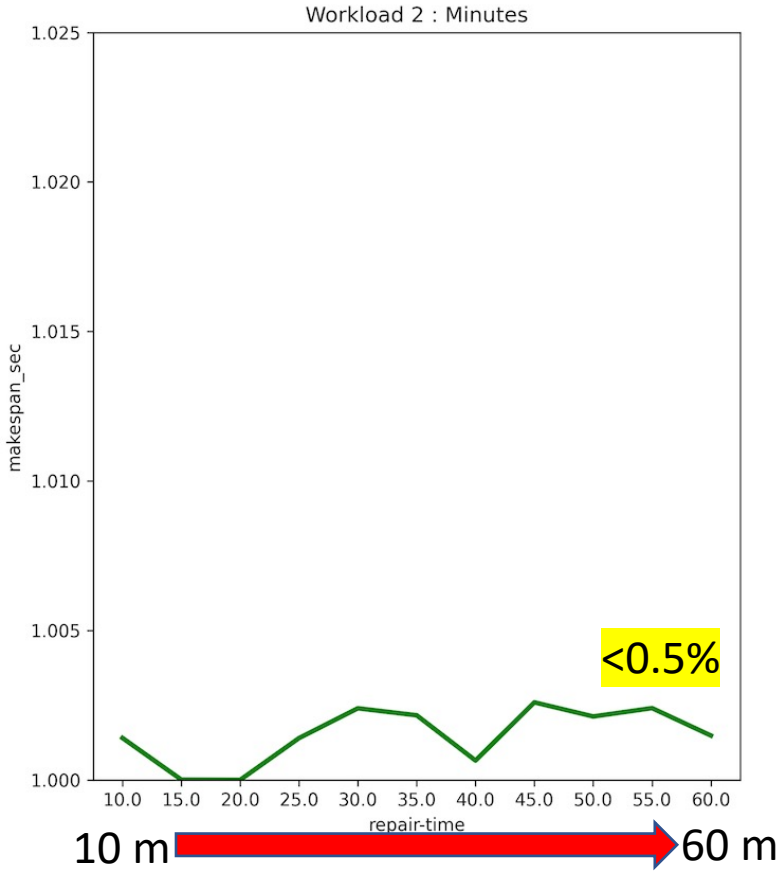
O(hours)



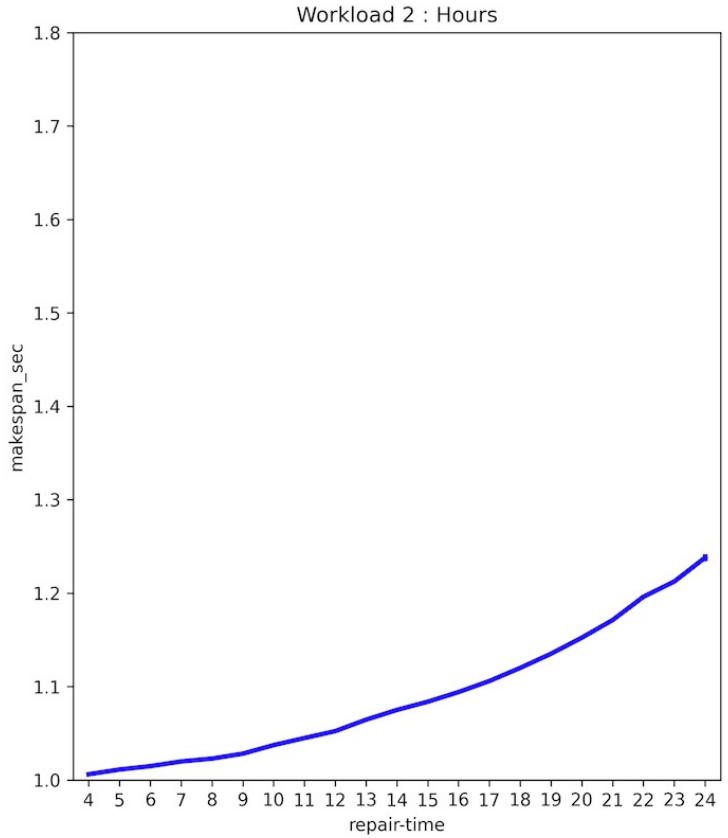
O(days)

Results (WL2 – Grizzly)

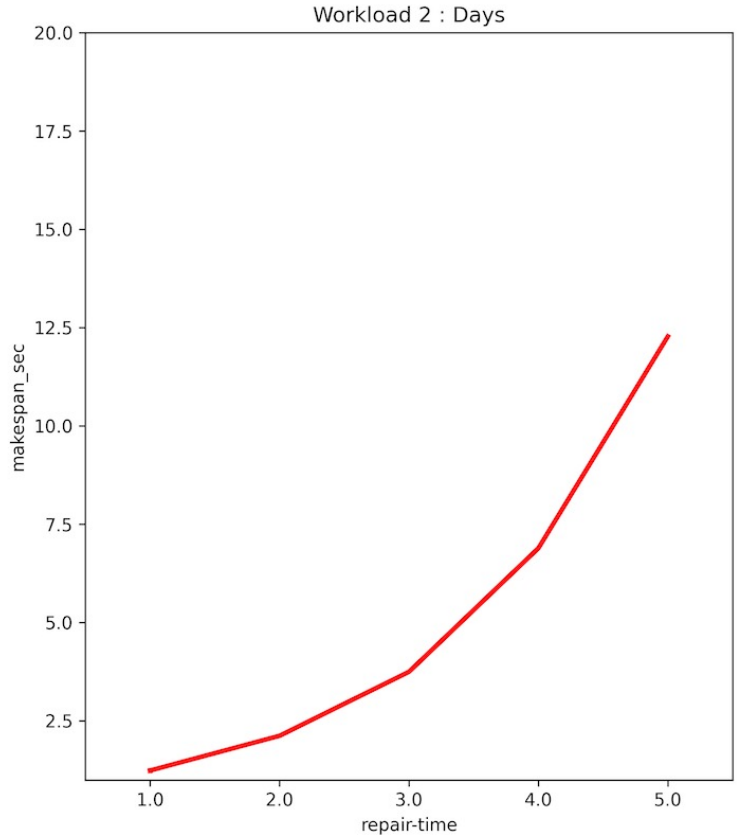
Jobs are based on 11 months of data from a capacity cluster at with 1490 nodes. This workload contained very few large jobs, with roughly 48% of the jobs requiring one node, and less than one hour of wall time, with the remaining jobs distributed in duration and width.



O(minutes)



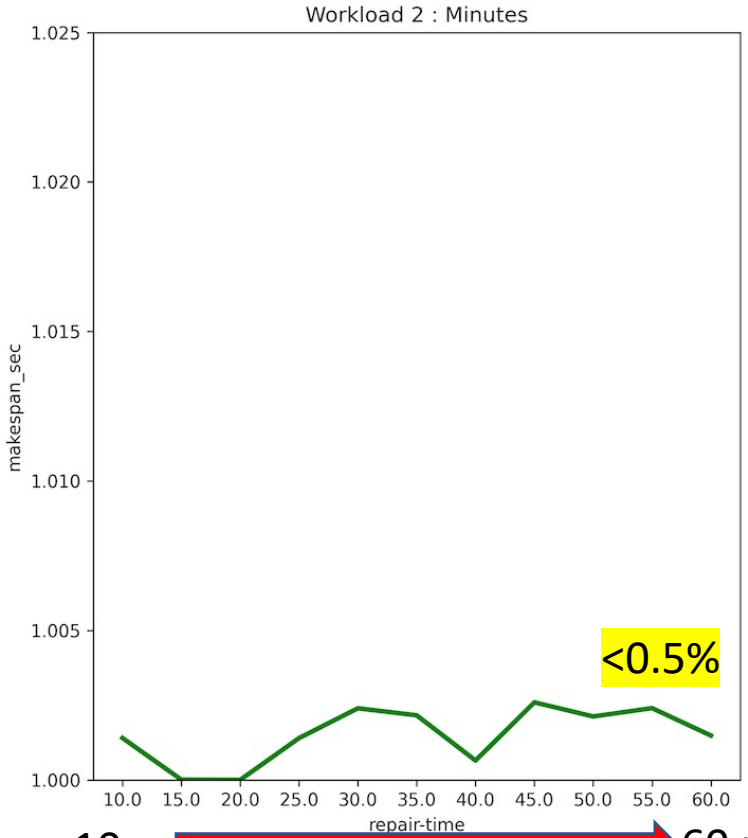
O(hours)



O(days)

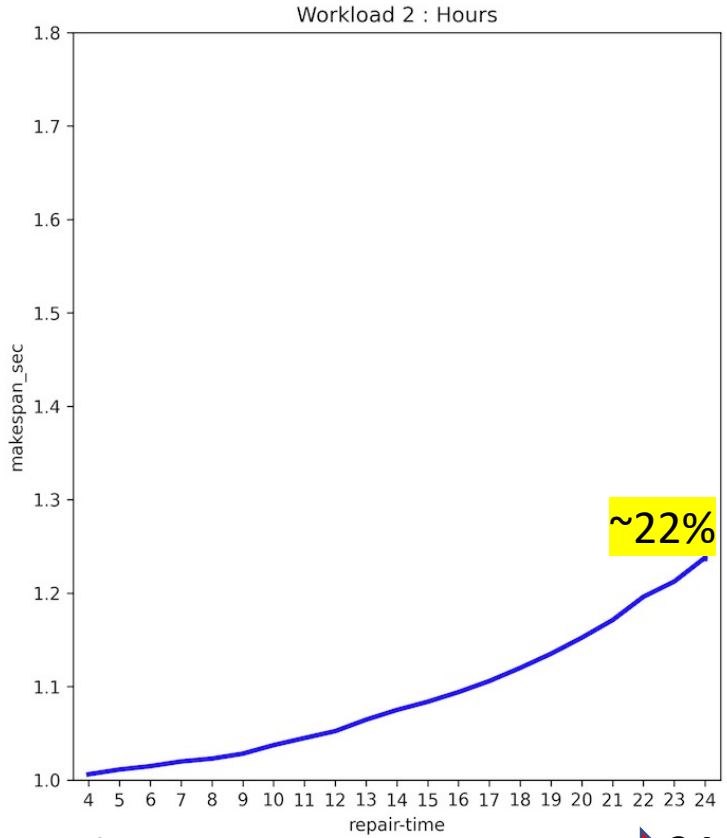
Results (WL2 – Grizzly)

Jobs are based on 11 months of data from a capacity cluster at with 1490 nodes. This workload contained very few large jobs, with roughly 48% of the jobs requiring one node, and less than one hour of wall time, with the remaining jobs distributed in duration and width.



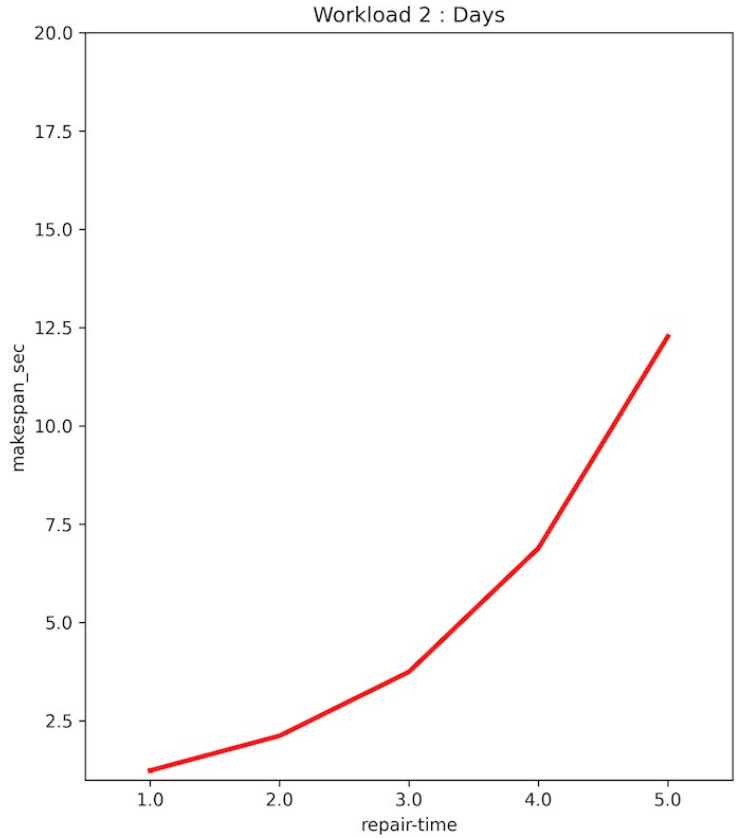
10 m → 60 m

O(minutes)



4 hr → 24 hr

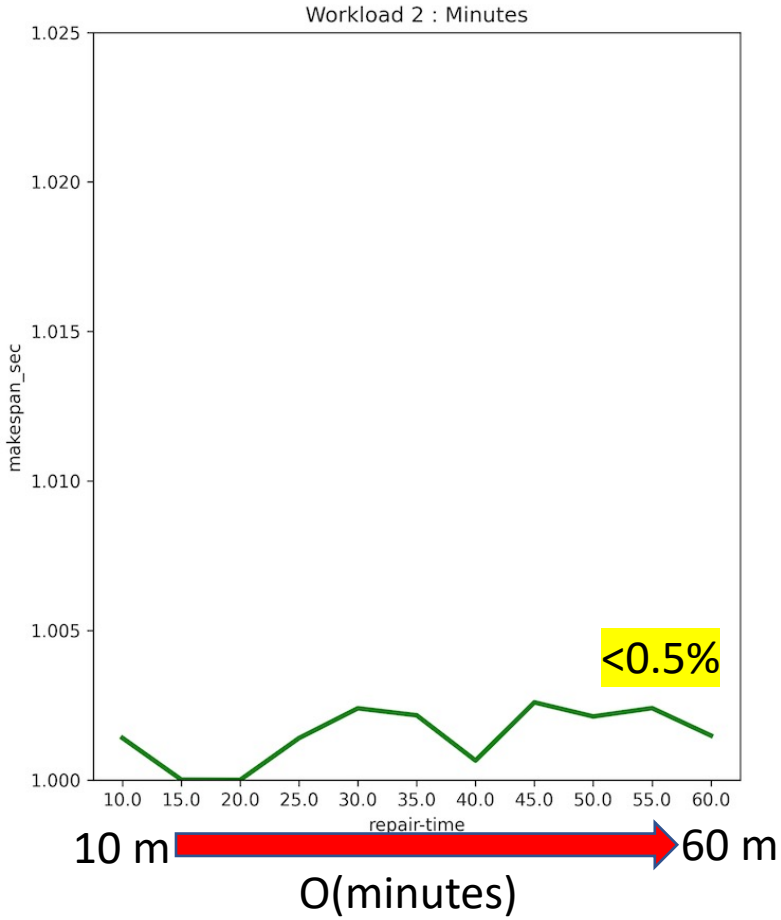
O(hours)

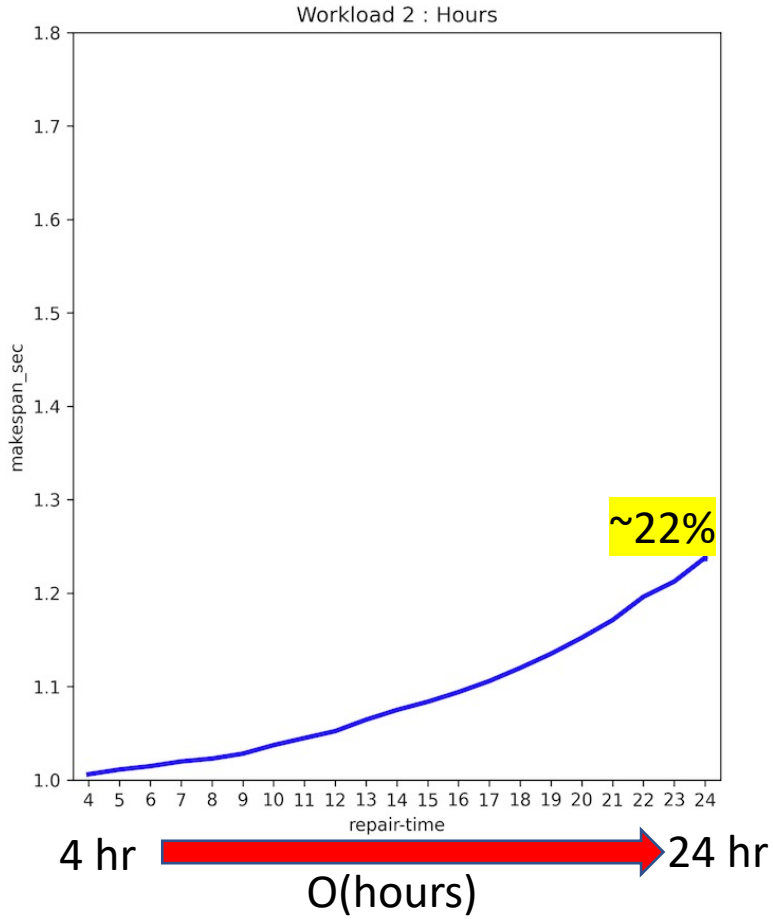


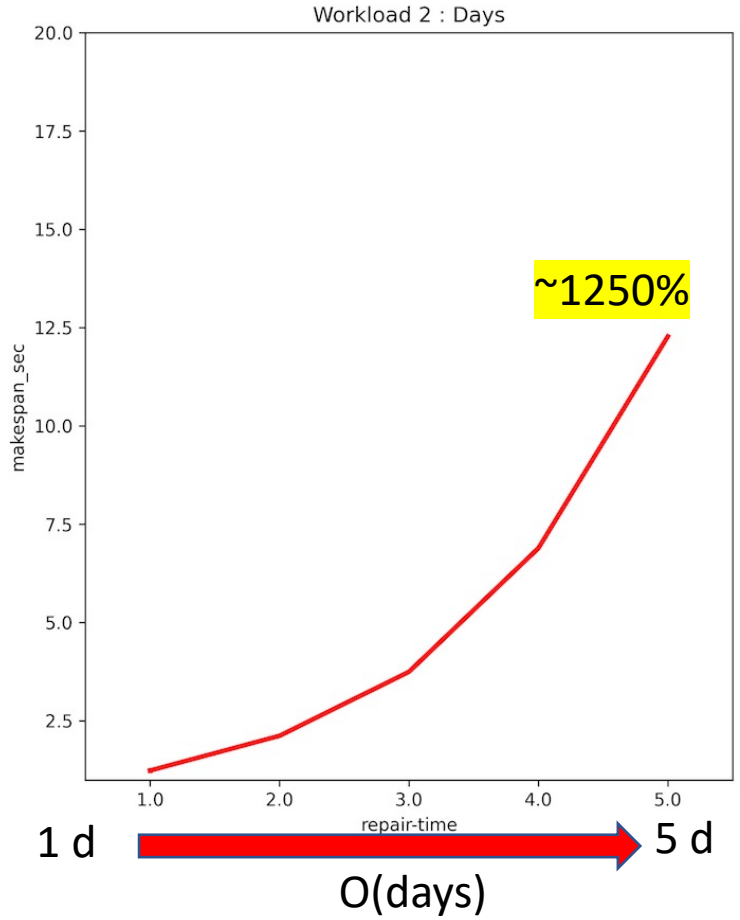
O(days)

Results (WL2 – Grizzly)

Jobs are based on 11 months of data from a capacity cluster at with 1490 nodes. This workload contained very few large jobs, with roughly 48% of the jobs requiring one node, and less than one hour of wall time, with the remaining jobs distributed in duration and width.

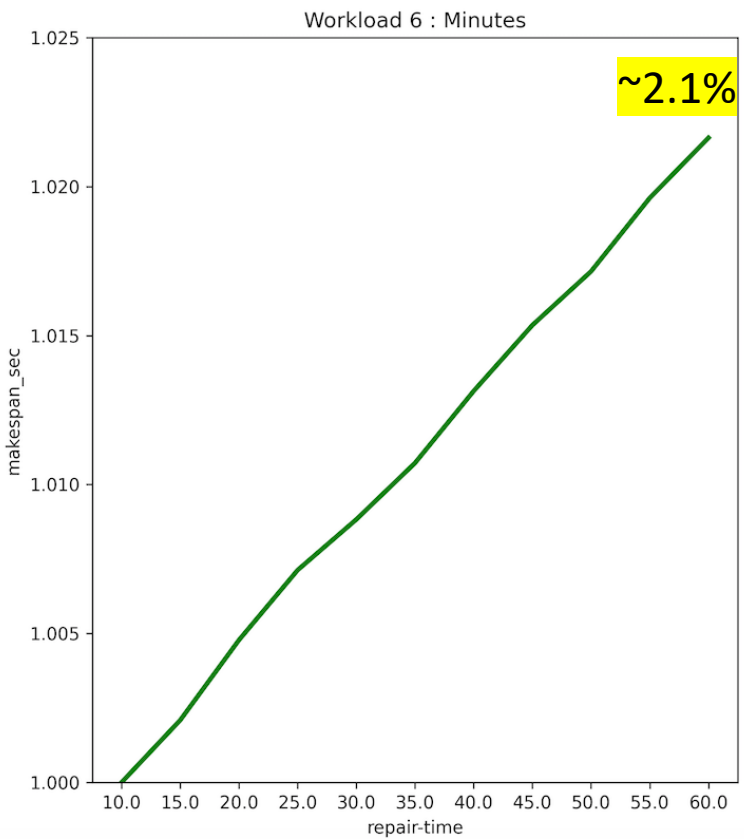




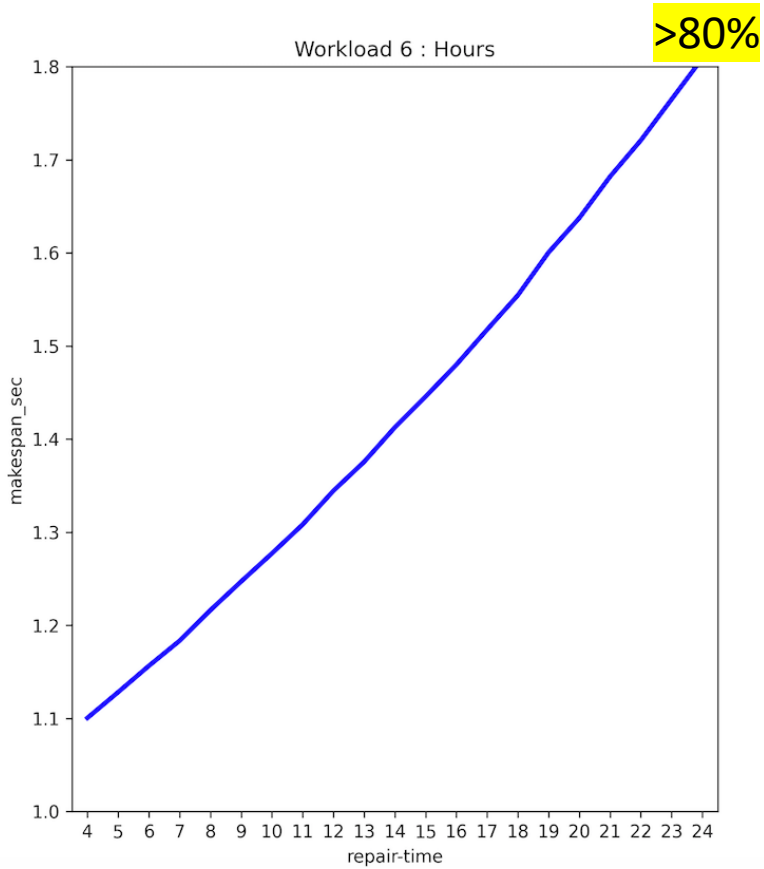


Results (WL6)

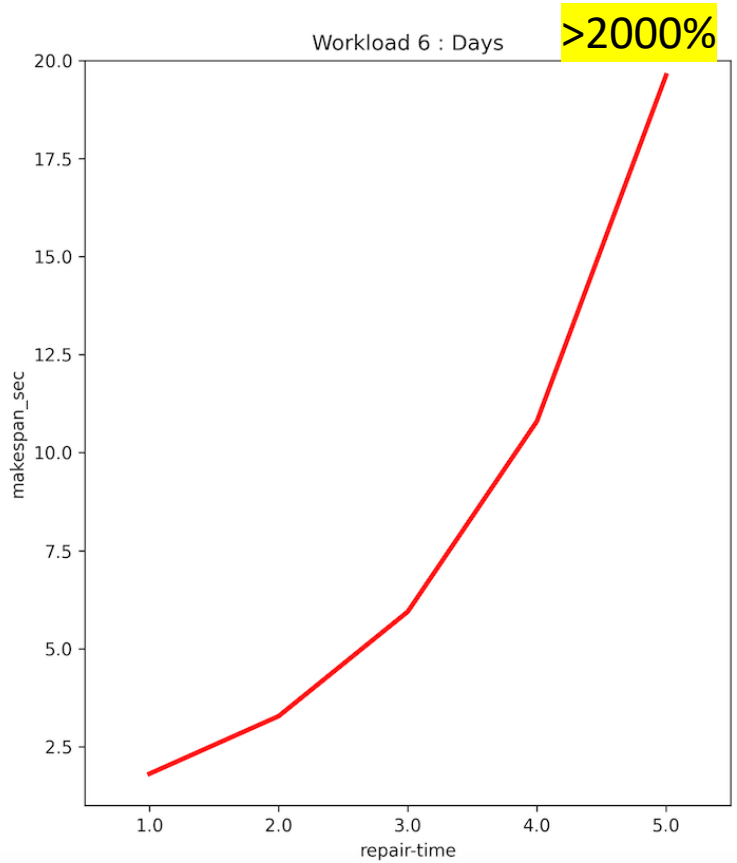
Jobs are all 1490 nodes wide with durations of 24 hours. This represents a "worst-case" scenario from a reliability point of view, as any node failure will result in a job failure that spans the entire cluster.



O(minutes)



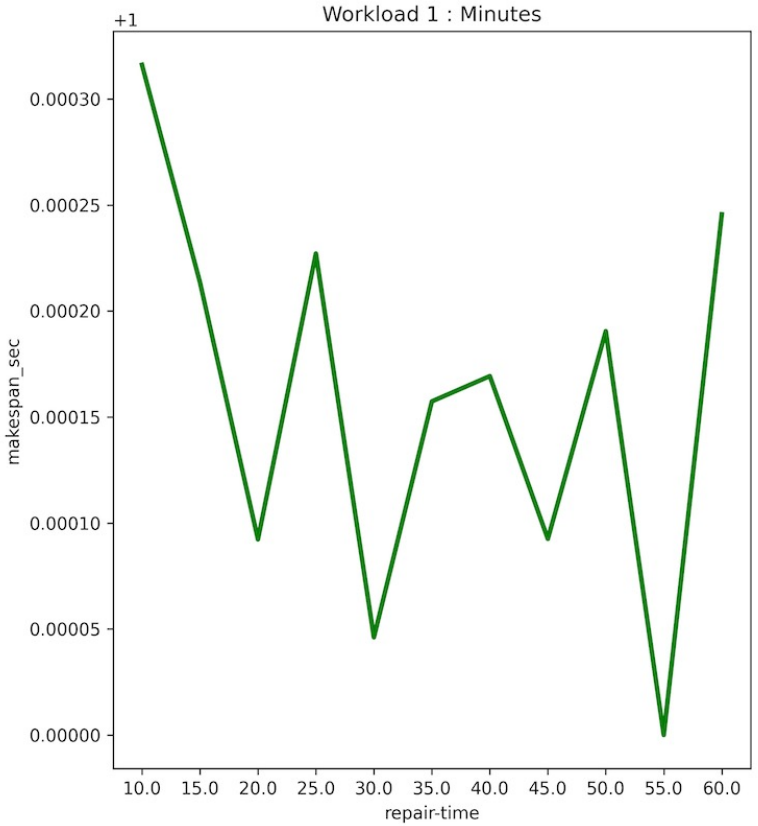
O(hours)



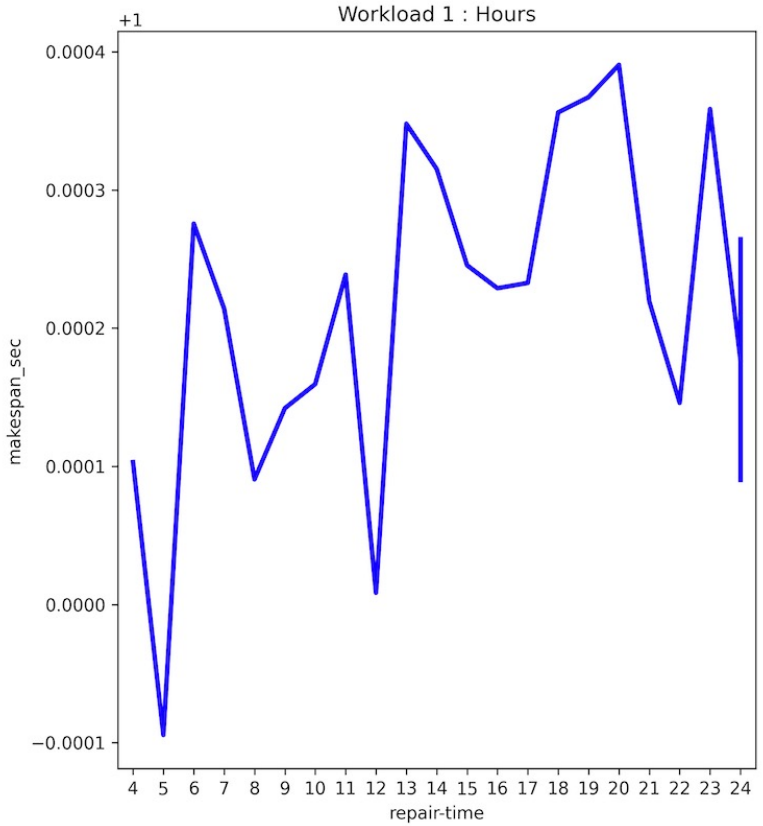
O(days)

Results (WL1)

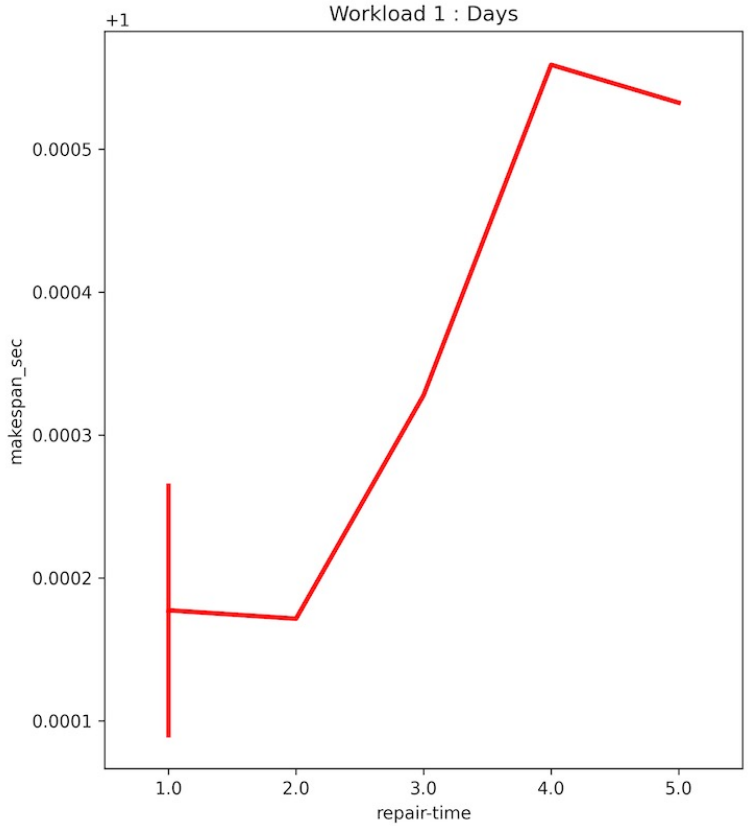
All jobs are one node wide and 24 hours long. This represents the analog of WL6, where all jobs span the entire cluster.



O(minutes)



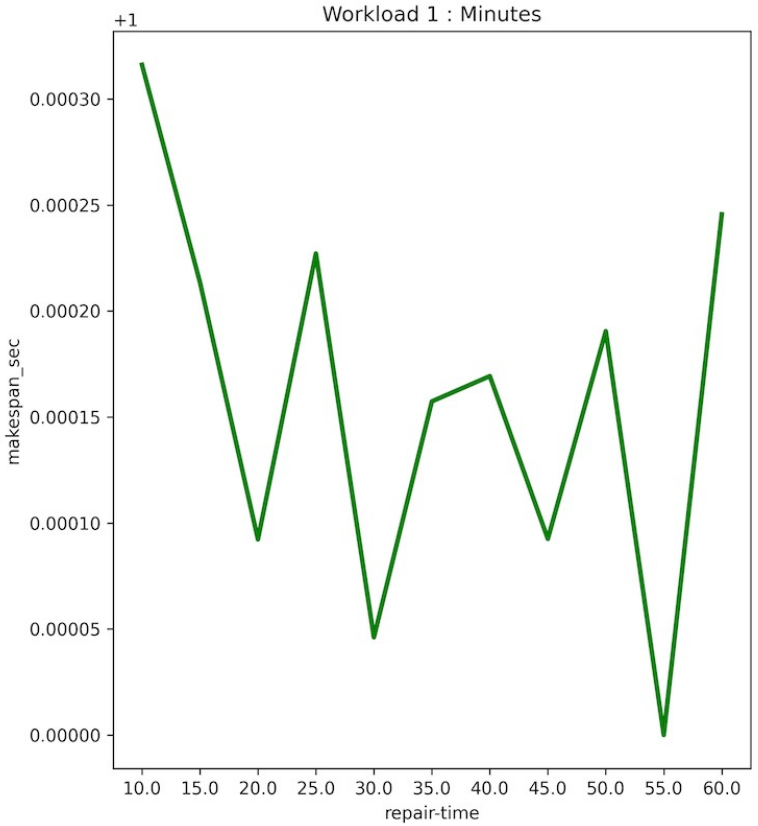
O(hours)



O(days)

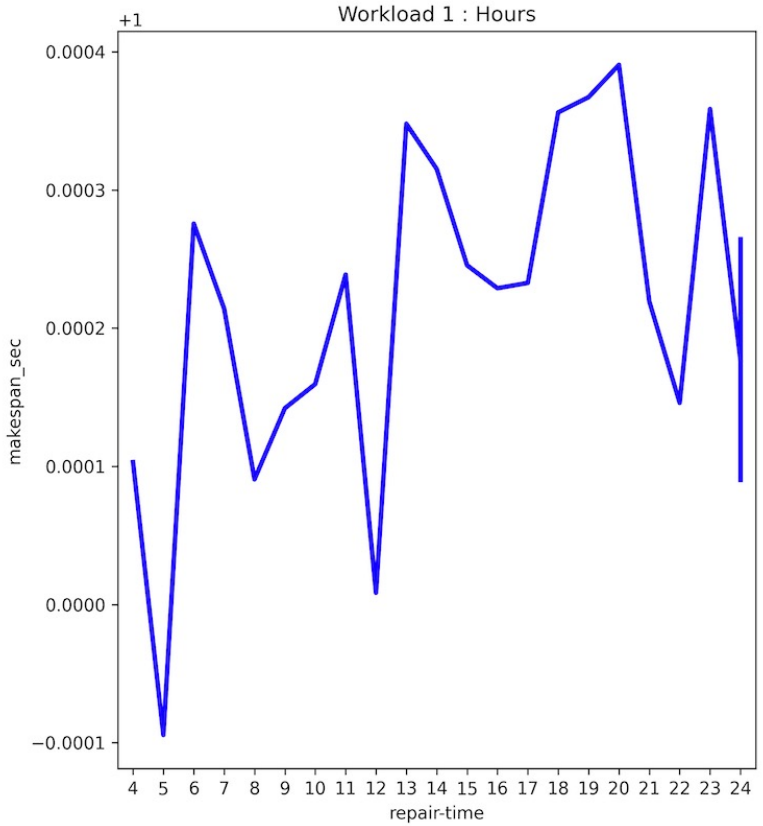
Results (WL1)

All jobs are one node wide and 24 hours long. This represents the analog of WL6, where all jobs span the entire cluster.



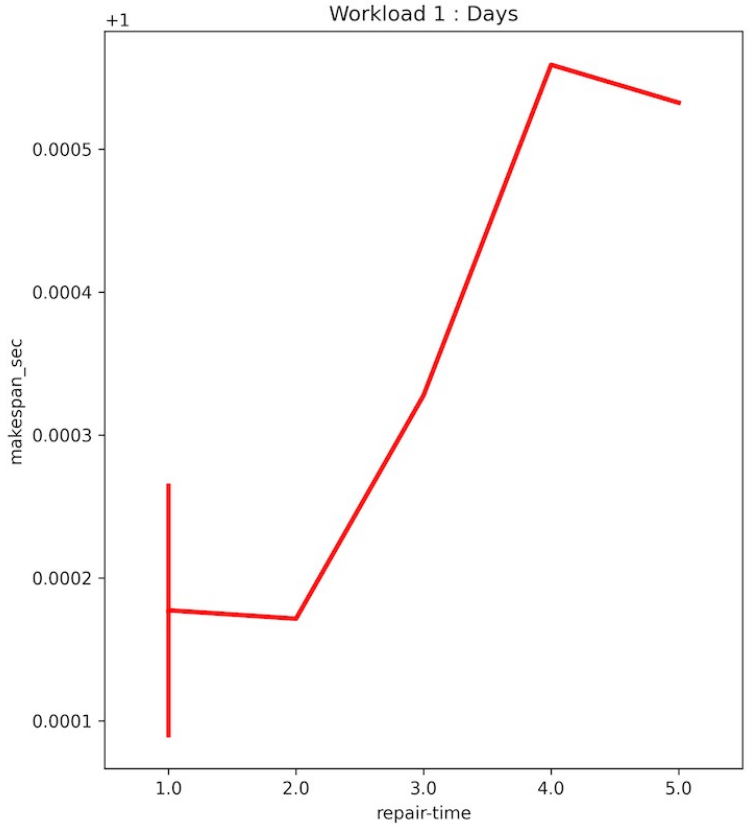
O(minutes)

Negligible impact



O(hours)

Negligible impact



O(days)

Negligible impact

Thrust Areas

Thrust Areas

Analyze workloads to (automatically / real-time?) **classify them as "production" or "development"**

What at the salient metrics?

Thrust Areas

Analyze workloads to (automatically / real-time?) **classify them as "production" or "development"**

What at the salient metrics?

Incorporate reservations into BatSim.

Backfilling ahead of reservations, what are impacts to performance?

Thrust Areas

Analyze workloads to (automatically / real-time?) **classify them as "production" or "development"**

What at the salient metrics?

Incorporate reservations into BatSim.

Backfilling ahead of reservations, what are impacts to performance?

Incorporate "partitions" into BatSim

What are the optimal relative sizes of the 'standard' and 'debug' partitions?

Thrust Areas

Analyze workloads to (automatically / real-time?) **classify them as "production" or "development"**

What at the salient metrics?

Incorporate reservations into BatSim.

Backfilling ahead of reservations, what are impacts to performance?

Incorporate "partitions" into BatSim

What are the optimal relative sizes of the 'standard' and 'debug' partitions?

Incorporate 'node sharing' into BatSim

How much share-packing is required to accommodate fewer nodes of greater resources?

Thrust Areas

Analyze workloads to (automatically / real-time?) **classify them as "production" or "development"**

What at the salient metrics?

Incorporate reservations into BatSim.

Backfilling ahead of reservations, what are impacts to performance?

Incorporate "partitions" into BatSim

What are the optimal relative sizes of the 'standard' and 'debug' partitions?

Incorporate 'node sharing' into BatSim

How much share-packing is required to accommodate fewer nodes of greater resources?

Production-driven goals

Thrust Areas

Analyze workloads to (automatically / real-time?) **classify them as "production" or "development"**

What at the salient metrics?

stats / basic ML

Incorporate reservations into BatSim.

Backfilling ahead of reservations, what are impacts to performance?

Incorporate "partitions" into BatSim

What are the optimal relative sizes of the 'standard' and 'debug' partitions?

Incorporate 'node sharing' into BatSim

How much share-packing is required to accommodate fewer nodes of greater resources?

Thrust Areas

Analyze workloads to (automatically / real-time?) **classify them as "production" or "development"**

What at the salient metrics?

Incorporate reservations into BatSim.

Backfilling ahead of reservations, what are impacts to performance?

Incorporate "partitions" into BatSim

What are the optimal relative sizes of the 'standard' and 'debug' partitions?

Incorporate 'node sharing' into BatSim

How much share-packing is required to accommodate fewer nodes of greater resources?

stats / basic ML

modifications to BatSim

Next Step

Incorporate 'node sharing' into BatSim

How much share-packing is required to accommodate fewer nodes of greater resources?

Next Step

View node as collection of cores / PEs

Incorporate 'node sharing' into BatSim

How much share-packing is required to accommodate fewer nodes of greater resources?

Next Step

View node as collection of cores / PEs

Prior work: job were just “rectangles”, i.e. didn’t use CPU model

Incorporate ‘node sharing’ into BatSim

How much share-packing is required to accommodate fewer nodes of greater resources?

Next Step

View node as collection of cores / PEs

Prior work: job were just “rectangles”, i.e. didn’t use CPU model

Model to address impact to baseline runtimes with concurrent jobs on same node

Incorporate ‘node sharing’ into BatSim

How much share-packing is required to accommodate fewer nodes of greater resources?

(*simplifying*) Assumptions

Assume all 1-node jobs in the Grizzly workload represent ‘serial’ jobs

that these jobs only really need 1 ”core”

(*simplifying*) Assumptions

Assume all 1-node jobs in the Grizzly workload represent ‘serial’ jobs
that these jobs only really need 1 ”core”

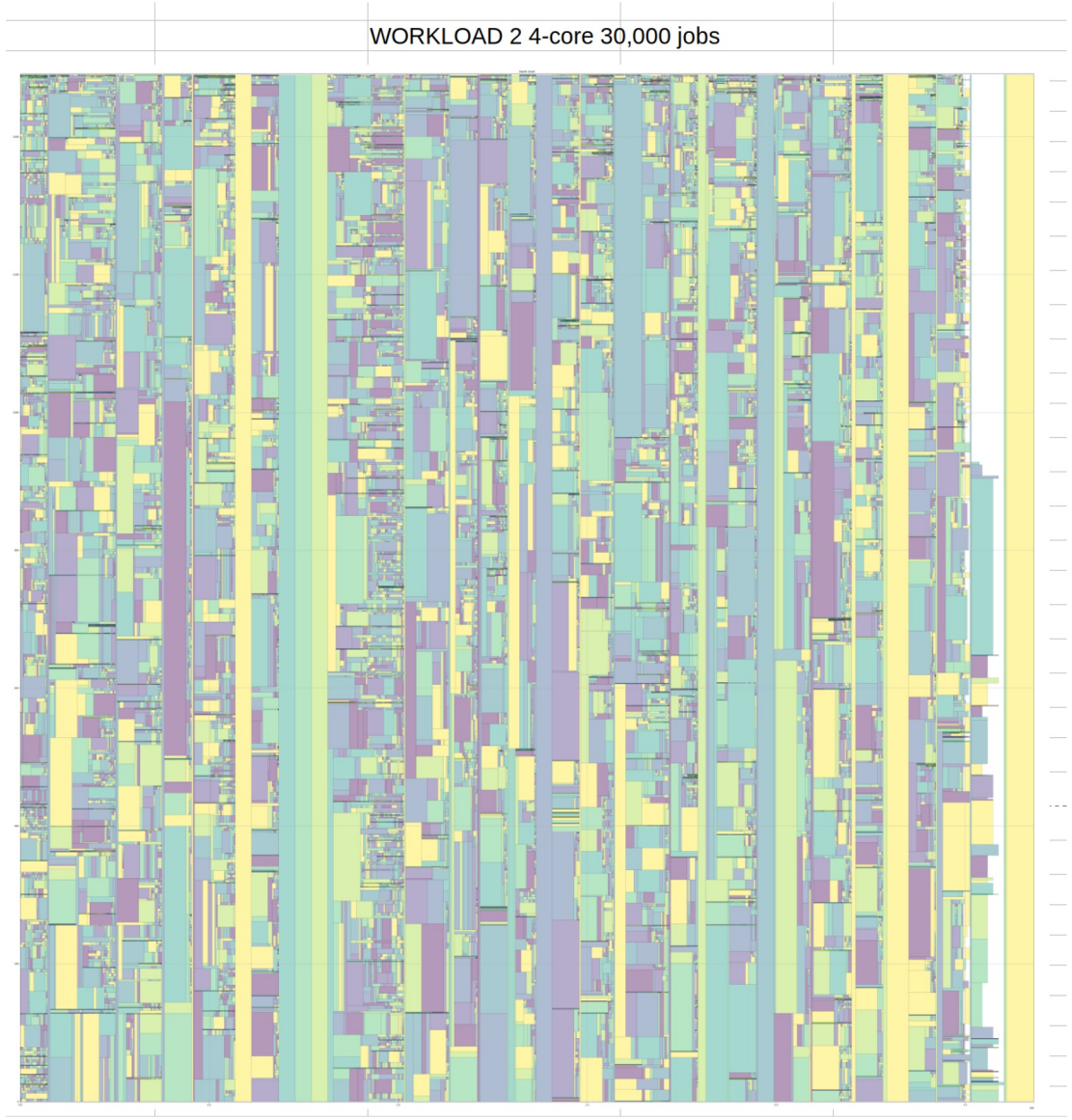
All others are true, parallel (multi-node) jobs
fully use a node (or nodes)

(*simplifying*) Assumptions

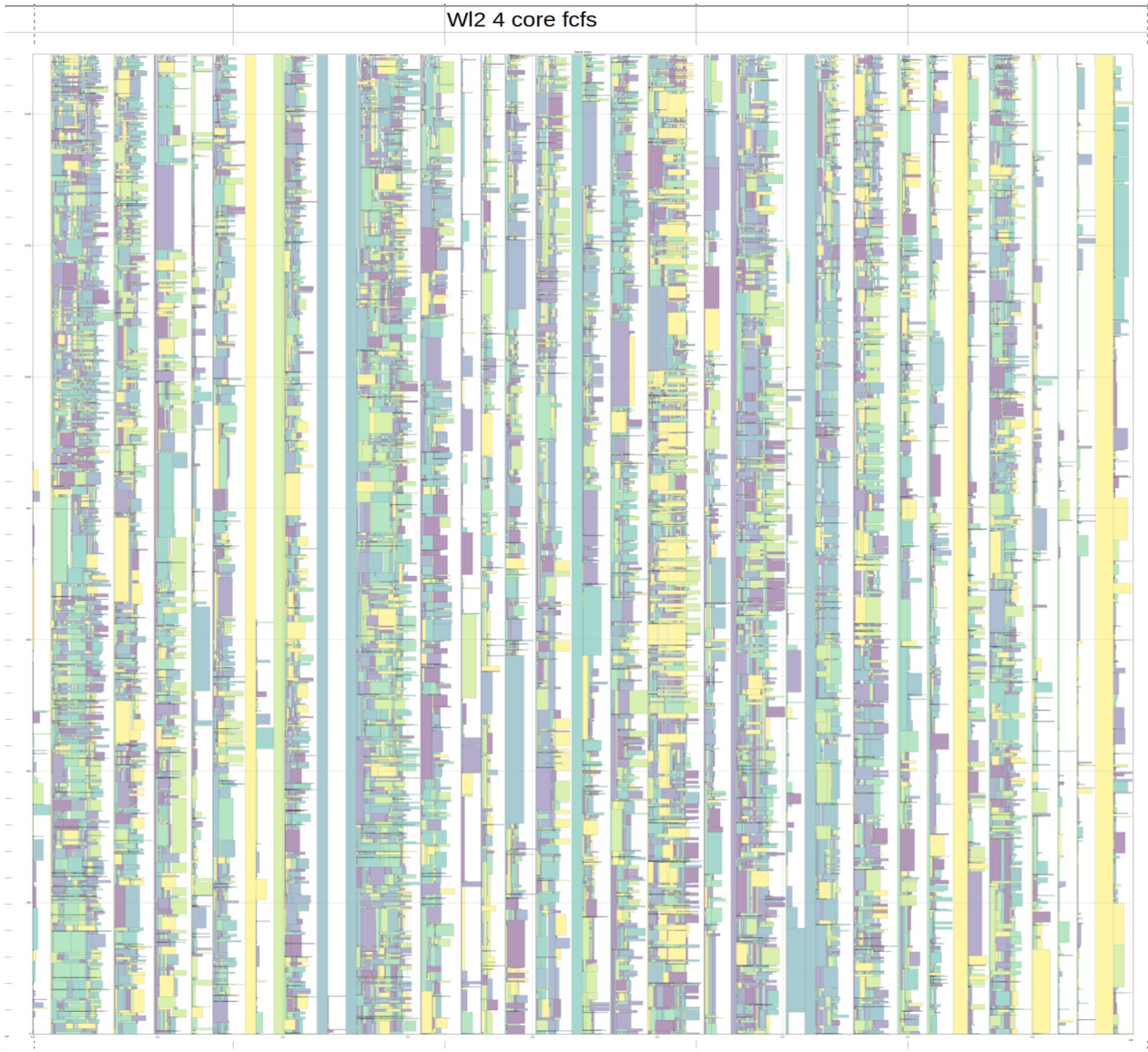
Assume all 1-node jobs in the Grizzly workload represent ‘serial’ jobs
that these jobs only really need 1 ”core”

All others are true, parallel (multi-node) jobs
fully use a node (or nodes)

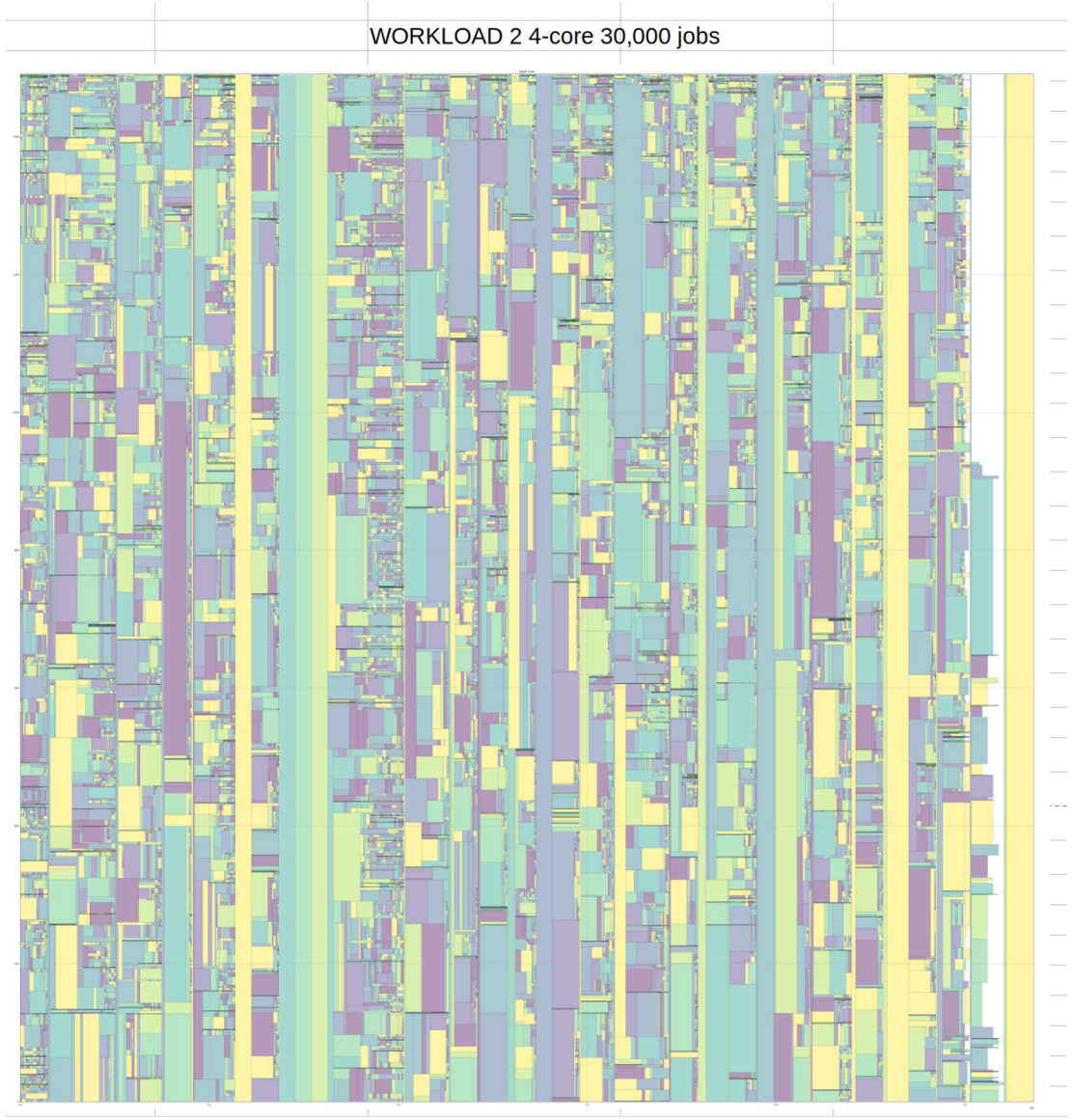
Assume that 1-core jobs can ‘fill’ up to X% of cores (configurable) of a node, without “interaction’ – not addressing ”interaction” at all at this point

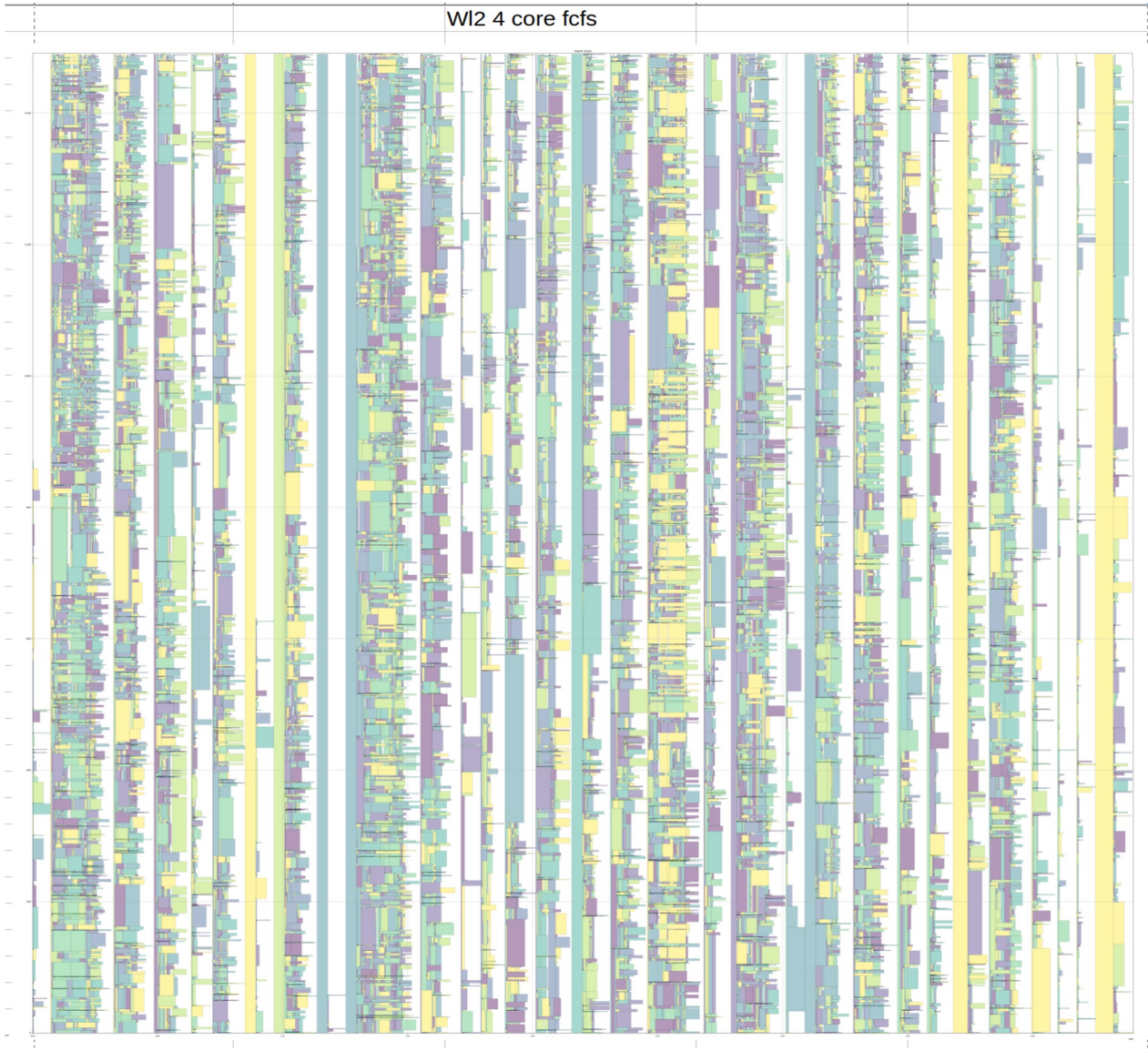


EASY BF (later conservative)



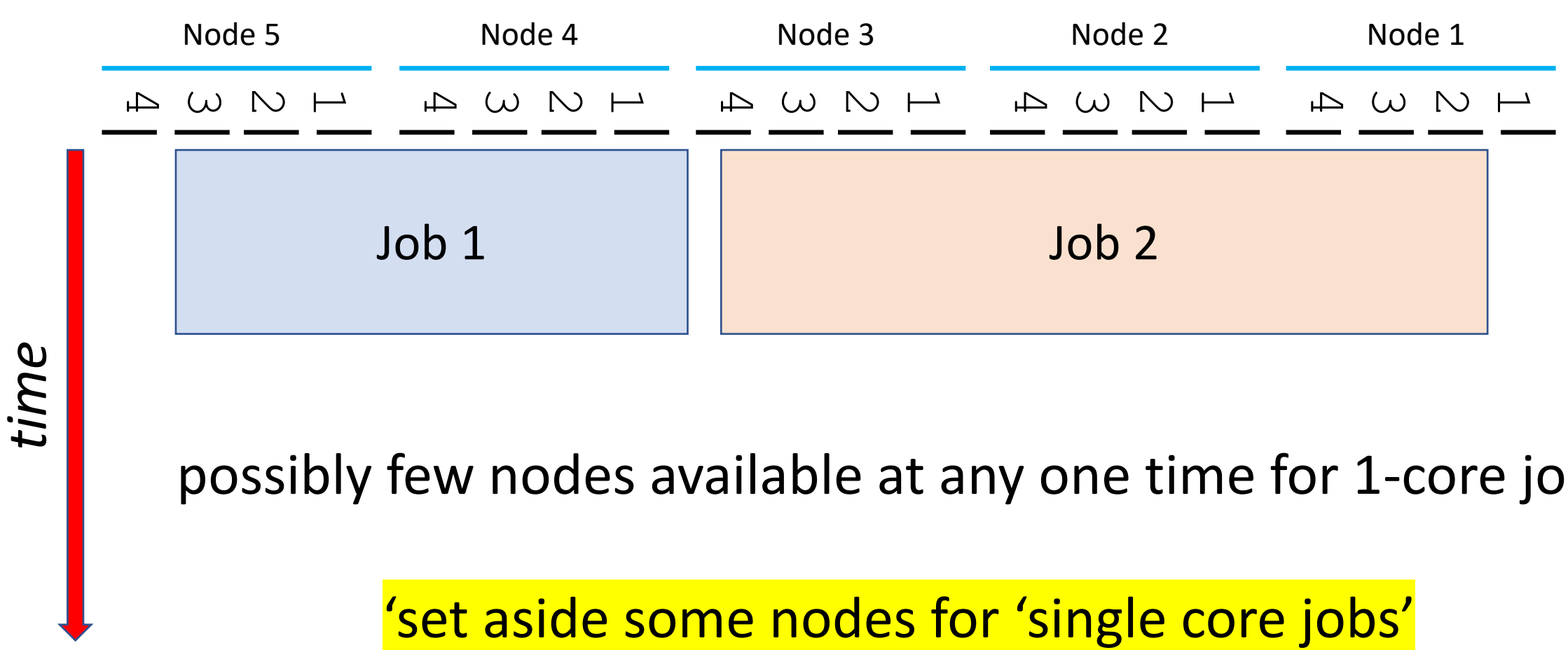
FCFS





EASY BF (later conservative)

FCFS



possibly few nodes available at any one time for 1-core jobs

‘set aside some nodes for ‘single core jobs’

partitions

congruent with debug scenarios

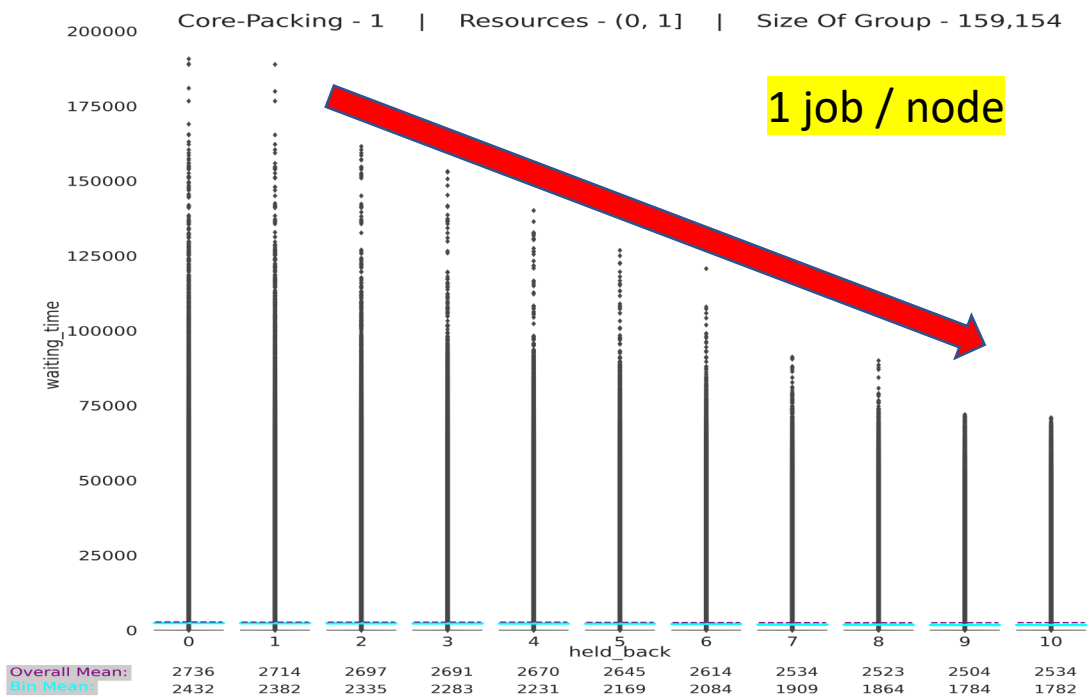
The next slides:

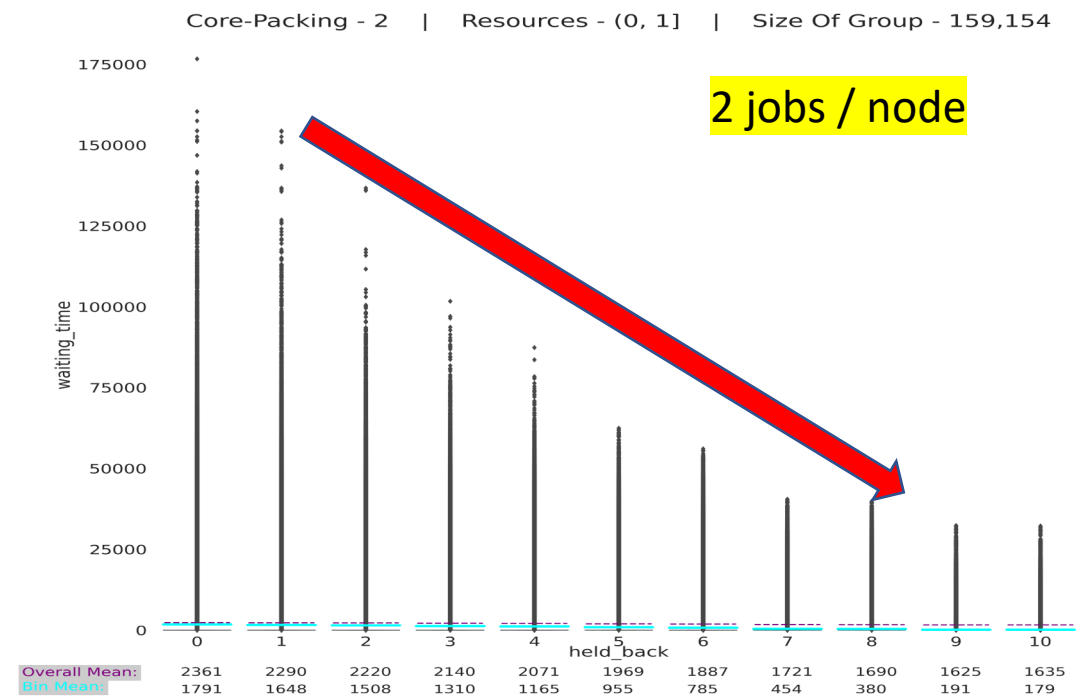
The next slides have 4 graphs differing in the amount of cores.

They are all for 300:exp interarrival time

They are all for the bin (0,1] ie 1 resource jobs

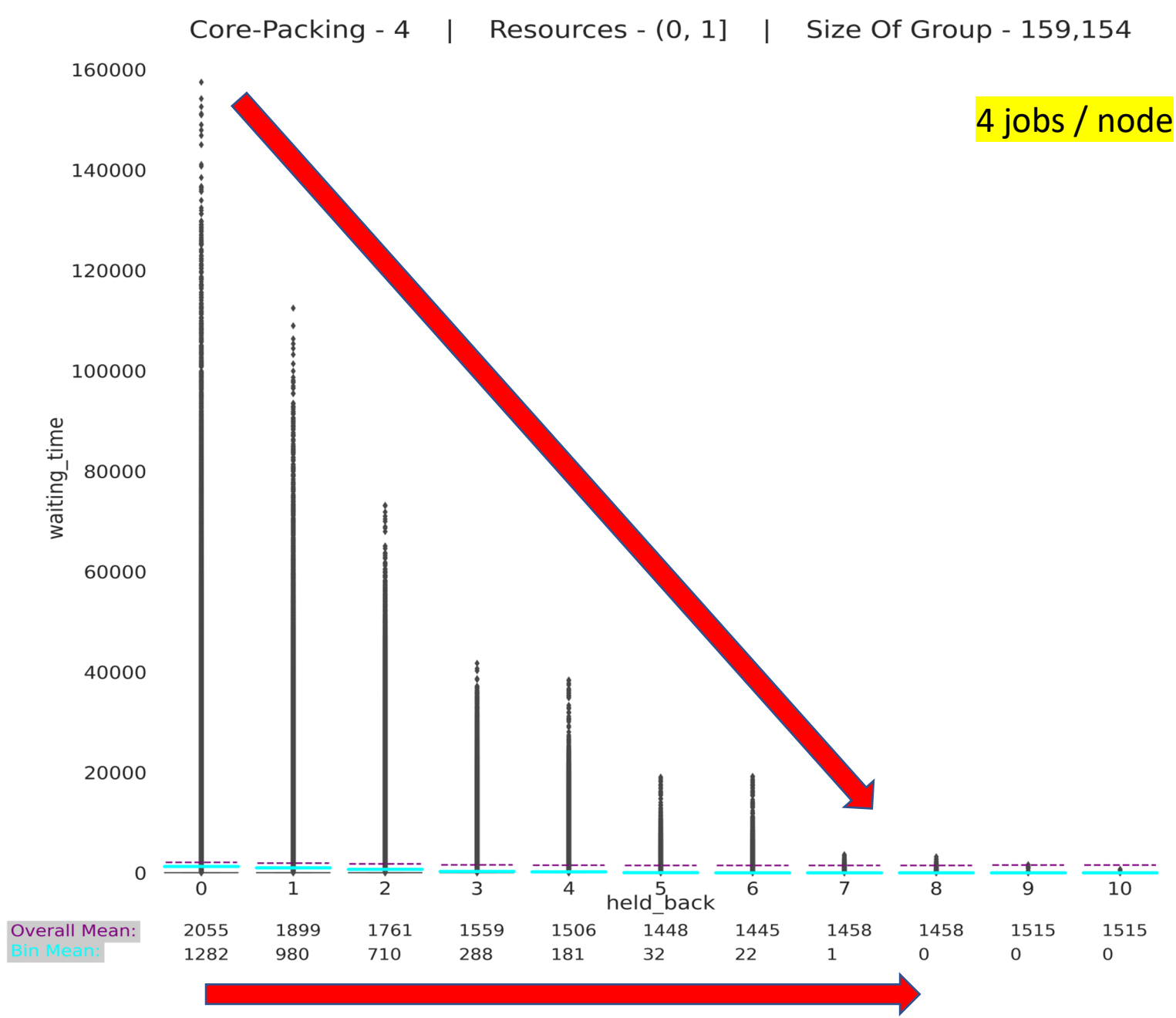
Assumed that Grizzly was 1500 nodes, with *up to* 10 “held back” exclusively for single core (packable) jobs.


Core-Packing - 1 | Resources - (0, 1] | Size Of Group - 159,154
1 job / node
waiting_time
held_back
Overall Mean: 2736 2714 2697 2691 2670 2645 2614 2534 2523 2504 2534
Bin Mean: 2432 2382 2335 2283 2231 2169 2084 1909 1864 1784 1782

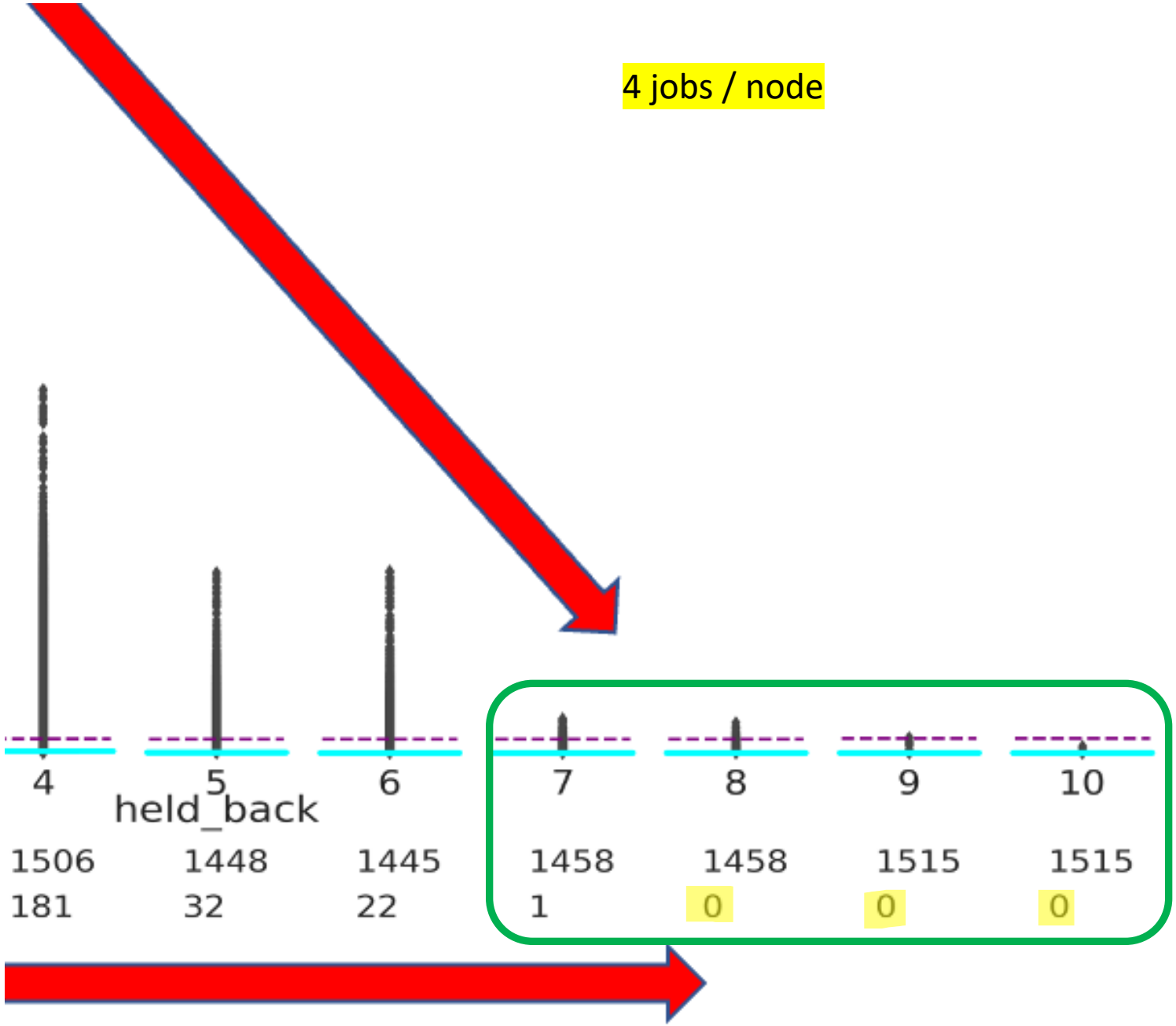

Core-Packing - 2 | Resources - (0, 1] | Size Of Group - 159,154
2 jobs / node
waiting_time
held_back
Overall Mean: 2361 2290 2220 2140 2071 1969 1887 1721 1690 1625 1635
Bin Mean: 1791 1648 1508 1310 1165 955 785 454 380 191 179


Core-Packing - 3 | Resources - (0, 1] | Size Of Group - 159,154
3 jobs / node
waiting_time
held_back
Overall Mean: 2141 2038 1919 1734 1661 1583 1564 1475 1472 1516 1536
Bin Mean: 1419 1213 975 584 436 274 225 28 21 2 1


Core-Packing - 4 | Resources - (0, 1] | Size Of Group - 159,154
4 jobs / node
waiting_time
held_back
Overall Mean: 2053 1899 1761 1559 1506 1448 1445 1458 1458 1515 1515
Bin Mean: 1282 980 710 288 181 32 22 1 0 0 0

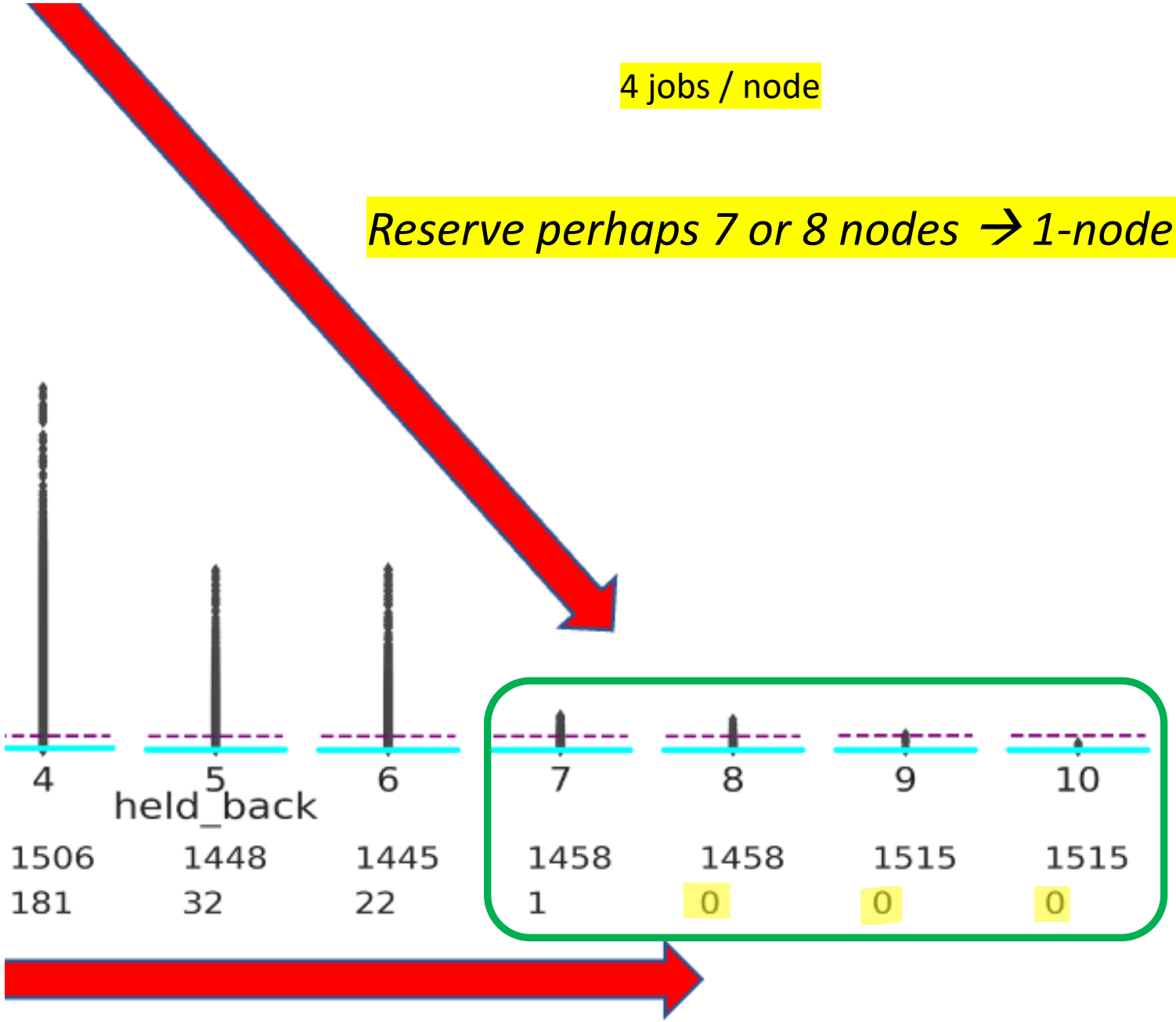

Core-Packing - 4 | Resources - (0, 1] | Size Of Group - 159,154
4 jobs / node
waiting_time
160000
140000
120000
100000
80000
60000
40000
20000
0
held_back
0 1 2 3 4 5 6 7 8 9 10
Overall Mean: 2055 1899 1761 1559 1506 1448 1445 1458 1458 1515 1515
Bin Mean: 1282 980 710 288 181 32 22 1 0 0 0

4 jobs / node


4
5
6
7
8
9
10
held_back
1506
1448
1445
1458
1458
1515
1515
181
32
22
1
0
0
0

4 jobs / node

Reserve perhaps 7 or 8 nodes → 1-node jobs never wait



Thrust Areas

Analyze workloads to (automatically / real-time?) **classify them as "production" or "development"**

What at the salient metrics?

Incorporate reservations into BatSim.

Backfilling ahead of reservations, what are impacts to performance?

Incorporate "partitions" into BatSim

What are the optimal relative sizes of the 'standard' and 'debug' partitions?

Incorporate 'node sharing' into BatSim

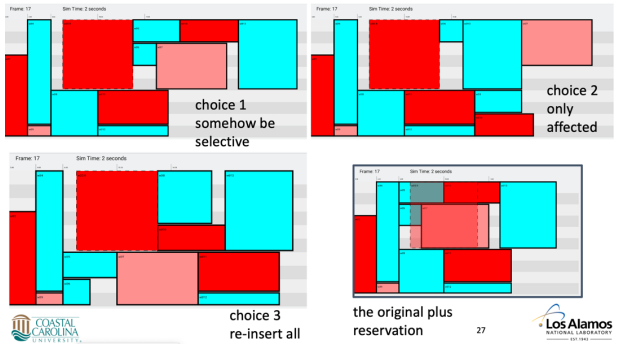
How much share-packing is required to accommodate fewer nodes of greater resources?

stats / basic ML

modifications to BatSim

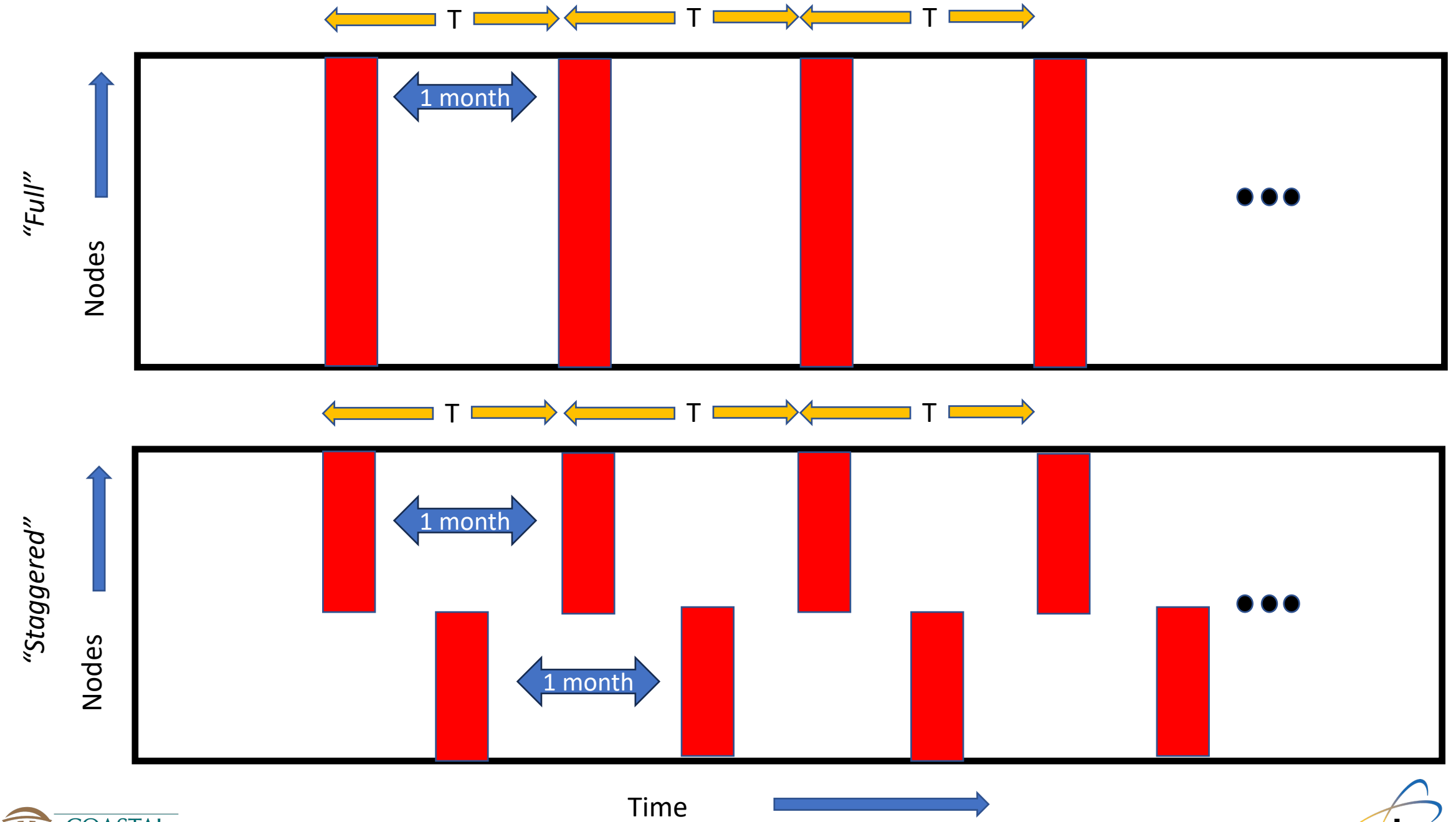
Incorporate Reservations into BatSim

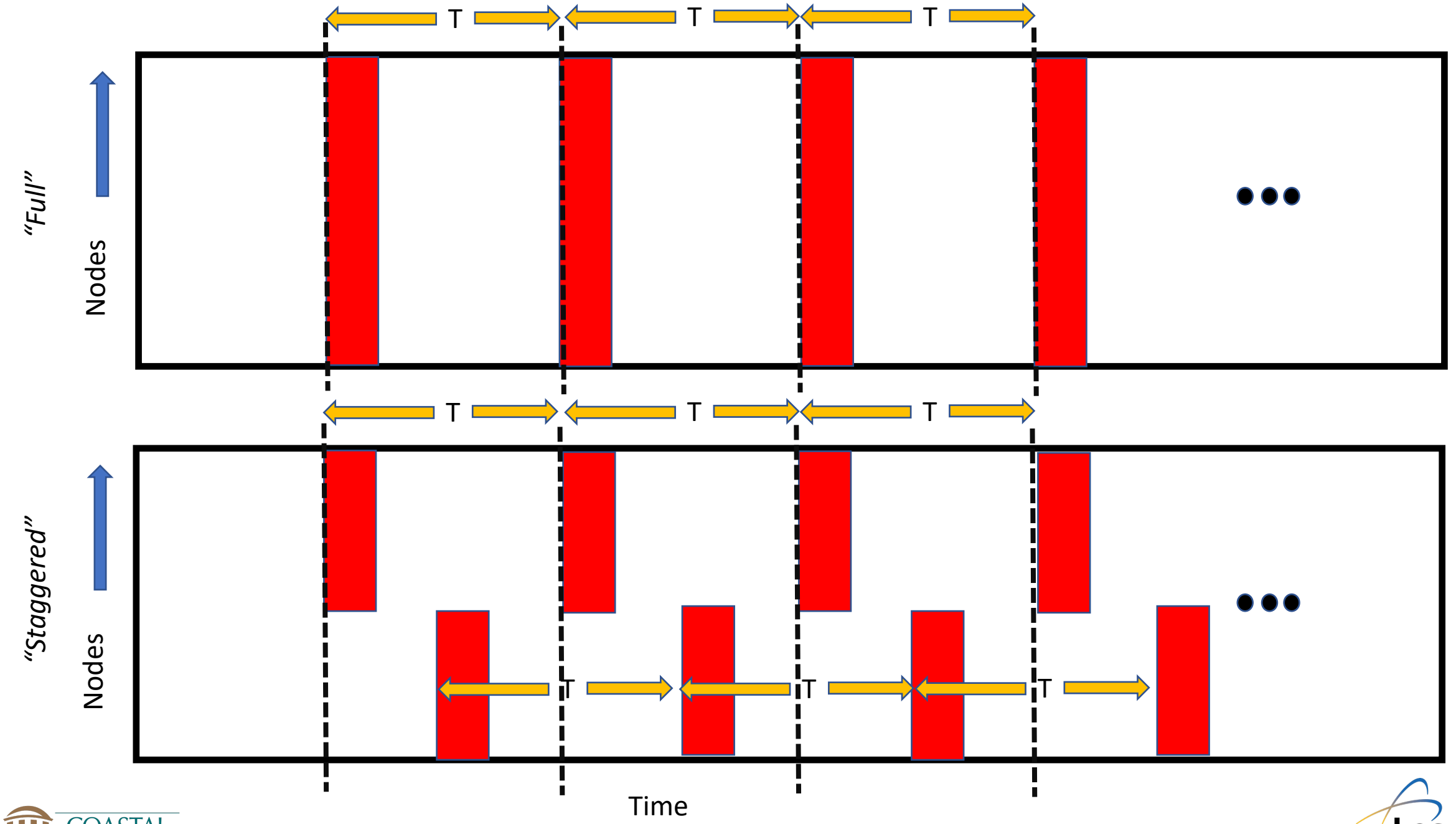
- Different scenarios
  - schedule reservations “in the future”
    - before jobs have been scheduled – jobs ‘flow’ around these reservations
    - conflict between scheduled jobs and future reservation
    - conflict between running jobs and future (current) reservation
  - schedule reservations
- Policies
  - when a conflict happens:
    - only reschedule impacted jobs
    - reschedule all jobs so that the implicit priorities are respected
  - impacted jobs – progress they make deducted from requested wall times?
- Maintain compatibility with existing improvements
  - checkpointing / faults / jobs restarting

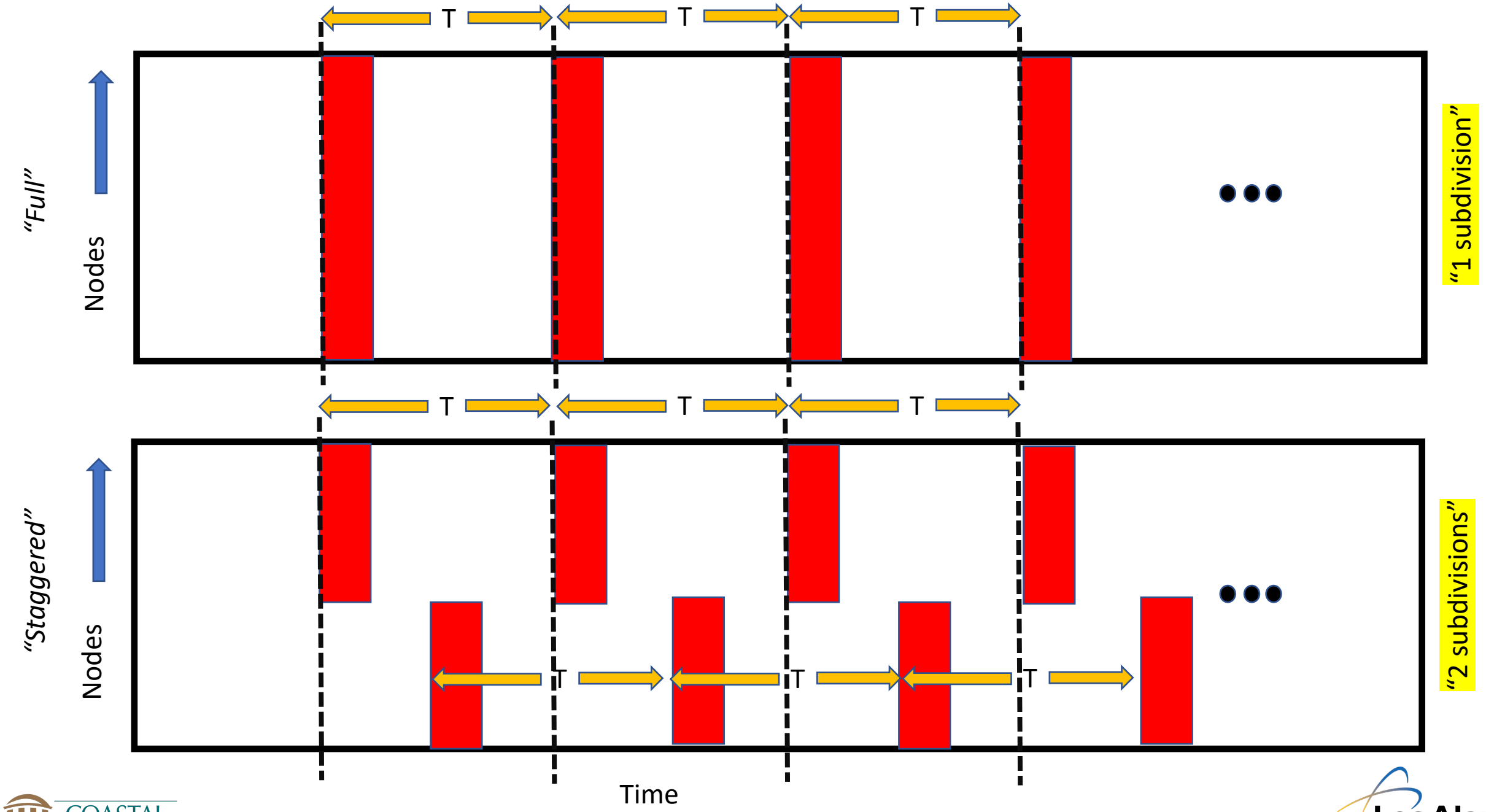


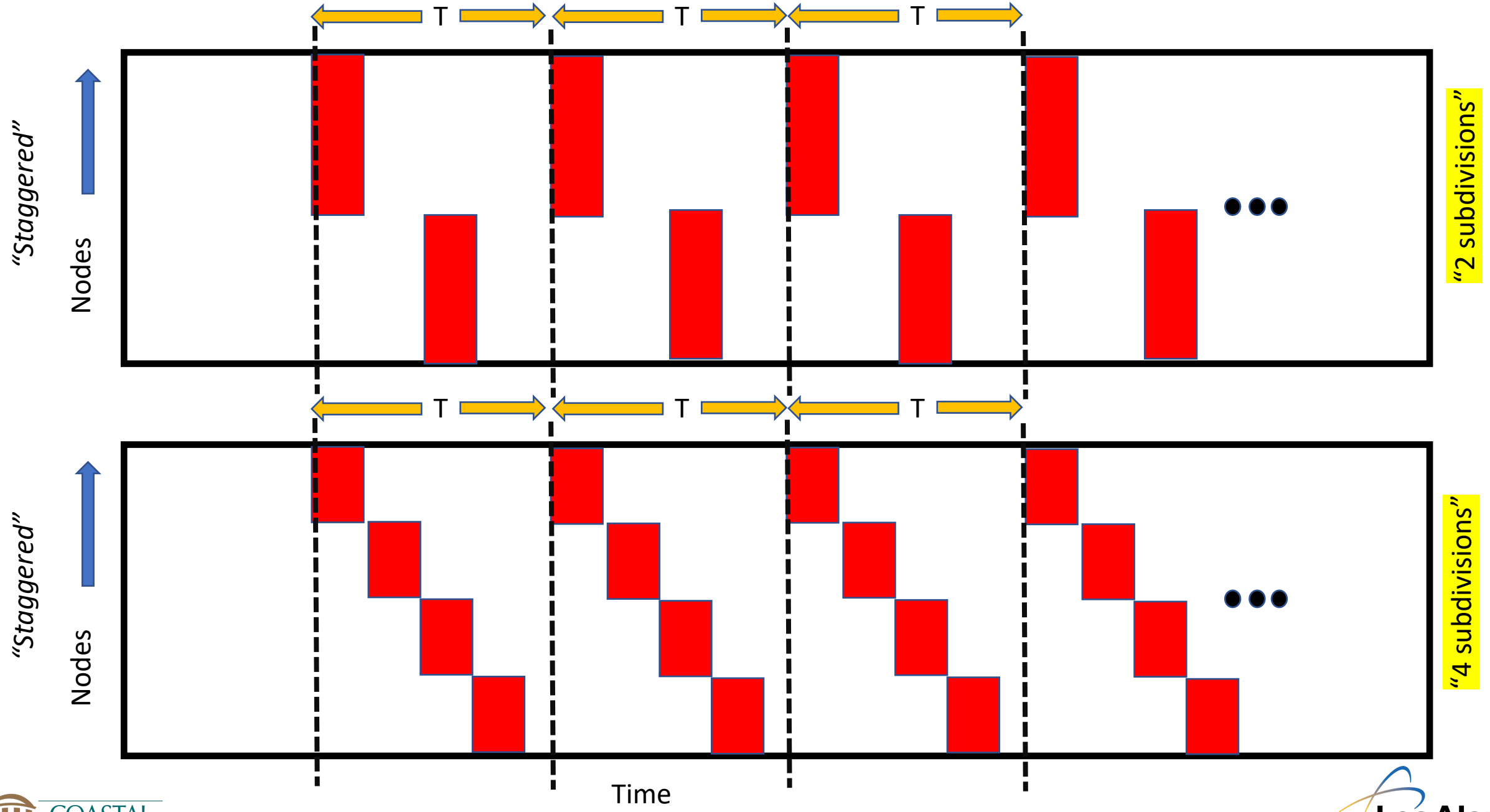
Some initial investigation

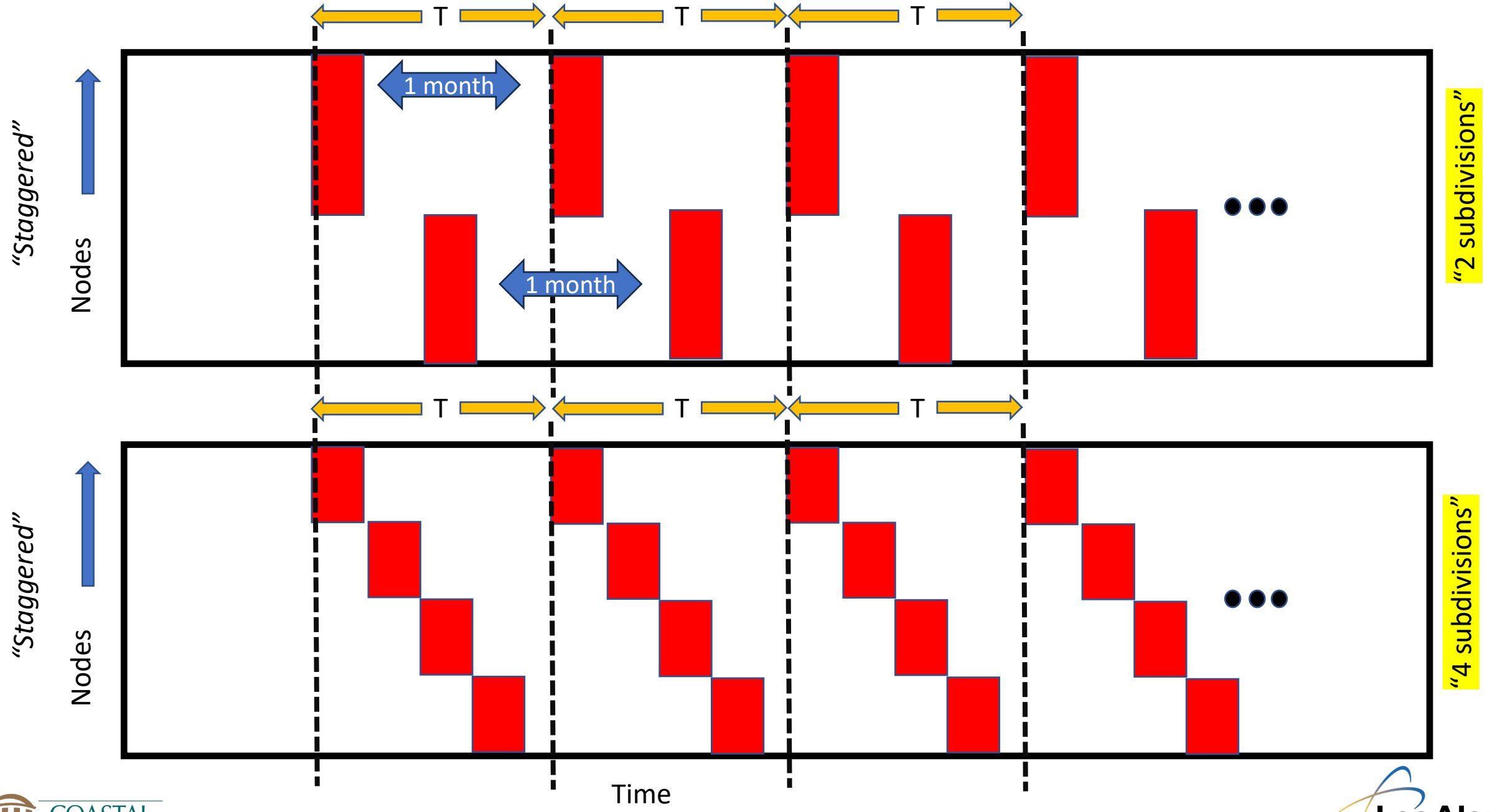
- Compare and contrast periodic reservations
  - Span the full cluster
  - Staggered reservations

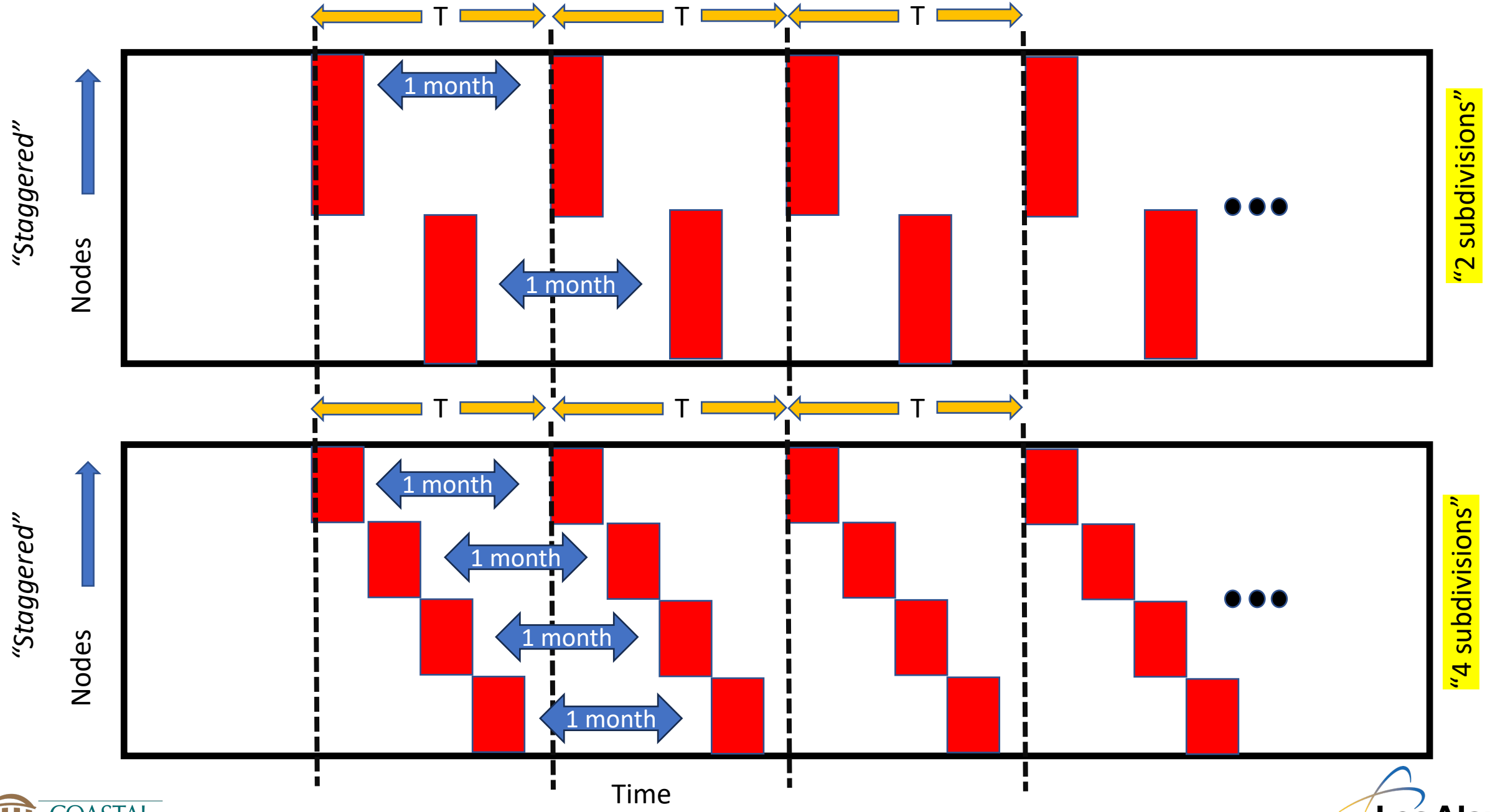

T
T
T
1 month
"Full"
Nodes
T
T
T
1 month
1 month
"Staggered"
Nodes
Time

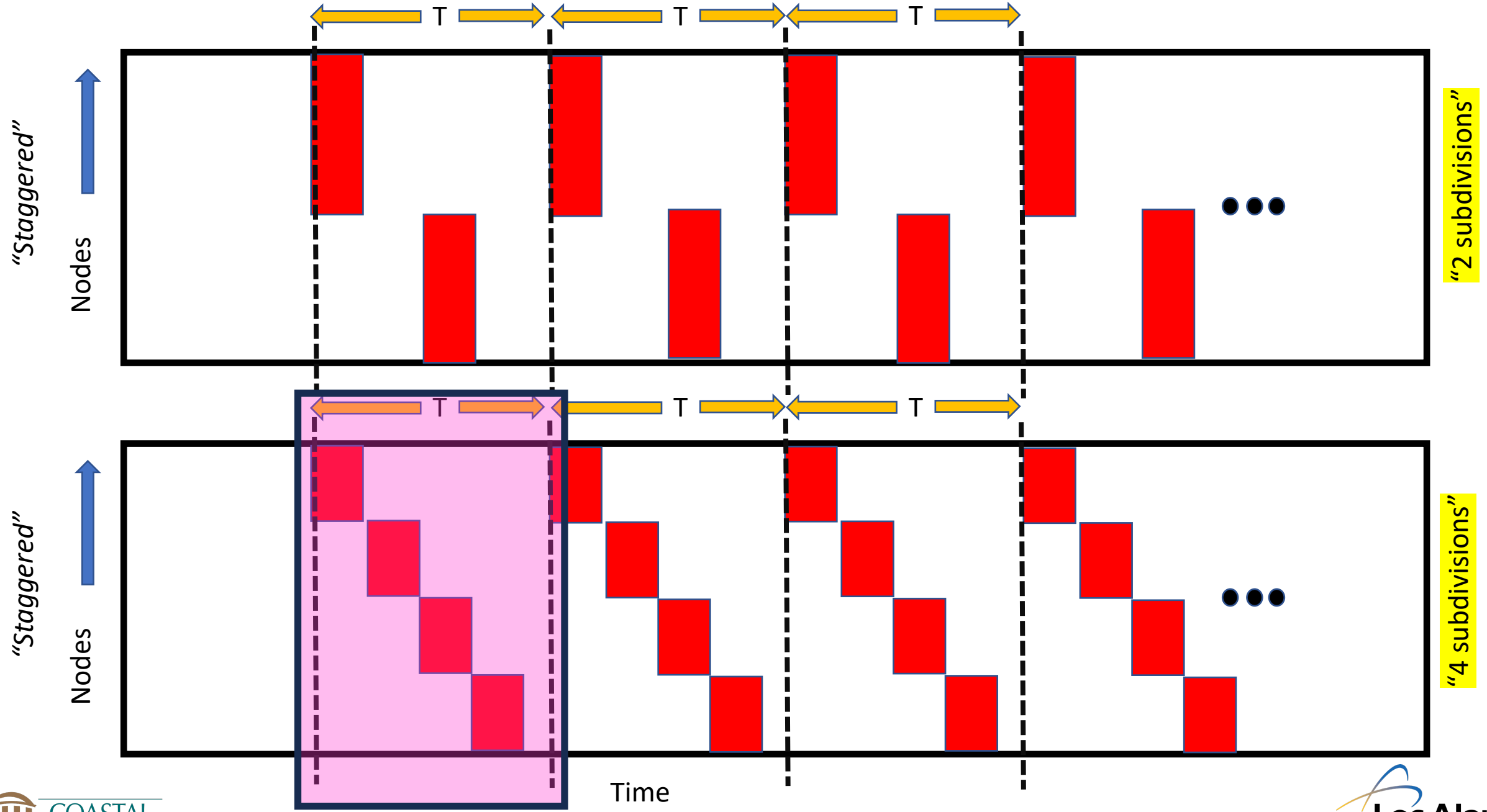

T
T
T
"Full"
Nodes
T
T
T
"Staggered"
Nodes
T
T
T
Time

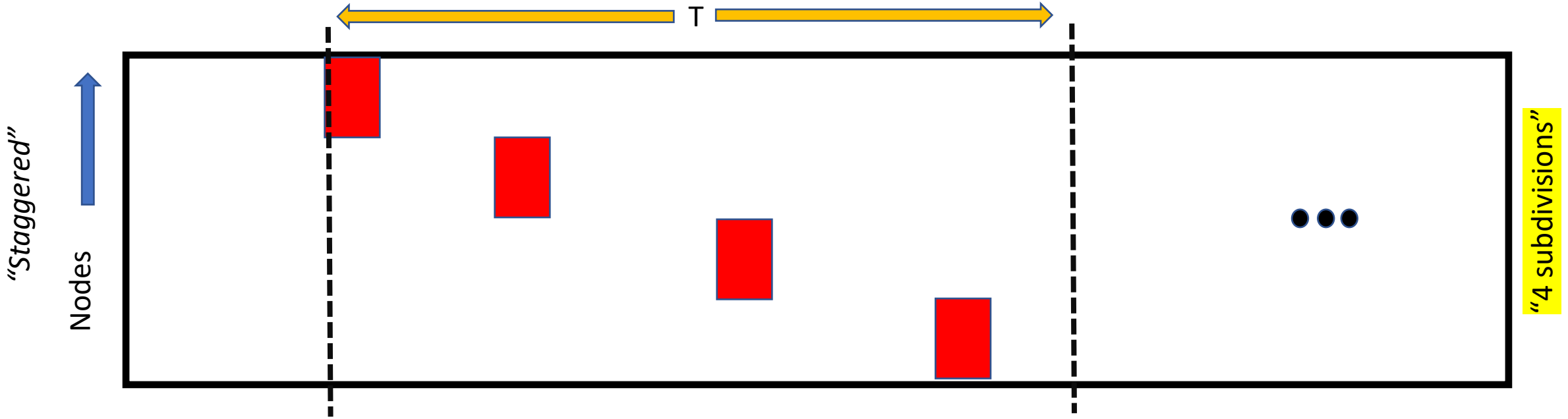

T
T
T
"Full"
Nodes
"1 subdivision"
T
T
T
"Staggered"
Nodes
T
T
T
"2 subdivisions"
Time

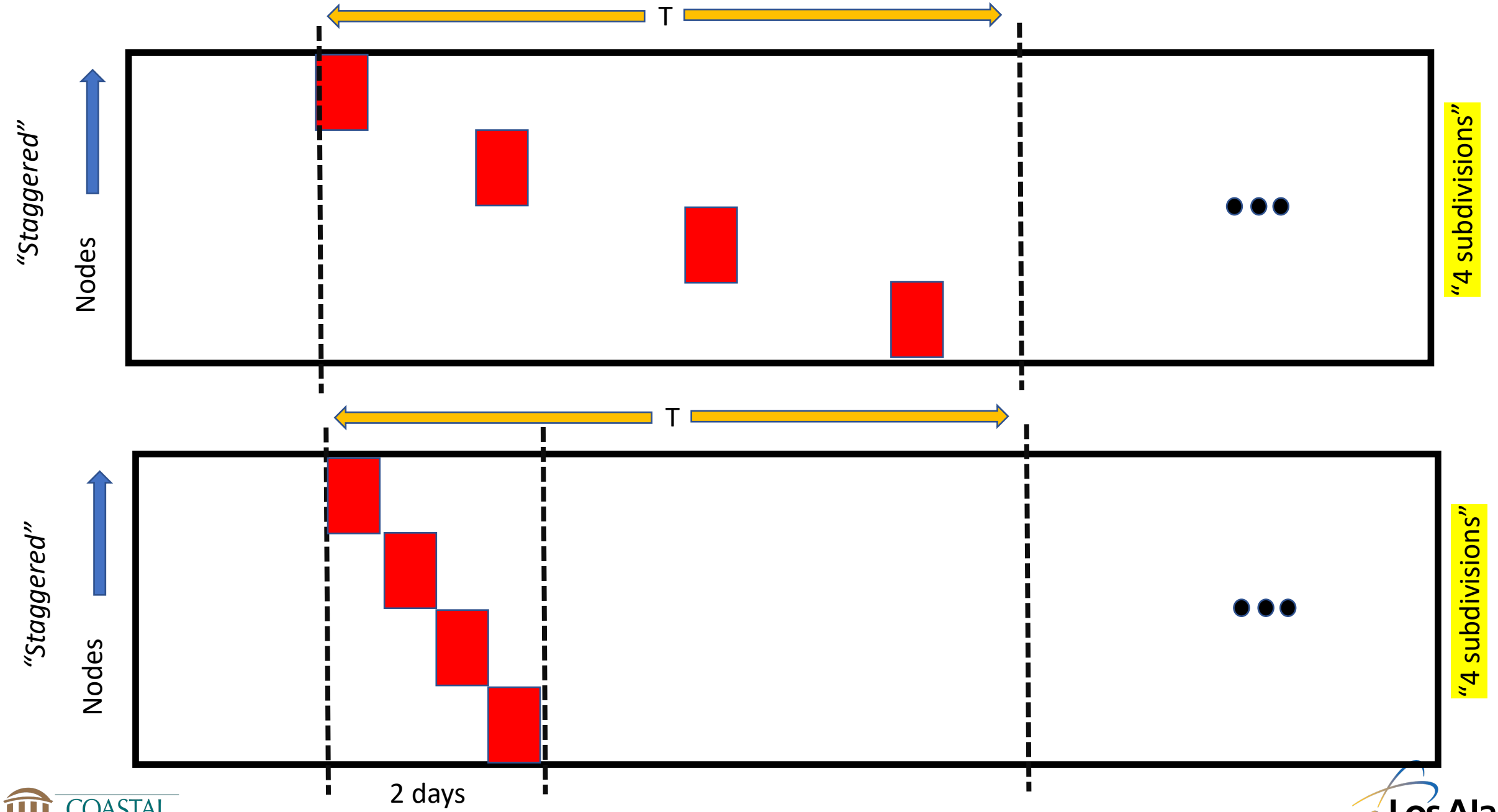

T
T
T
"Staggered"
Nodes
"2 subdivisions"
T
T
T
"Staggered"
Nodes
"4 subdivisions"
Time


T
T
T
1 month
1 month
"Staggered"
Nodes
"2 subdivisions"
T
T
T
"Staggered"
Nodes
"4 subdivisions"
Time


T
T
T
1 month
1 month
"Staggered"
Nodes
"2 subdivisions"
T
T
T
1 month
1 month
1 month
1 month
"Staggered"
Nodes
"4 subdivisions"
Time


T
T
T
"Staggered"
Nodes
"2 subdivisions"
T
T
T
"Staggered"
Nodes
"4 subdivisions"
Time


T
"Staggered"
Nodes
"4 subdivisions"


T
"Staggered"
Nodes
"4 subdivisions"
T
"Staggered"
Nodes
"4 subdivisions"
2 days

Parameter Sweeps

Parameter Sweeps

Reservation durations: 1, 15, 60, 240, and 480 minutes (5 cases)

Parameter Sweeps

Reservation durations: 1, 15, 60, 240, and 480 minutes (5 cases)

Number of subdivisions: 1, 2, 4, 8 (4 cases)

Parameter Sweeps

Reservation durations:	1, 15, 60, 240, and 480 minutes (5 cases)
Number of subdivisions:	1, 2, 4, 8 (4 cases)
Interval:	1 month (across entire res. cycle), {2, 4, 8} days (4 cases)

Parameter Sweeps

Reservation durations:	1, 15, 60, 240, and 480 minutes (5 cases)
Number of subdivisions:	1, 2, 4, 8 (4 cases)
Interval:	1 month (across entire res. cycle), {2, 4, 8} days (4 cases)

Parameter Sweeps

Reservation durations: 1, 15, 60, 240, and 480 minutes (5 cases)

Number of subdivisions: 1, 2, 4, 8 (4 cases)

Interval: 1 month (across entire res. cycle), {2, 4, 8} days (4 cases)

80 total combinations

Parameter Sweeps

Reservation durations: 1, 15, 60, 240, and 480 minutes (5 cases)

Number of subdivisions: 1, 2, 4, 8 (4 cases)

Interval: 1 month (across entire res. cycle), {2, 4, 8} days (4 cases)

80 total combinations

Grizzly 2018 logs

Grizzly 2022 logs

Parameter Sweeps

Reservation durations: 1, 15, 60, 240, and 480 minutes (5 cases)

Number of subdivisions: 1, 2, 4, 8 (4 cases)

Interval: 1 month (across entire res. cycle), {2, 4, 8} days (4 cases)

80 total combinations

Grizzly 2018 logs → 25 Monte Carlo runs per experiment

Grizzly 2022 logs

Parameter Sweeps

Reservation durations: 1, 15, 60, 240, and 480 minutes (5 cases)

Number of subdivisions: 1, 2, 4, 8 (4 cases)

Interval: 1 month (across entire res. cycle), {2, 4, 8} days (4 cases)

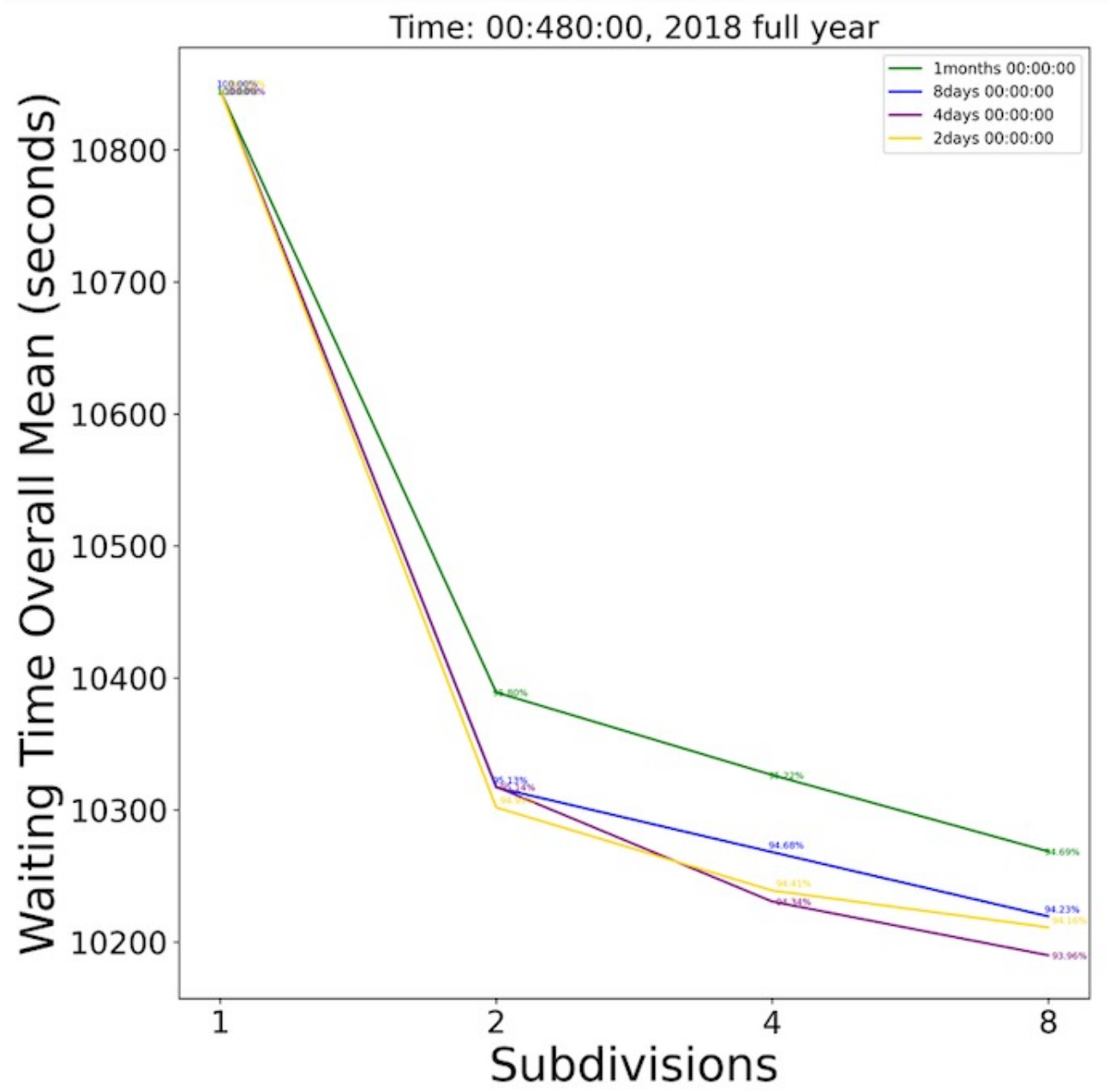
80 total combinations

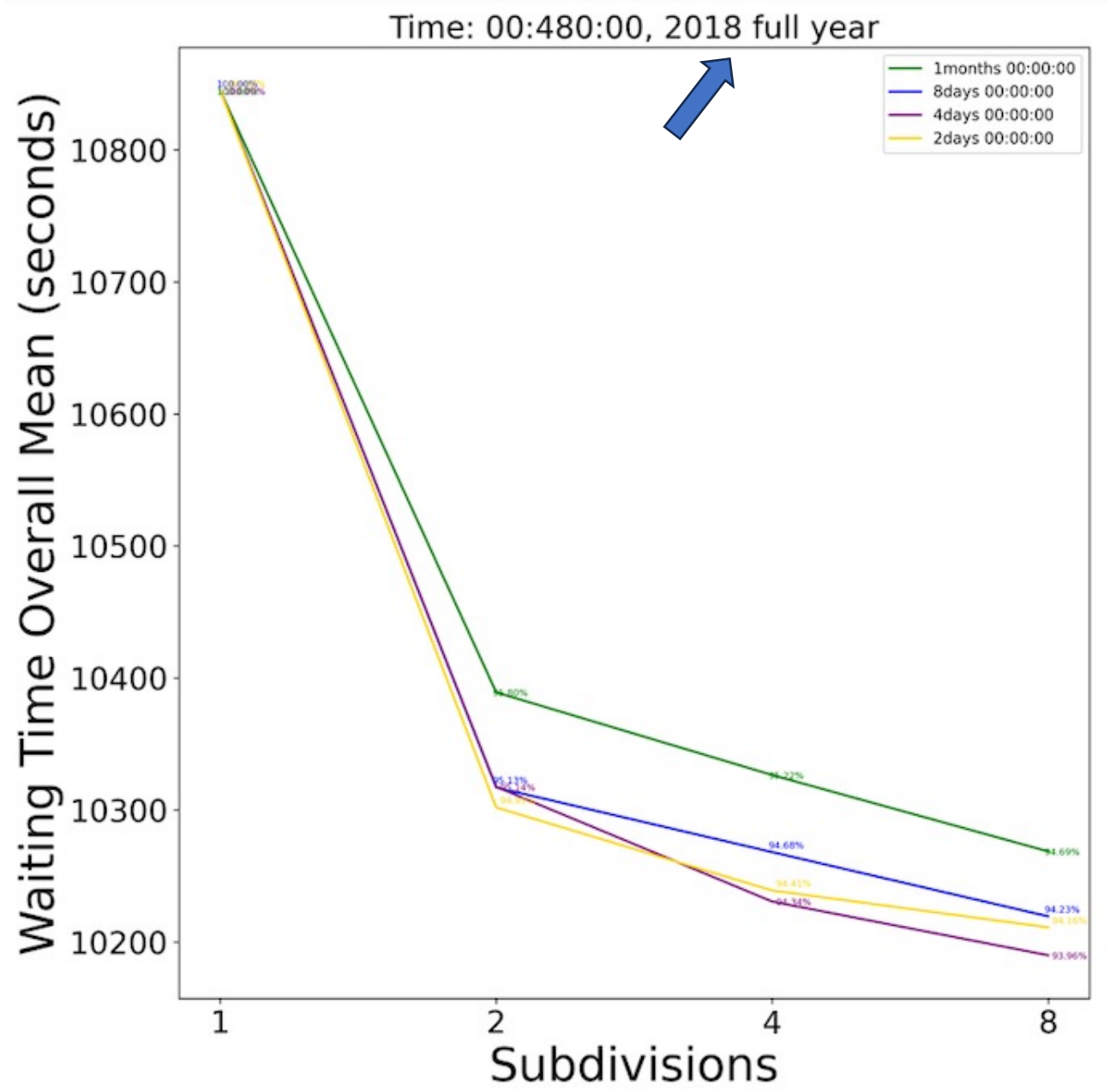
Grizzly 2018 logs → 25 Monte Carlo runs per experiment

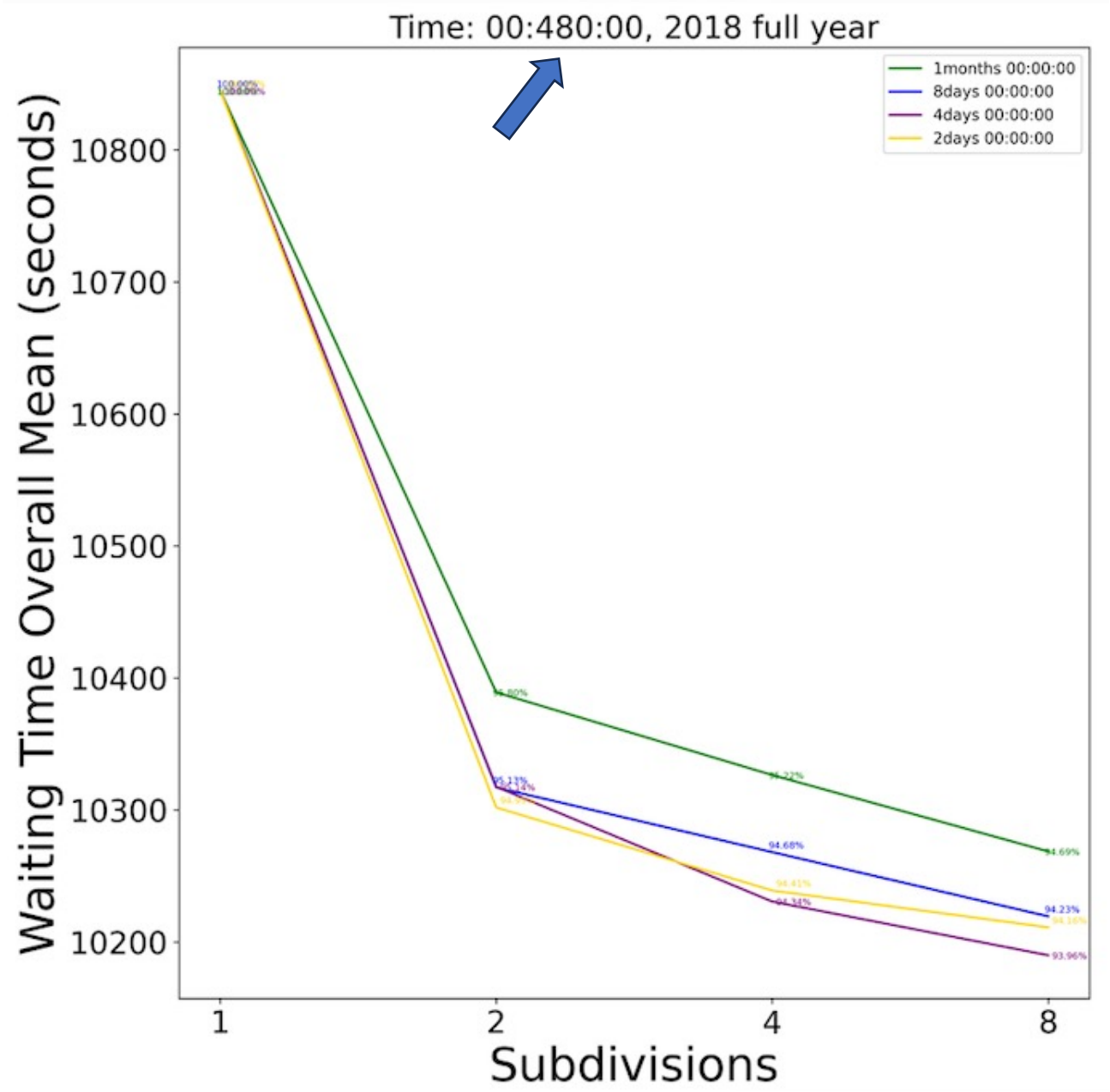
Grizzly 2022 logs

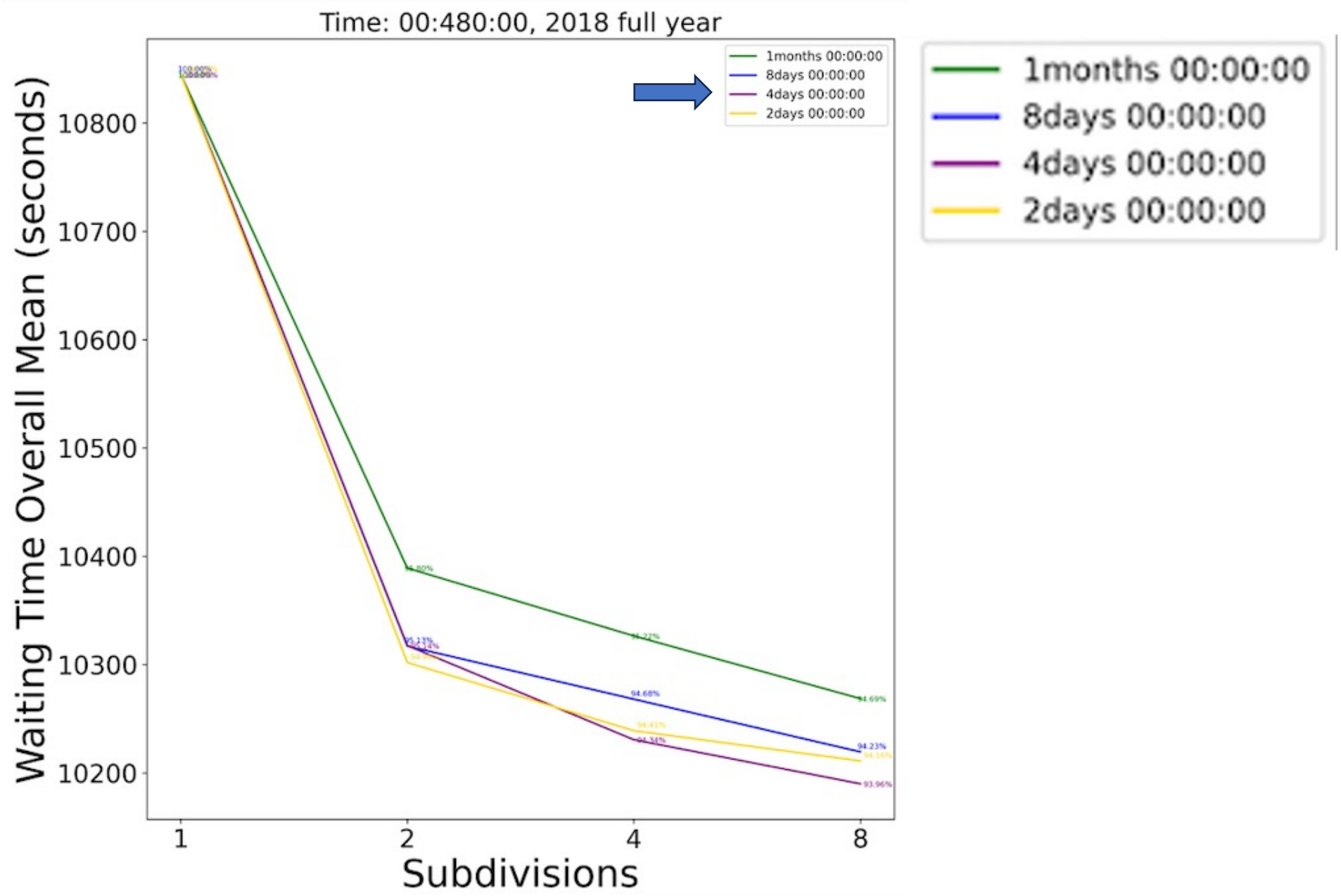
Very resource intensive – 50-100GB per experiment,
3-8 days to complete.

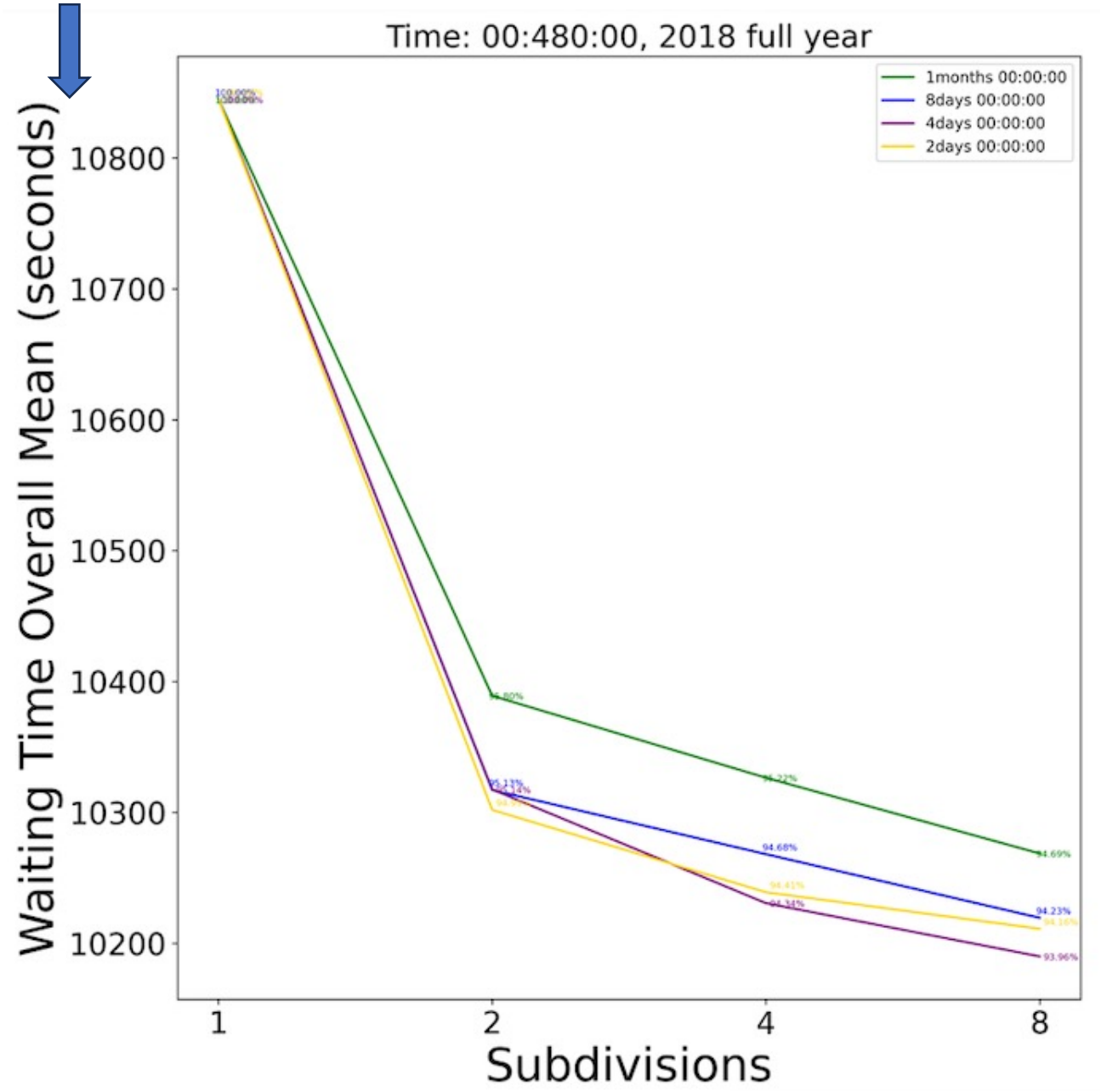
As compared to <20GB and 5 minutes for earlier experiments.

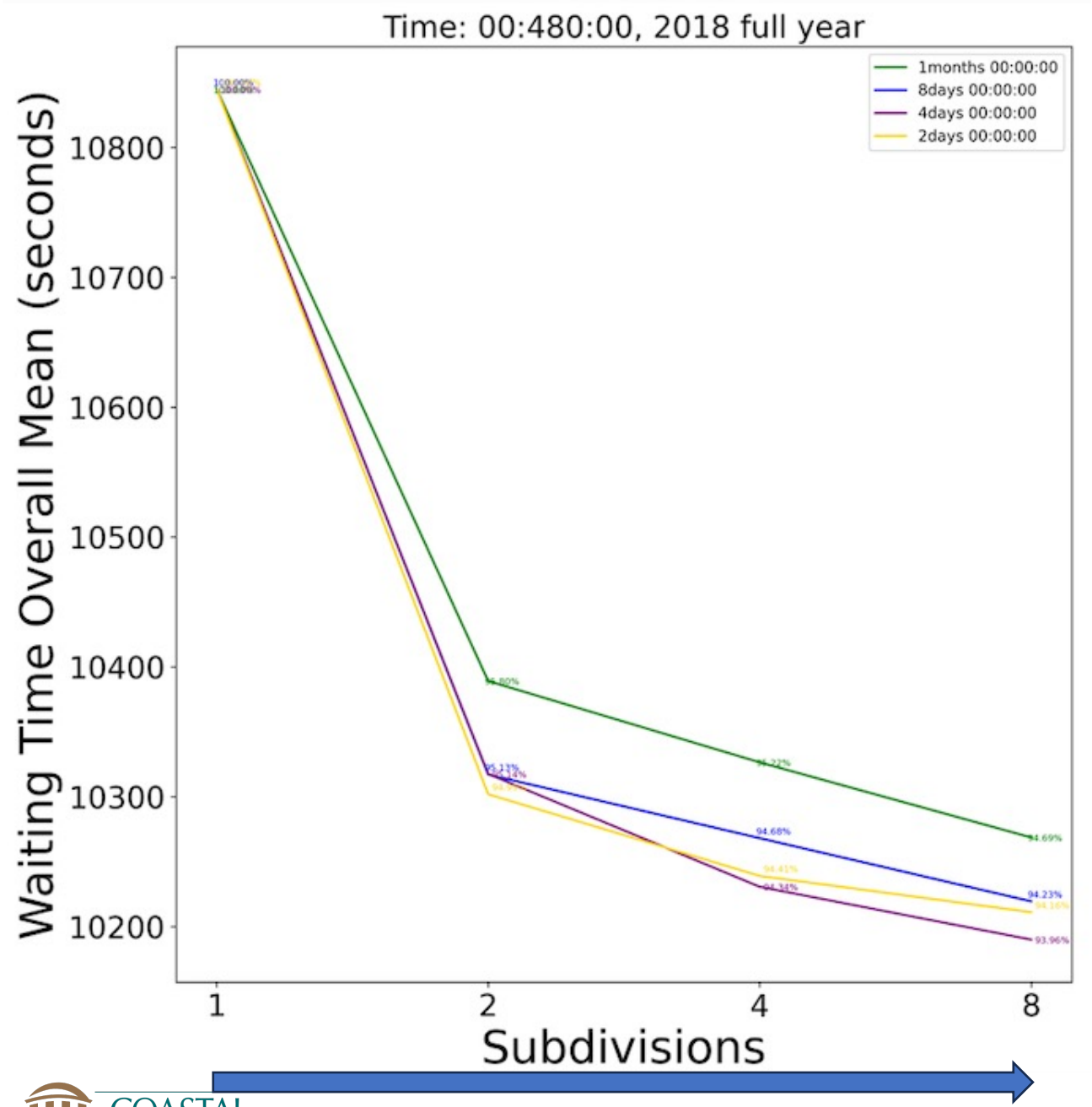

Time: 00:480:00, 2018 full year
1months 00:00:00
8days 00:00:00
4days 00:00:00
2days 00:00:00
Waiting Time Overall Mean (seconds)
10800
10700
10600
10500
10400
10300
10200
1
2
4
8
Subdivisions
94.69%
94.68%
94.23%
93.96%

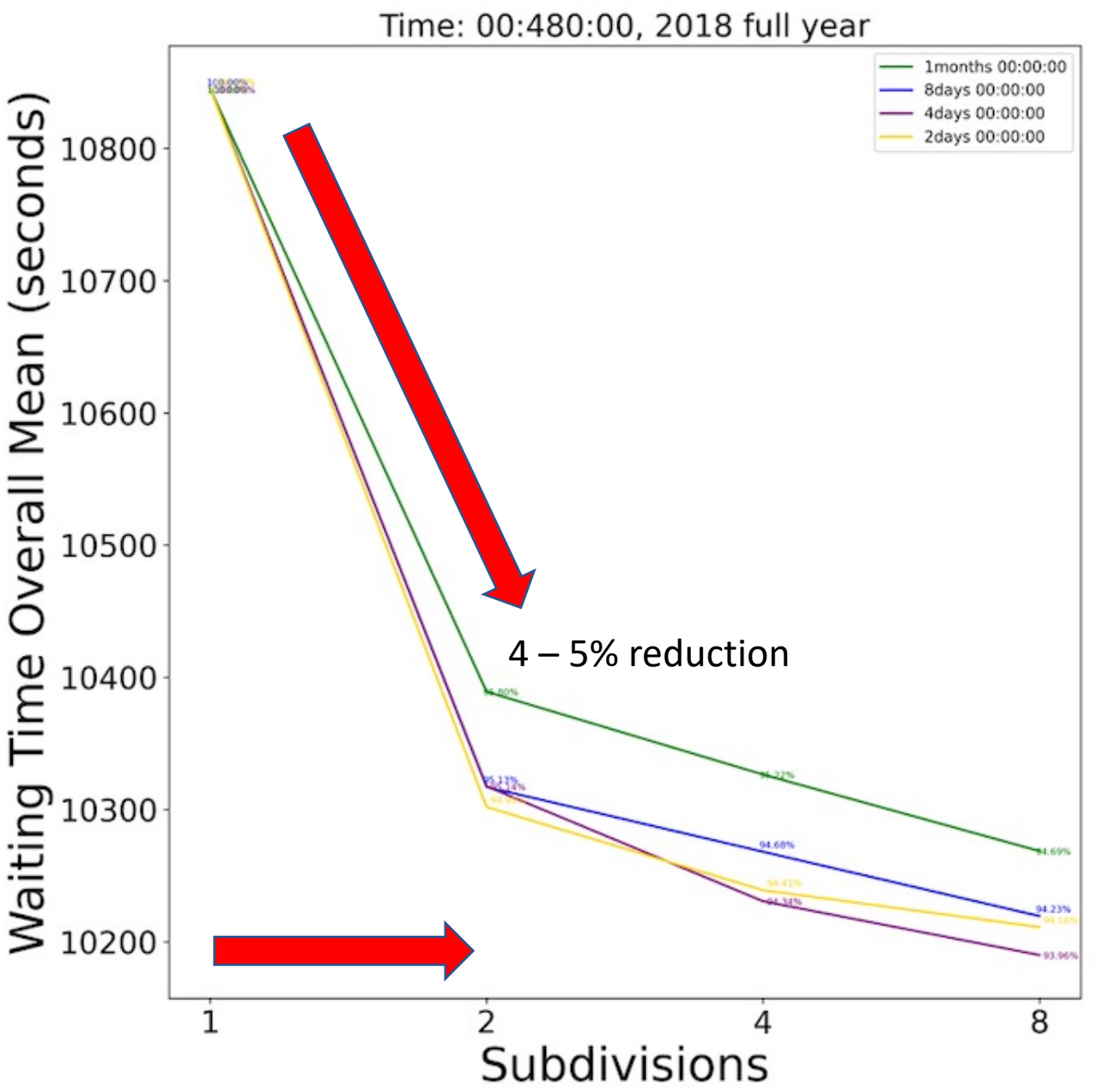

Time: 00:480:00, 2018 full year
1months 00:00:00
8days 00:00:00
4days 00:00:00
2days 00:00:00
Waiting Time Overall Mean (seconds)
10800
10700
10600
10500
10400
10300
10200
1
2
4
8
Subdivisions
94.69%
94.68%
94.23%
93.96%

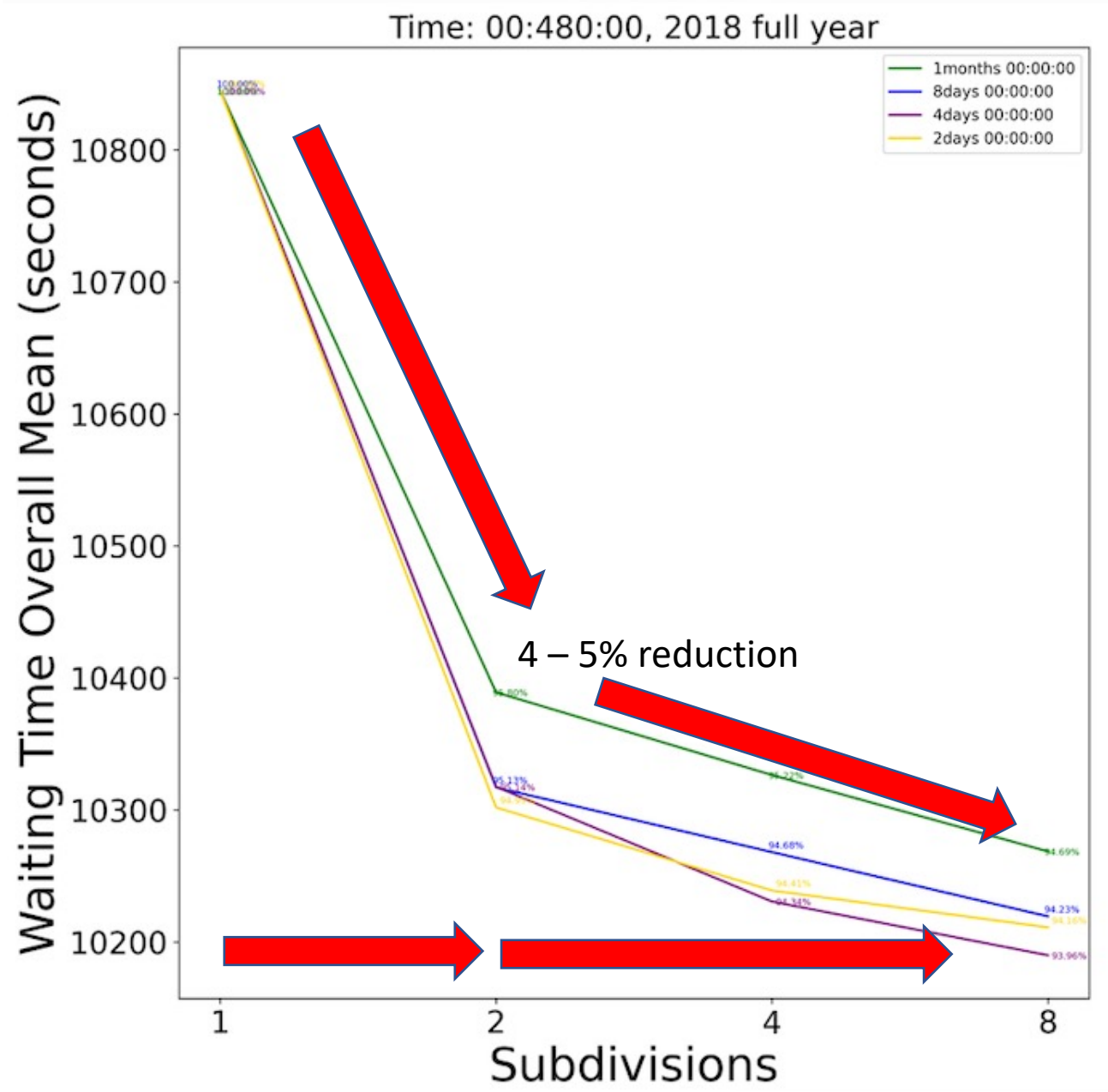

Time: 00:480:00, 2018 full year
1months 00:00:00
8days 00:00:00
4days 00:00:00
2days 00:00:00
Waiting Time Overall Mean (seconds)
10800
10700
10600
10500
10400
10300
10200
1
2
4
8
Subdivisions
94.69%
94.68%
94.23%
93.96%

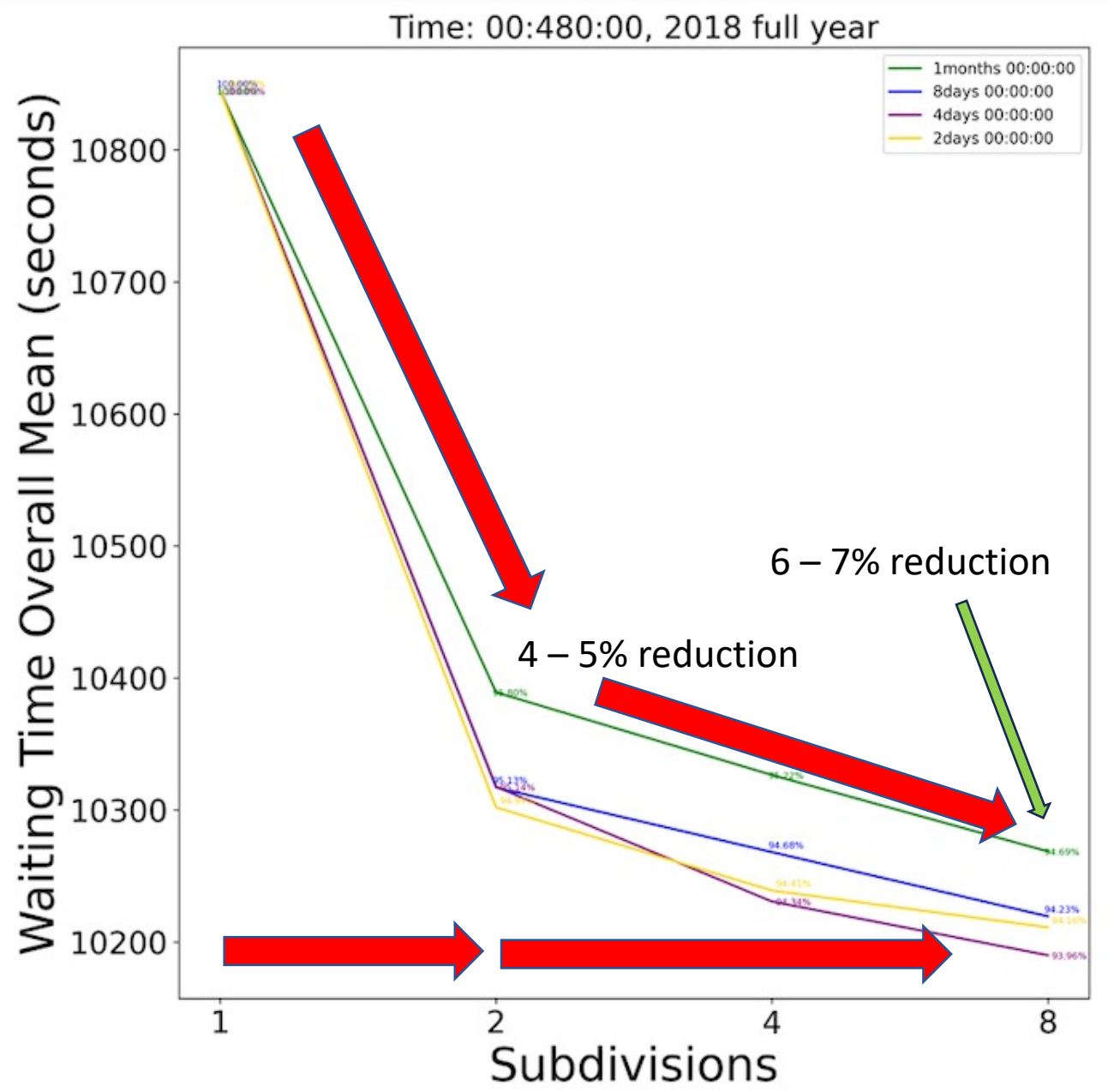

Time: 00:480:00, 2018 full year
Waiting Time Overall Mean (seconds)
10800
10700
10600
10500
10400
10300
10200
1
2
4
8
Subdivisions
1months 00:00:00
8days 00:00:00
4days 00:00:00
2days 00:00:00
94.69%
94.68%
94.23%
93.96%

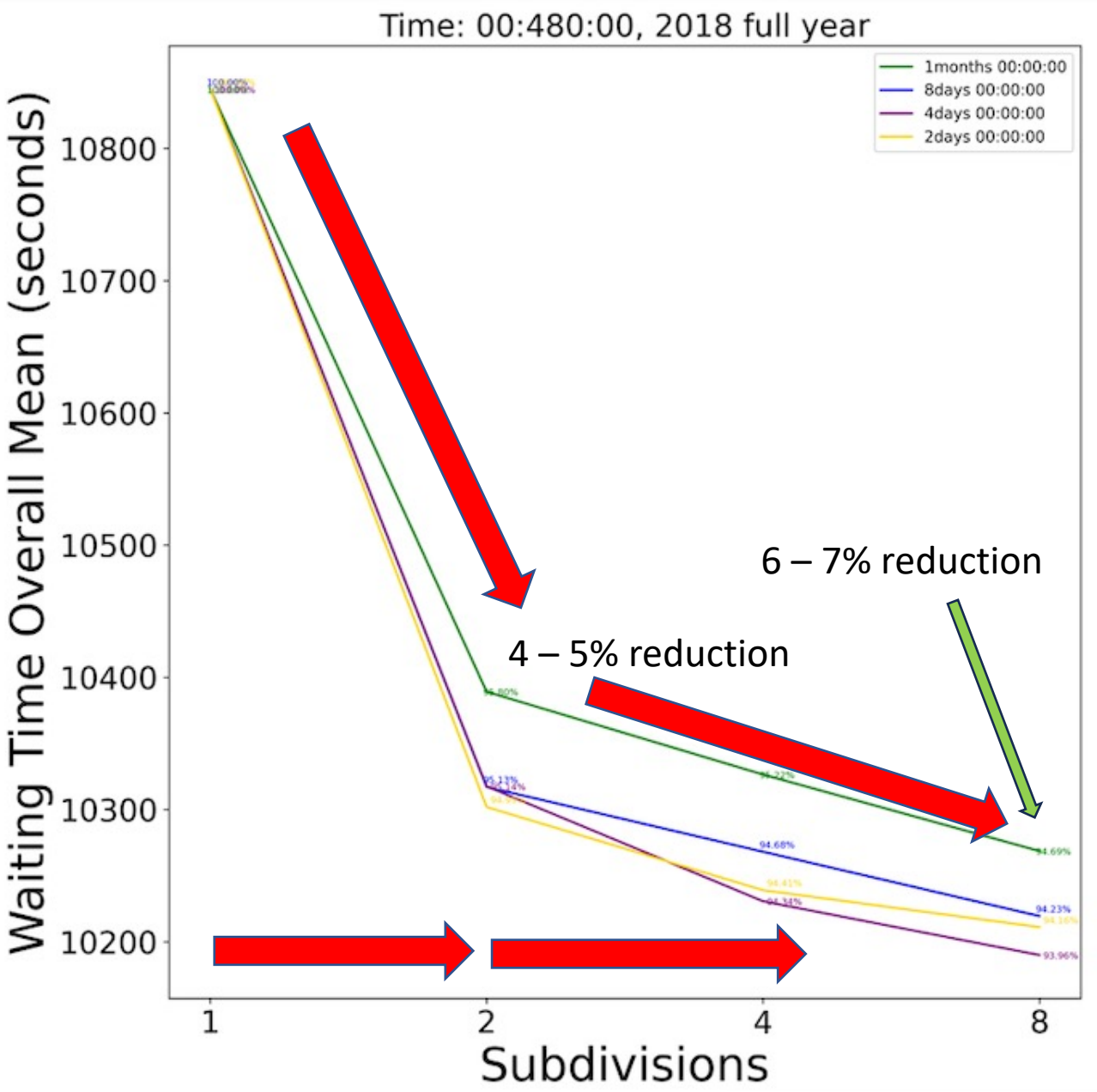

Time: 00:480:00, 2018 full year
1months 00:00:00
8days 00:00:00
4days 00:00:00
2days 00:00:00
Waiting Time Overall Mean (seconds)
10800
10700
10600
10500
10400
10300
10200
1
2
4
8
Subdivisions
94.69%
94.68%
94.23%
93.96%


Time: 00:480:00, 2018 full year
1months 00:00:00
8days 00:00:00
4days 00:00:00
2days 00:00:00
Waiting Time Overall Mean (seconds)
10800
10700
10600
10500
10400
10300
10200
1
2
4
8
Subdivisions
94.69%
94.68%
94.23%
93.96%


Time: 00:480:00, 2018 full year
1months 00:00:00
8days 00:00:00
4days 00:00:00
2days 00:00:00
Waiting Time Overall Mean (seconds)
10800
10700
10600
10500
10400
10300
10200
1
2
4
8
Subdivisions
4 – 5% reduction
94.69%
94.68%
94.23%
94.34%
93.96%

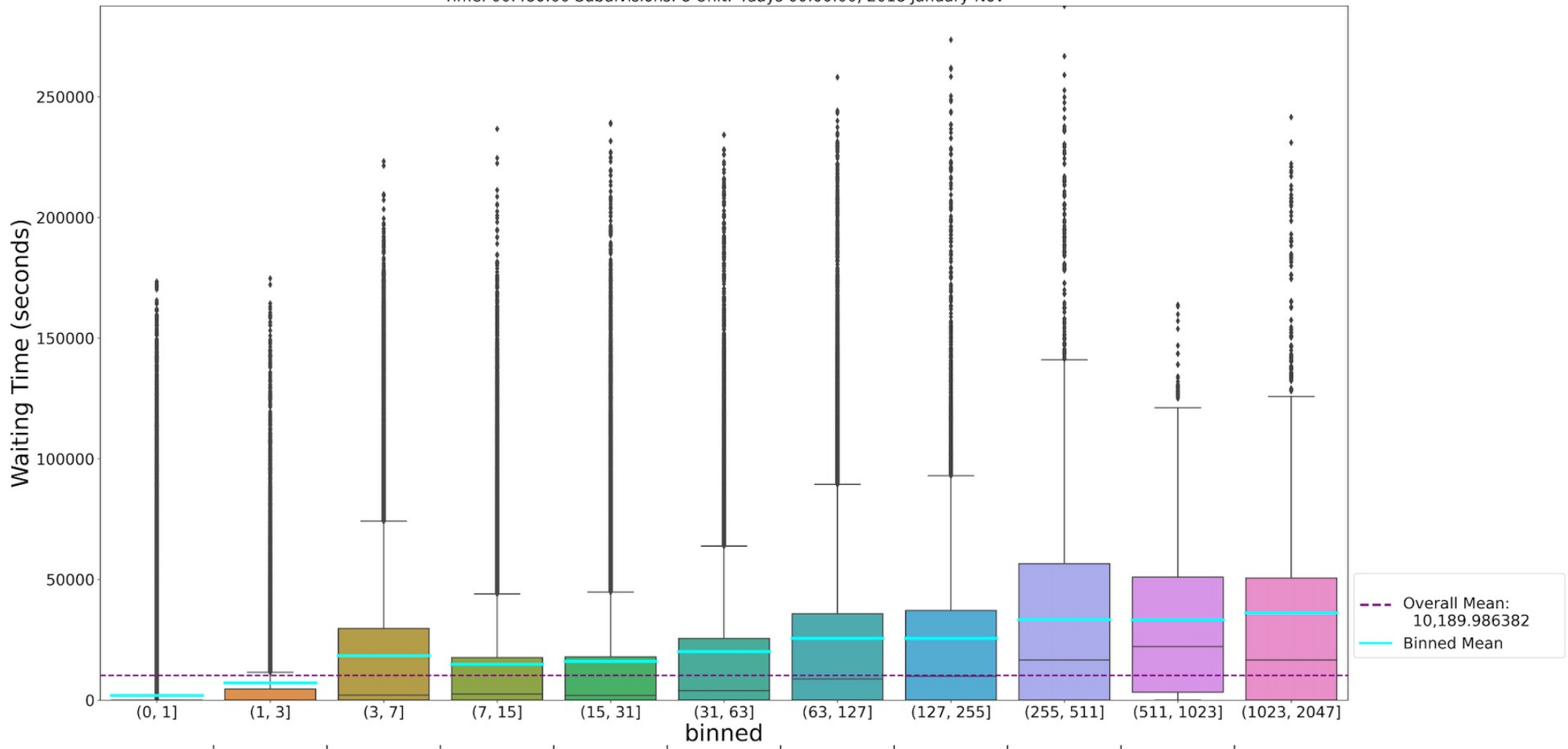

Time: 00:480:00, 2018 full year
1months 00:00:00
8days 00:00:00
4days 00:00:00
2days 00:00:00
Waiting Time Overall Mean (seconds)
10800
10700
10600
10500
10400
10300
10200
1
2
4
8
Subdivisions
4 – 5% reduction
94.69%
94.68%
94.23%
94.41%
94.34%
93.96%


Time: 00:480:00, 2018 full year
1months 00:00:00
8days 00:00:00
4days 00:00:00
2days 00:00:00
Waiting Time Overall Mean (seconds)
10800
10700
10600
10500
10400
10300
10200
1
2
4
8
Subdivisions
6 – 7% reduction
4 – 5% reduction
94.69%
94.68%
94.23%
93.96%



Drill down on impact to jobs in the 4-day interval simulations with 8 subdivisions with reservations of length 480 minutes

Time: 00:480:00 Subdivisions: 8 Unit: 4days 00:00:00, 2018 January-Nov



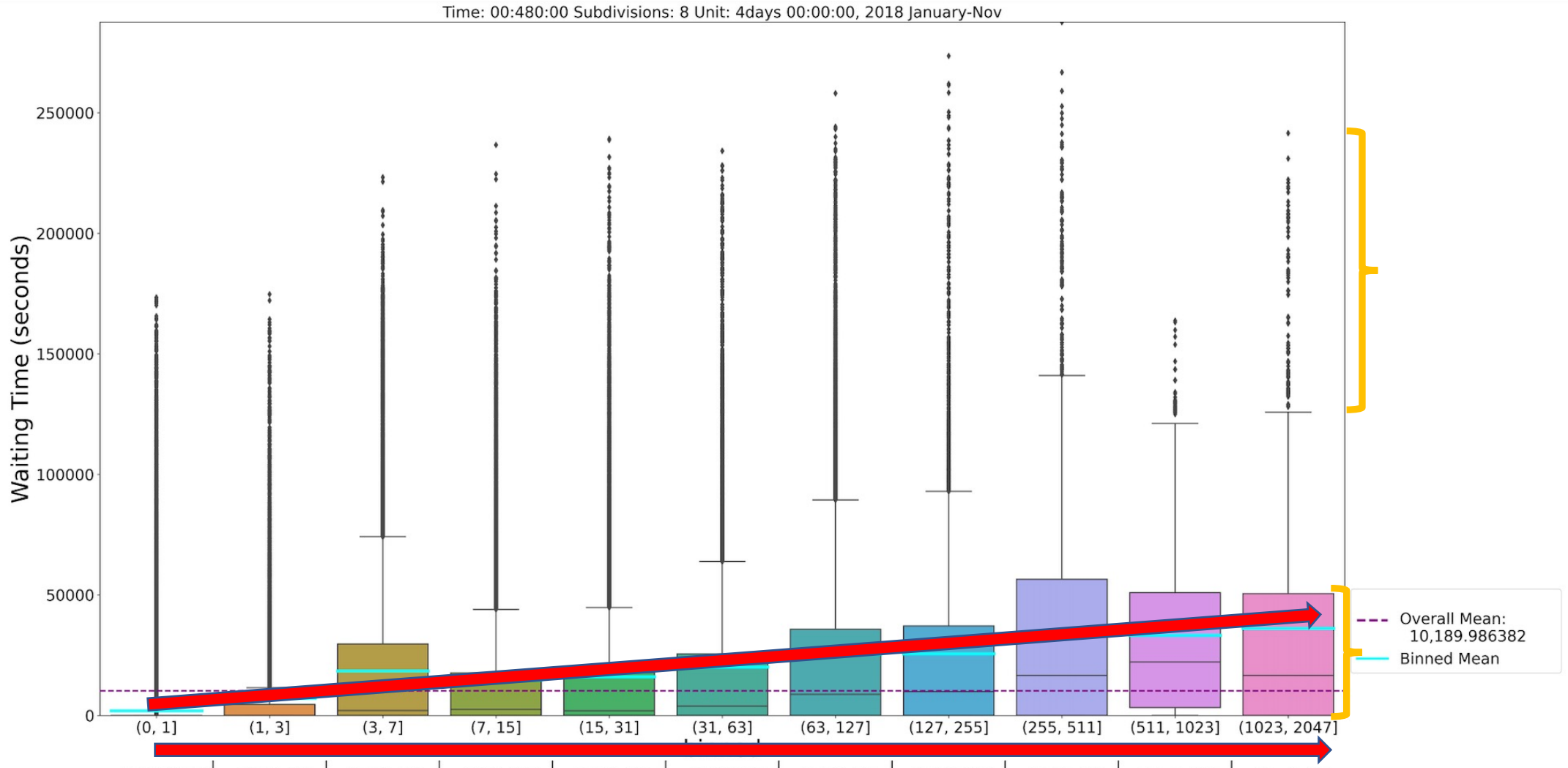
	n=1,712,900	n=143,450	n=696,550	n=241,225	n=227,600	n=125,975	n=252,700	n= 32,225	n= 20,675	n= 3,825	n= 4,425
Bin Mean:	μ= 1,945.59	μ= 7,133.92	μ= 18,458.80	μ= 14,855.02	μ= 16,075.87	μ= 20,140.29	μ= 25,629.46	μ= 25,641.38	μ= 33,384.33	μ= 33,255.89	μ= 36,156.35
Bin Max:	↑= 173,477.00	↑= 174,750.00	↑= 223,274.00	↑= 236,643.00	↑= 239,189.00	↑= 234,199.00	↑= 258,079.00	↑= 273,620.00	↑= 287,624.00	↑= 163,756.00	↑= 241,535.00
Bin Min:	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00
Bin Avg Max:	↕= 173,477.00	↕= 174,750.00	↕= 223,274.00	↕= 236,643.00	↕= 239,189.00	↕= 234,199.00	↕= 258,079.00	↕= 273,620.00	↕= 287,624.00	↕= 163,756.00	↕= 241,535.00
Bin Avg Min:	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00
Bin Std:	σ= 8,711.15	σ= 16,602.94	σ= 30,464.97	σ= 26,342.08	σ= 28,949.69	σ= 32,947.96	σ= 37,166.00	σ= 36,149.40	σ= 41,883.84	σ= 35,339.87	σ= 46,287.12

Bin Mean:	μ= 32.43 min	μ= 1.98 hrs	μ= 5.13 hrs	μ= 4.13 hrs	μ= 4.47 hrs	μ= 5.59 hrs	μ= 7.12 hrs	μ= 7.12 hrs	μ= 9.27 hrs	μ= 9.24 hrs	μ= 10.04 hrs
Bin Max:	↑= 2.01 days	↑= 2.02 days	↑= 2.58 days	↑= 2.74 days	↑= 2.77 days	↑= 2.71 days	↑= 2.99 days	↑= 3.17 days	↑= 3.33 days	↑= 1.90 days	↑= 2.80 days
Bin Min:	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec
Bin Avg Max:	↕= 2.01 days	↕= 2.02 days	↕= 2.58 days	↕= 2.74 days	↕= 2.77 days	↕= 2.71 days	↕= 2.99 days	↕= 3.17 days	↕= 3.33 days	↕= 1.90 days	↕= 2.80 days
Bin Avg Min:	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec
Bin Std:	σ= 2.42 hrs	σ= 4.61 hrs	σ= 8.46 hrs	σ= 7.32 hrs	σ= 8.04 hrs	σ= 9.15 hrs	σ= 10.32 hrs	σ= 10.04 hrs	σ= 11.63 hrs	σ= 9.82 hrs	σ= 12.86 hrs

Overall Mean	μ= 10,189.99
Overall Max	↑= 287,624.00
Overall Min	↓= 0.00
Overall Avg Max	↕= 287,624.00
Overall Avg Min	↧= 0.00
Overall Std	σ= 24,202.54

Overall Mean	μ= 2.83 hrs
Overall Max	↑= 3.33 days
Overall Min	↓= 0.00 sec
Overall Avg Max	↕= 3.33 days
Overall Avg Min	↧= 0.00 sec
Overall Std	σ= 6.72 hrs

amos LABORATORY 943



	n=1,712,900	n=143,450	n=696,550	n=241,225	n=227,600	n=125,975	n=252,700	n= 32,225	n= 20,675	n= 3,825	n= 4,425
Bin Mean:	μ= 1,945.59	μ= 7,133.92	μ= 18,458.80	μ= 14,855.02	μ= 16,075.87	μ= 20,140.29	μ= 25,629.46	μ= 25,641.38	μ= 33,384.33	μ= 33,255.89	μ= 36,156.35
Bin Max:	↑= 173,477.00	↑= 174,750.00	↑= 223,274.00	↑= 236,643.00	↑= 239,189.00	↑= 234,199.00	↑= 258,079.00	↑= 273,620.00	↑= 287,624.00	↑= 163,756.00	↑= 241,535.00
Bin Min:	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00	↓= 0.00
Bin Avg Max:	↕= 173,477.00	↕= 174,750.00	↕= 223,274.00	↕= 236,643.00	↕= 239,189.00	↕= 234,199.00	↕= 258,079.00	↕= 273,620.00	↕= 287,624.00	↕= 163,756.00	↕= 241,535.00
Bin Avg Min:	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00	↧= 0.00
Bin Std:	σ= 8,711.15	σ= 16,602.94	σ= 30,464.97	σ= 26,342.08	σ= 28,949.69	σ= 32,947.96	σ= 37,166.00	σ= 36,149.40	σ= 41,883.84	σ= 35,339.87	σ= 46,287.12

Bin Mean:	μ= 32.43 min	μ= 1.98 hrs	μ= 5.13 hrs	μ= 4.13 hrs	μ= 4.47 hrs	μ= 5.59 hrs	μ= 7.12 hrs	μ= 7.12 hrs	μ= 9.27 hrs	μ= 9.24 hrs	μ= 10.04 hrs
Bin Max:	↑= 2.01 days	↑= 2.02 days	↑= 2.58 days	↑= 2.74 days	↑= 2.77 days	↑= 2.71 days	↑= 2.99 days	↑= 3.17 days	↑= 3.33 days	↑= 1.90 days	↑= 2.80 days
Bin Min:	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec	↓= 0.00 sec
Bin Avg Max:	↕= 2.01 days	↕= 2.02 days	↕= 2.58 days	↕= 2.74 days	↕= 2.77 days	↕= 2.71 days	↕= 2.99 days	↕= 3.17 days	↕= 3.33 days	↕= 1.90 days	↕= 2.80 days
Bin Avg Min:	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec	↧= 0.00 sec
Bin Std:	σ= 2.42 hrs	σ= 4.61 hrs	σ= 8.46 hrs	σ= 7.32 hrs	σ= 8.04 hrs	σ= 9.15 hrs	σ= 10.32 hrs	σ= 10.04 hrs	σ= 11.63 hrs	σ= 9.82 hrs	σ= 12.86 hrs

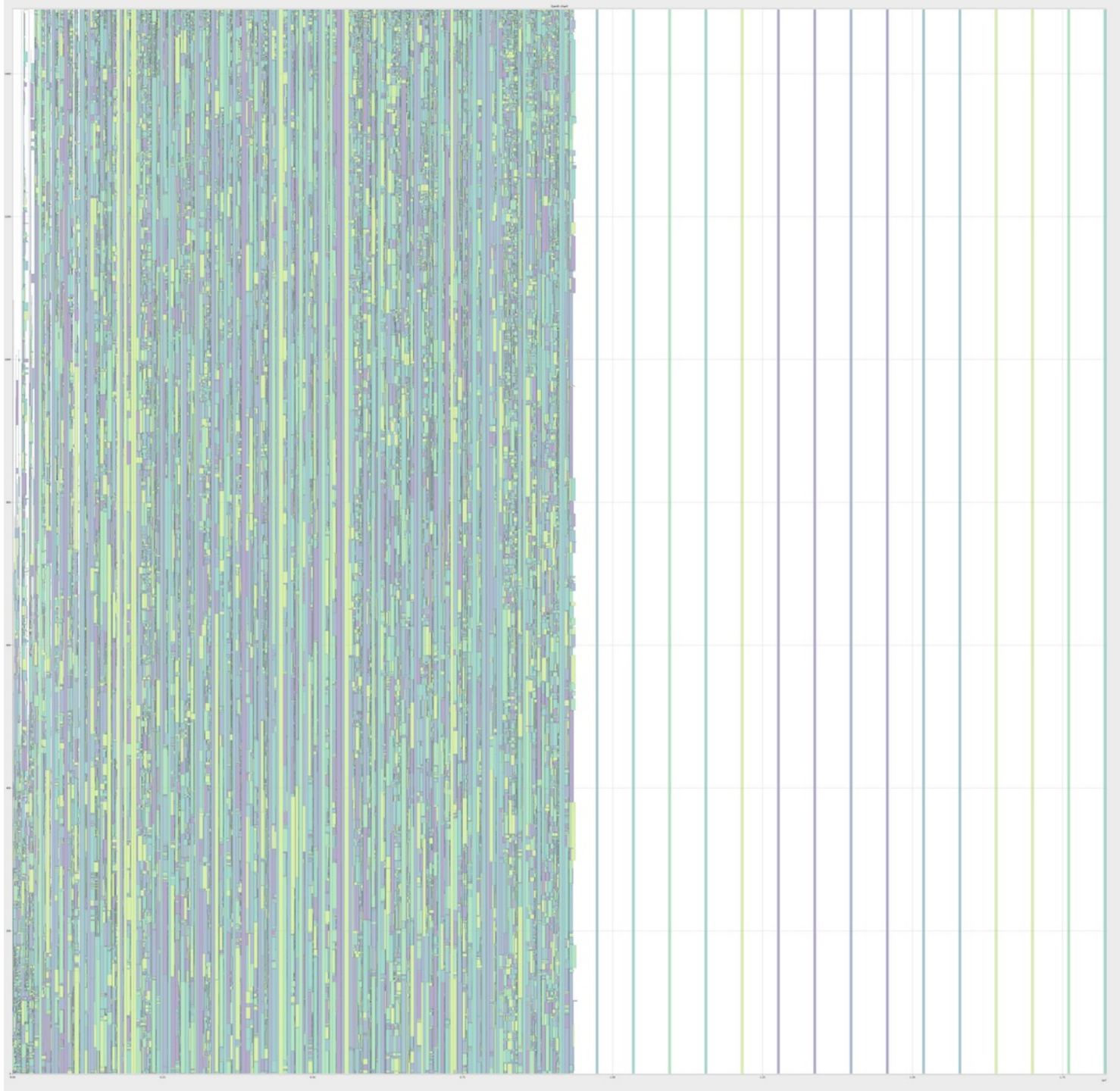
Overall Mean	μ= 10,189.99
Overall Max	↑= 287,624.00
Overall Min	↓= 0.00
Overall Avg Max	↕= 287,624.00
Overall Avg Min	↧= 0.00
Overall Std	σ= 24,202.54

Overall Mean	μ= 2.83 hrs
Overall Max	↑= 3.33 days
Overall Min	↓= 0.00 sec
Overall Avg Max	↕= 3.33 days
Overall Avg Min	↧= 0.00 sec
Overall Std	σ= 6.72 hrs

What's happening 'around' these reservations?

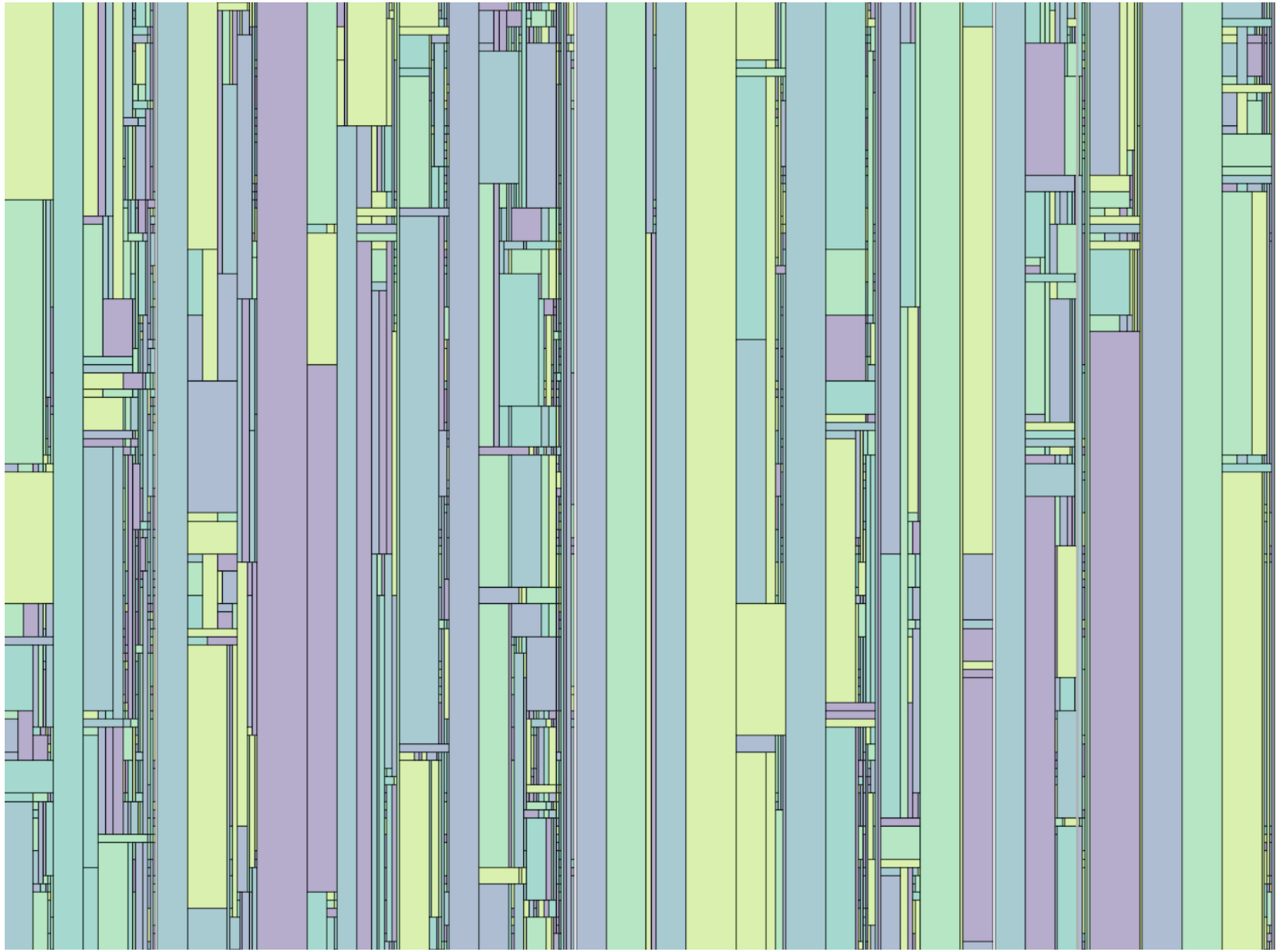
What's happening 'around' these reservations?

Determine impact to different job classes



COASTAL CAROLINA UNIVERSITY

Los Alamos NATIONAL LABORATORY EST. 1943



Evalys

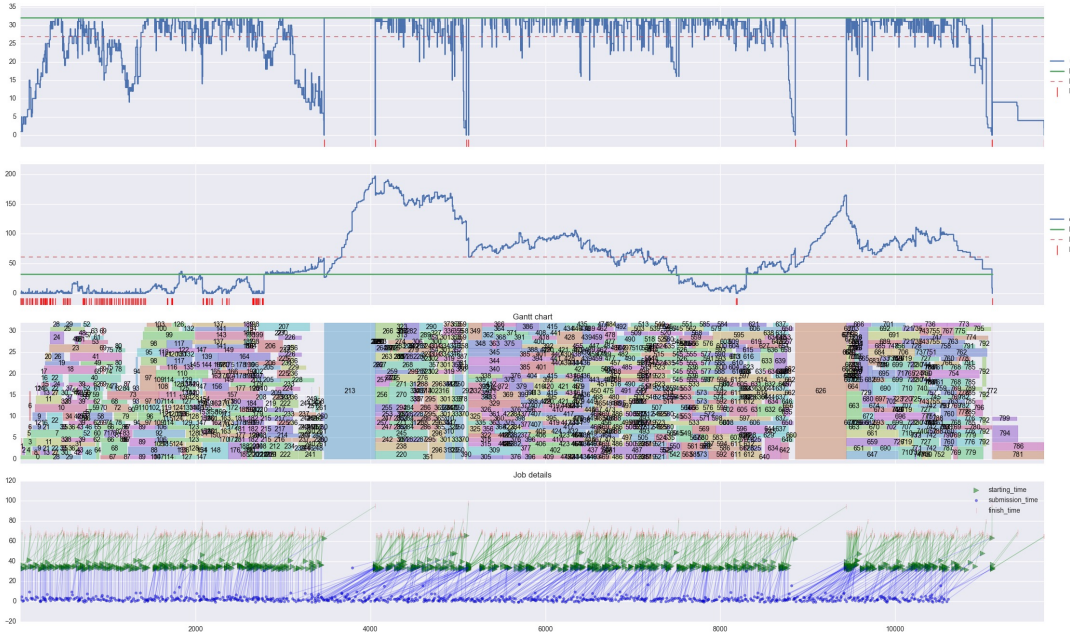
Evalys - Overview

pypi v4.0.6

Offline or online cluster monitoring

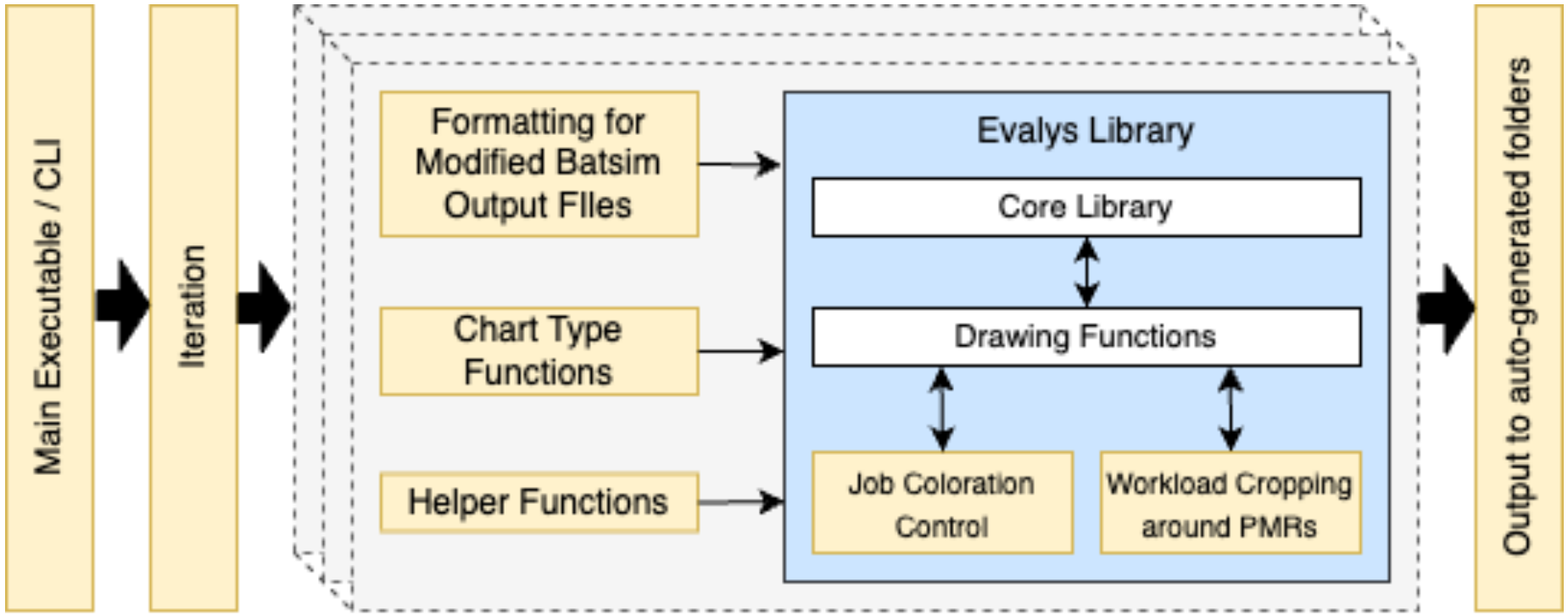
Features

- Load and all Batsim outputs files
  - Compute and plot free slots
  - Simple Gantt visualisation
  - Compute utilisation / queue
  - Compute fragmentation
  - Plot energy and machine state
- Load SWF workload files from Parallel Workloads Archive
  - Compute standard scheduling metrics
  - Show job details
  - Extract periods with a given mean utilisation

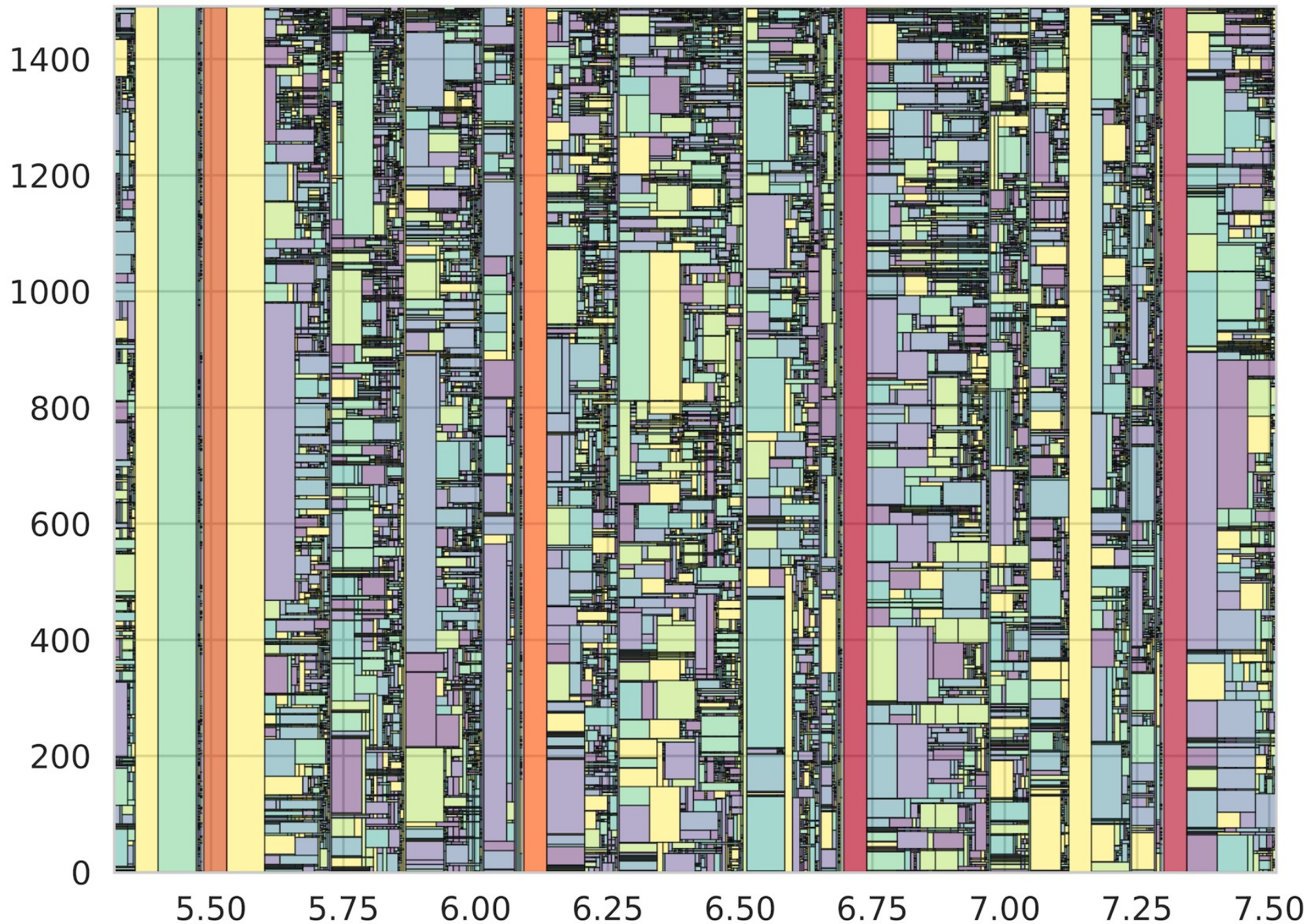


from gallery

Modifications to Evalys



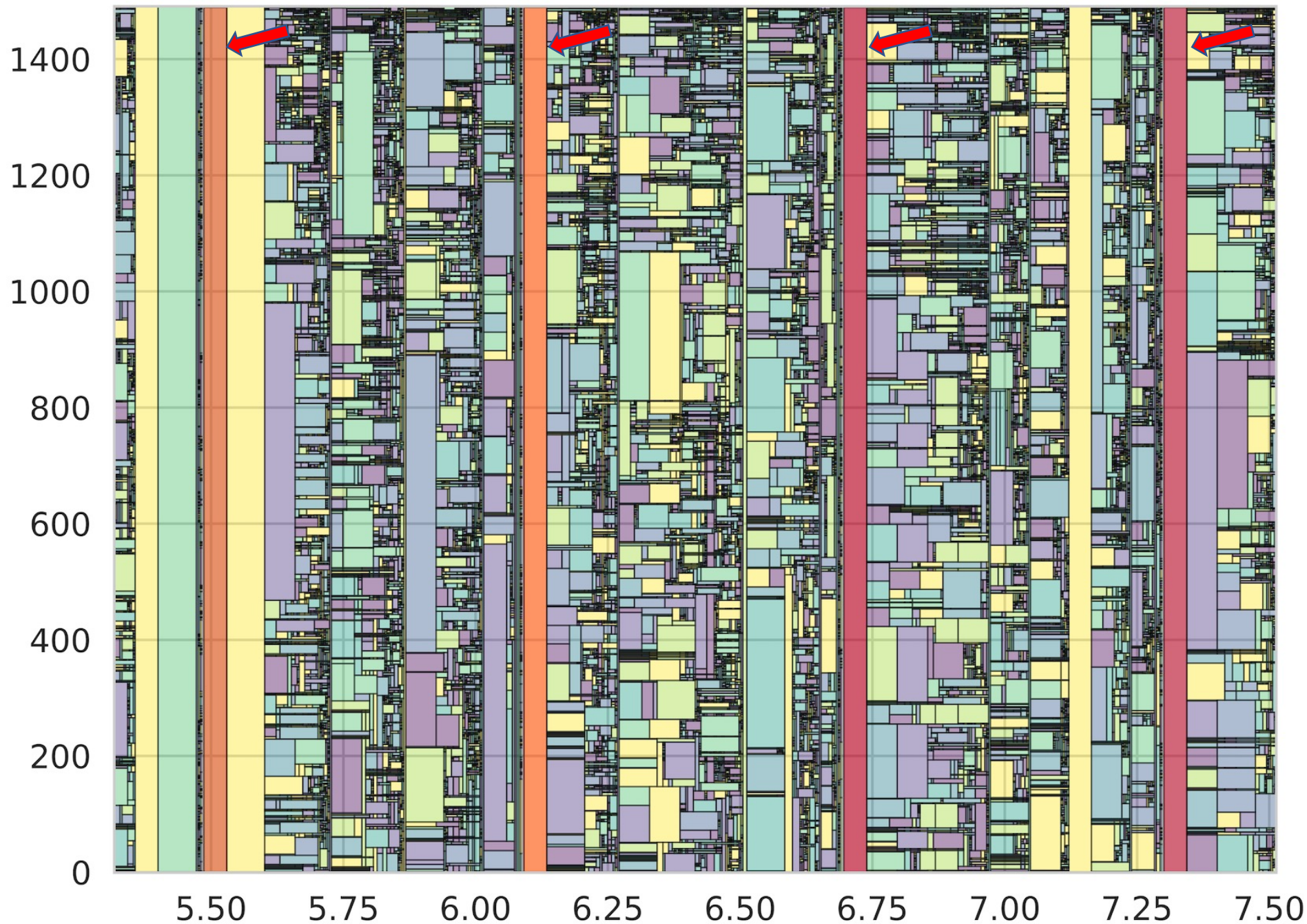
12-hour
Full



COASTAL CAROLINA UNIVERSITY

Los Alamos NATIONAL LABORATORY EST.1943

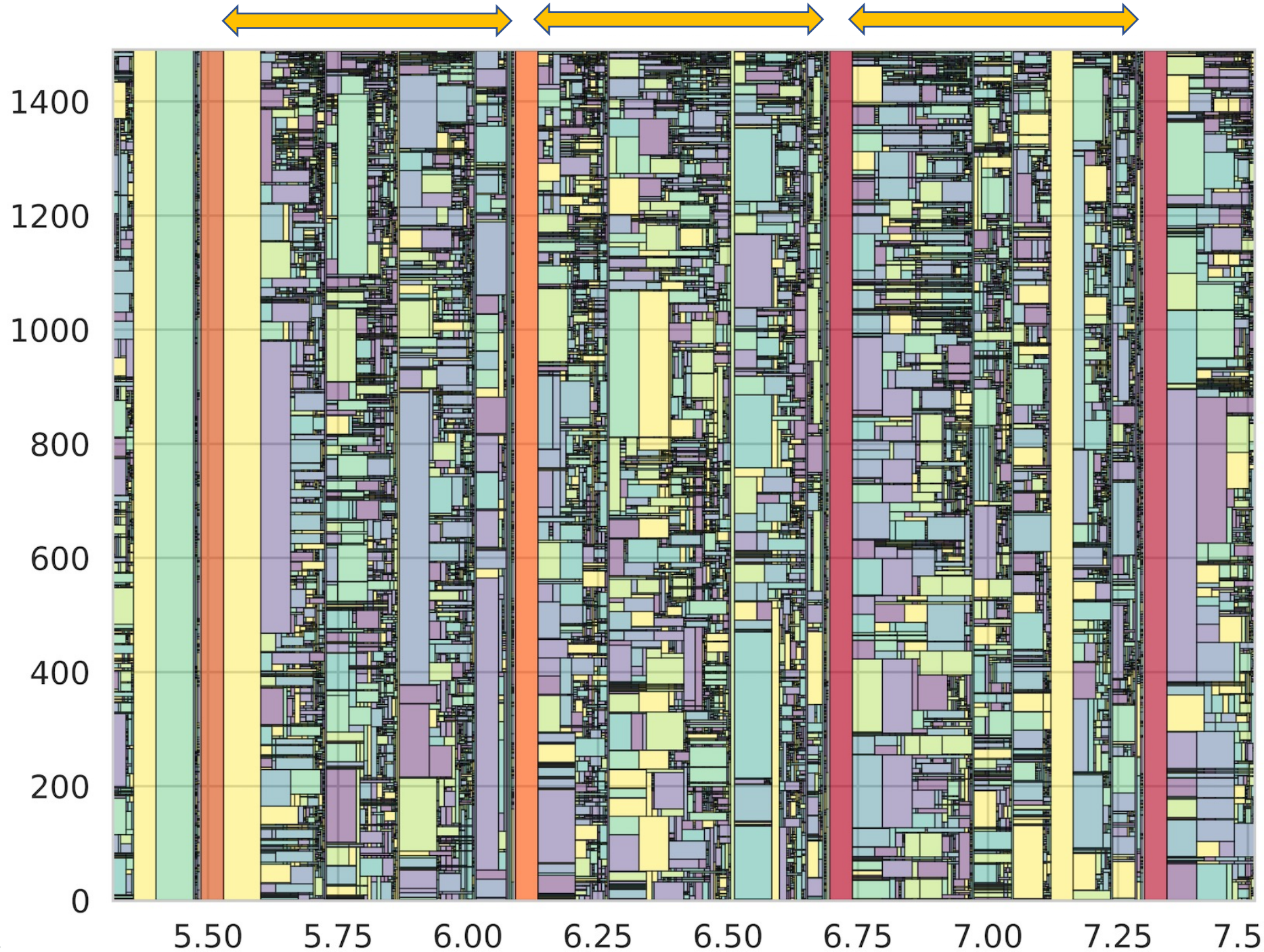
12-hour
Full



COASTAL CAROLINA UNIVERSITY

Los Alamos NATIONAL LABORATORY EST.1943

12-hour
Full

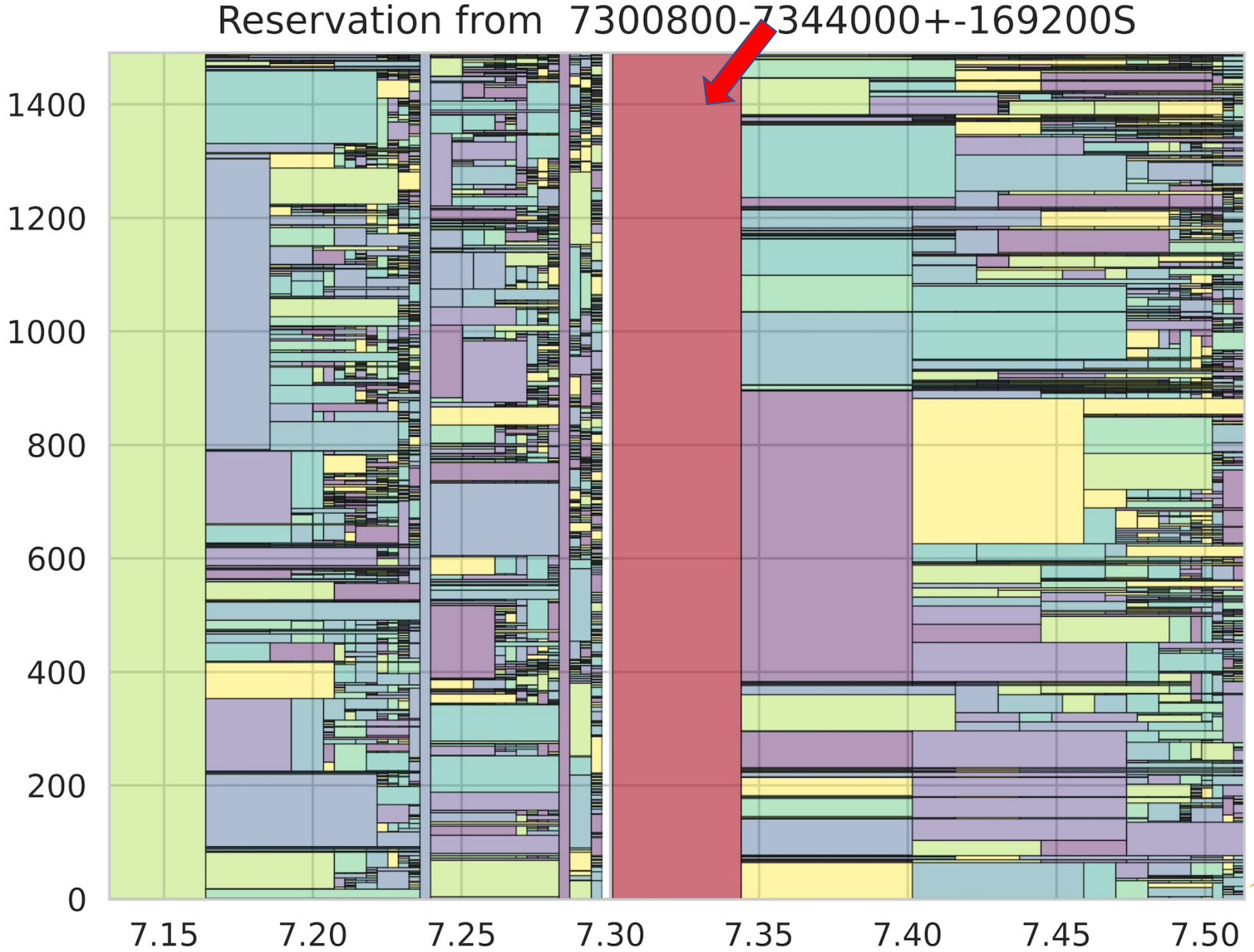


Red-ish = Res.

COASTAL CAROLINA UNIVERSITY

Los Alamos NATIONAL LABORATORY EST. 1943

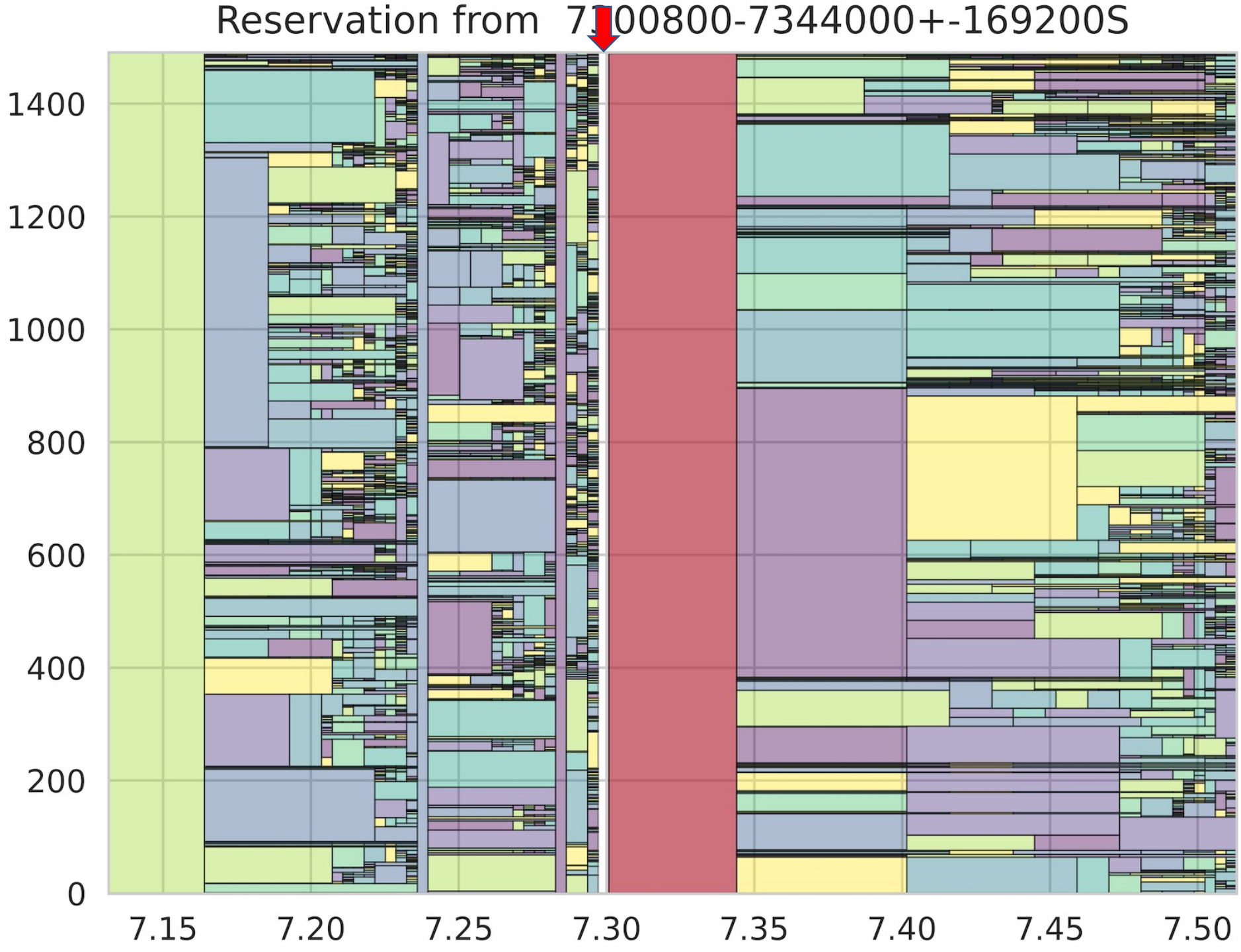
12-hour
Full



COASTAL CAROLINA UNIVERSITY

Los Alamos NATIONAL LABORATORY EST.1943

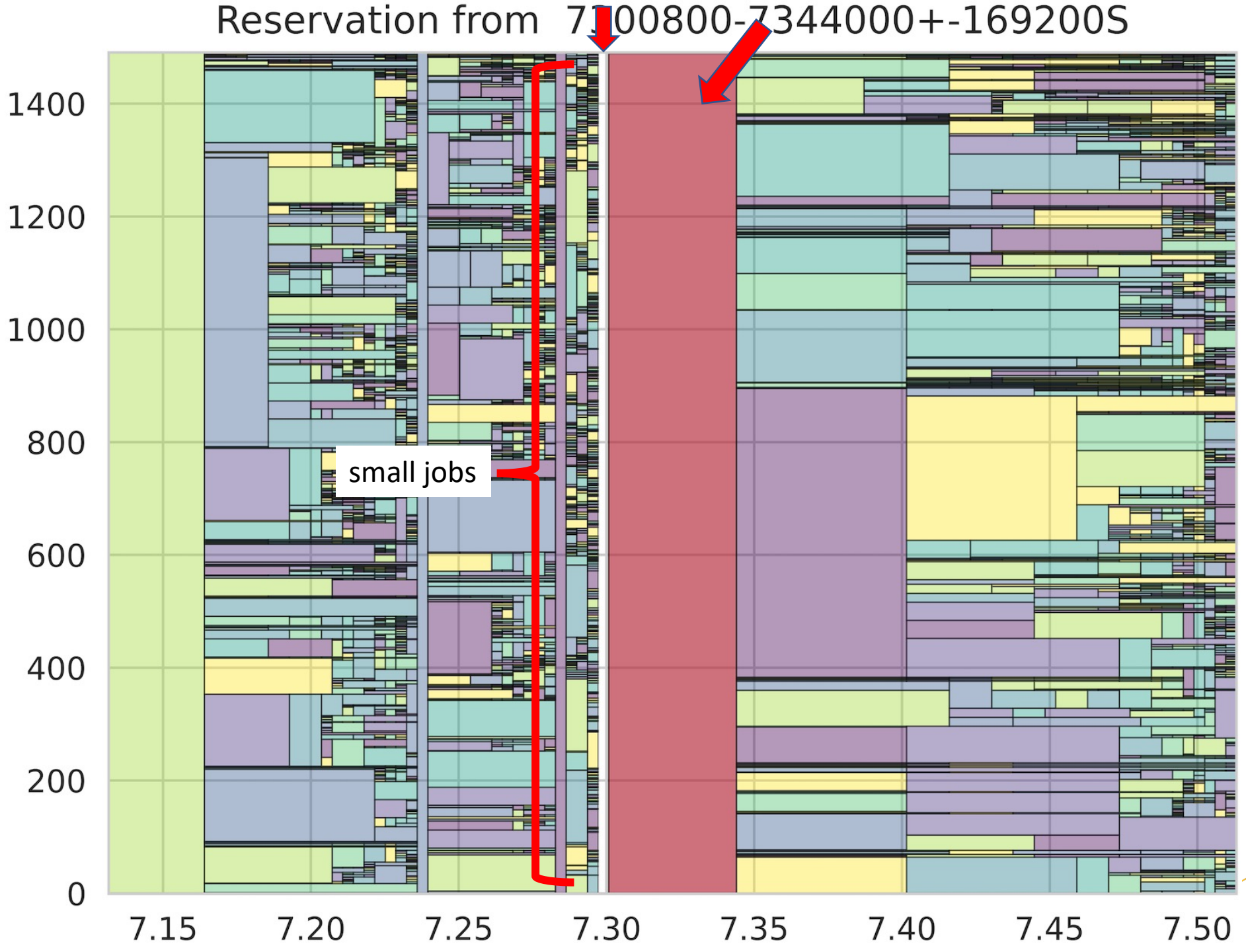
12-hour
Full



COASTAL CAROLINA UNIVERSITY

Los Alamos NATIONAL LABORATORY EST.1943

12-hour
Full



COASTAL CAROLINA UNIVERSITY

Los Alamos NATIONAL LABORATORY EST.1943

12-hour Full


Cluster Utilization by Job Type
Small job utilisation
Long job utilisation
Large job utilisation
Mean utilisation for Small Jobs (276.99)
Mean utilisation for Long Jobs (237.93)
Mean utilisation for Large Jobs (197.37)
Reset event (utilisation == 0)
Small jobs
Long jobs
Large Jobs
Machines
1e6

12-hour Full


Cluster Utilization by Job Type
Small job utilisation
Long job utilisation
Large job utilisation
Mean utilisation for Small Jobs (276.99)
Mean utilisation for Long Jobs (237.93)
Mean utilisation for Large Jobs (197.37)
Reset event (utilisation == 0)
Machines
Small jobs
Long jobs
Large Jobs
7.15
7.20
7.25
7.30
7.35
7.40
7.45
7.50
1e6

12-hour Full

small

12-hour Full

small

long

12-hour Full

small

long

large

12-hour Full

small

long

large

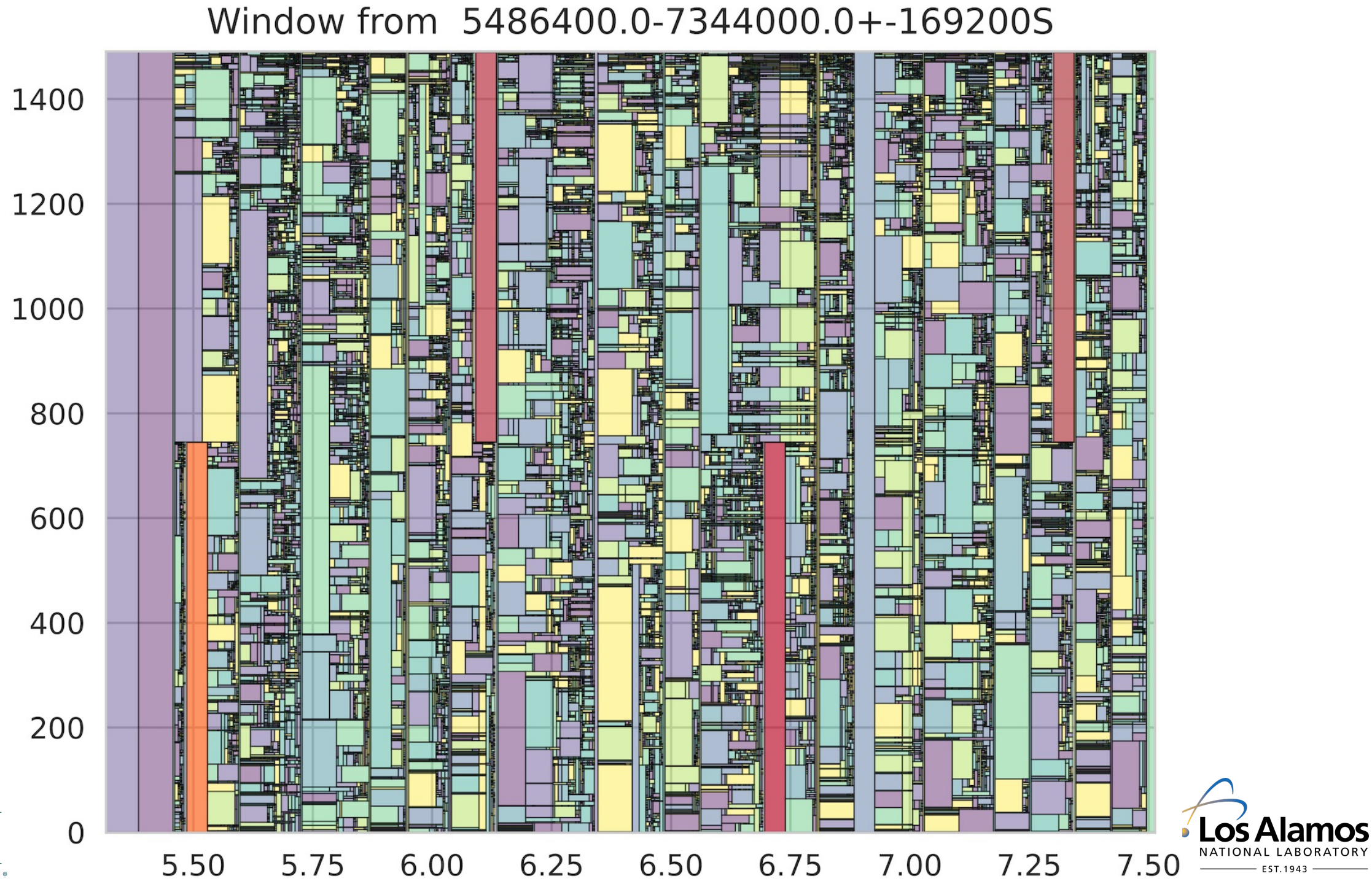
12-hour Full

12-Full

12-Half

12-hour

Half

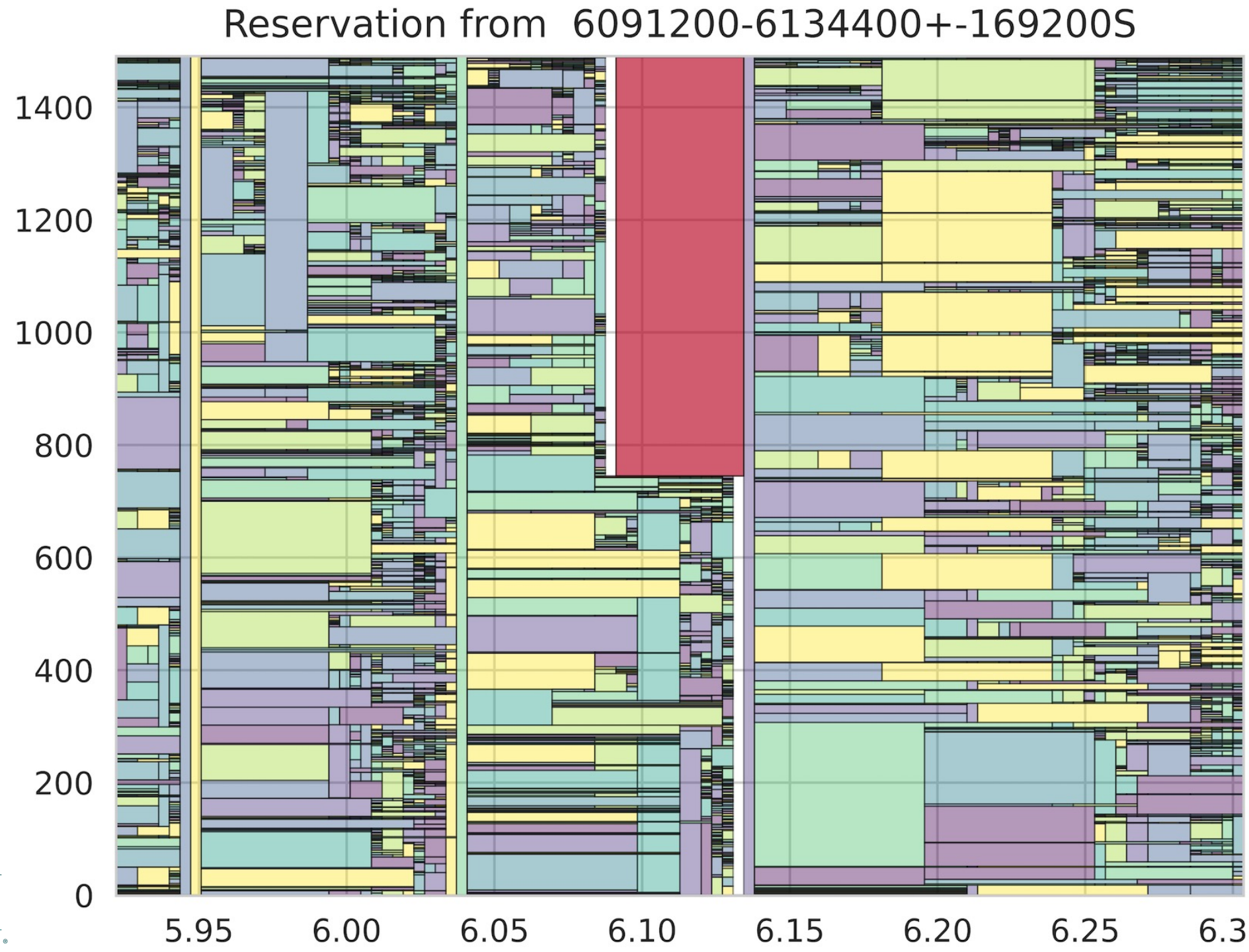


COASTAL CAROLINA UNIVERSITY

Los Alamos NATIONAL LABORATORY EST.1943

12-hour

Half



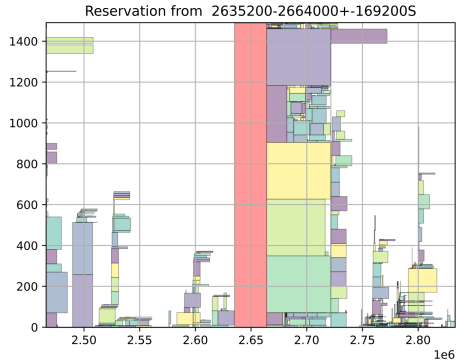
COASTAL CAROLINA UNIVERSITY

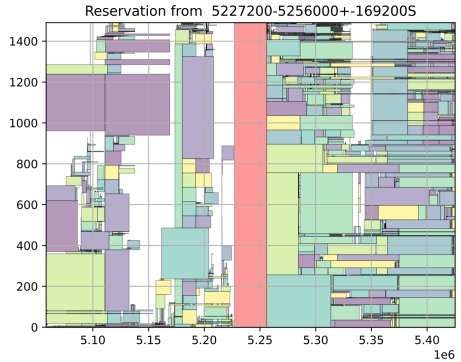
Los Alamos NATIONAL LABORATORY EST. 1943

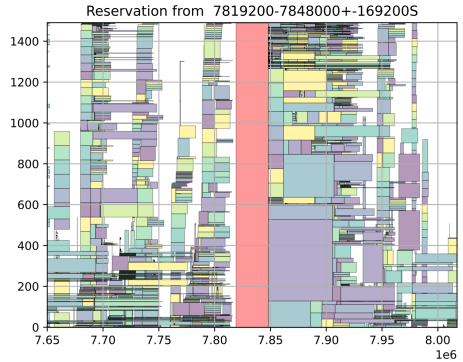
12-hour

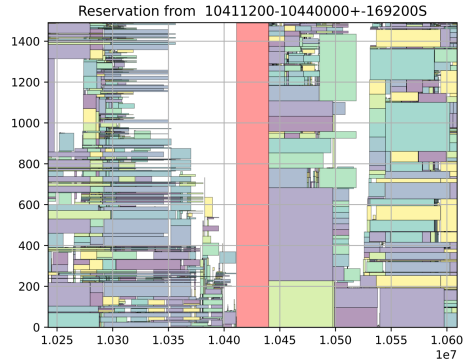
Half

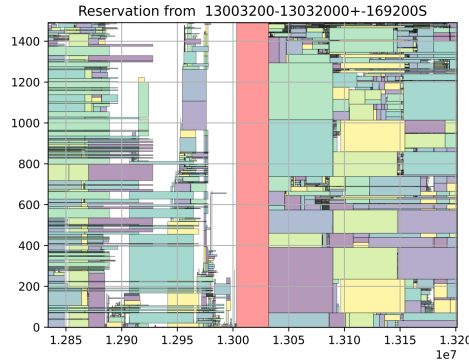

Cluster Utilization by Job Type
Small job utilisation
Long job utilisation
Large job utilisation
Mean utilisation for Small Jobs (347.67)
Mean utilisation for Long Jobs (293.14)
Mean utilisation for Large Jobs (110.50)
Reset event (utilisation == 0)
Machines
Small jobs
Long jobs
Large Jobs
1e6

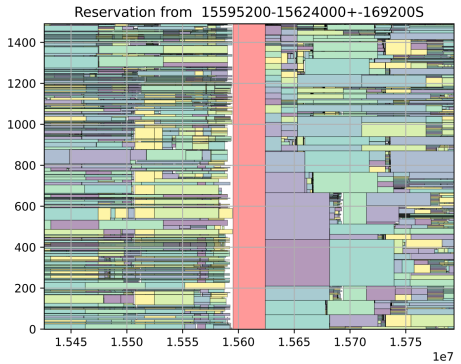

Reservation from 2635200-2664000+-169200S
1e6

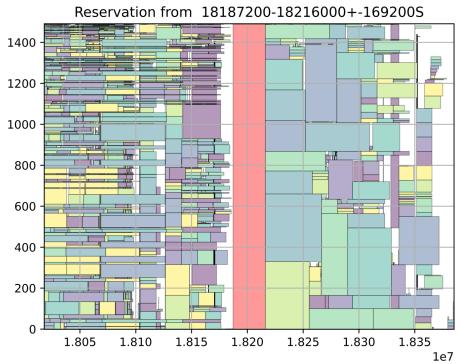

Reservation from 5227200-5256000+-169200S
1e6

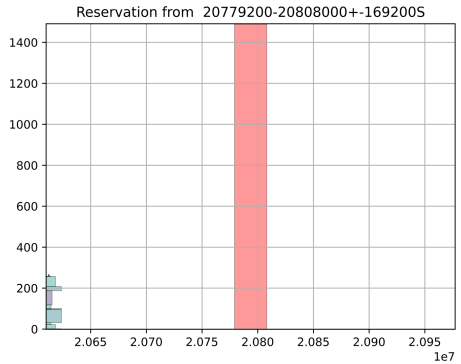

Reservation from 7819200-7848000+-169200S
1e6

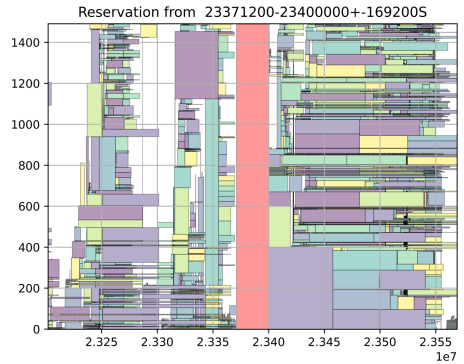

Reservation from 10411200-10440000+-169200S
1e7

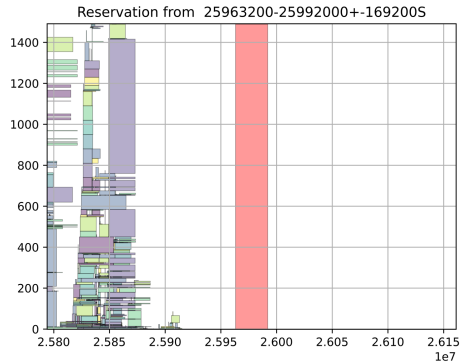

Reservation from 13003200-13032000+-169200S
1e7

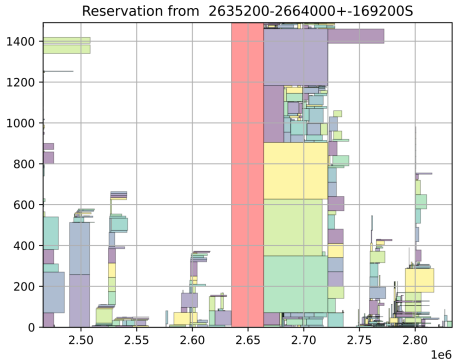

Reservation from 15595200-15624000+-169200S
1e7

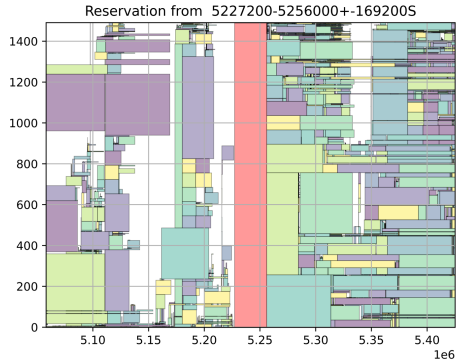

Reservation from 18187200-18216000+-169200S
1e7

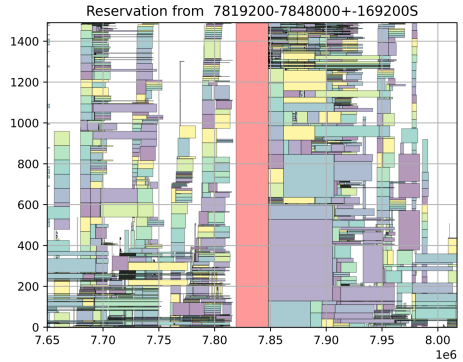

Reservation from 20779200-20808000+-169200S
1e7

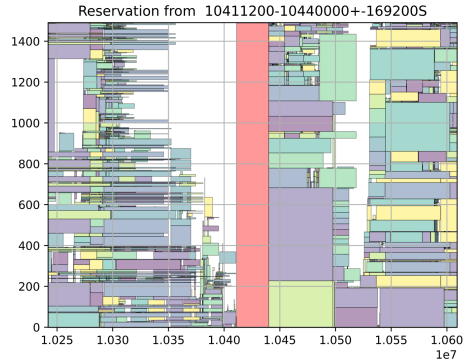

Reservation from 23371200-23400000+-169200S
1e7

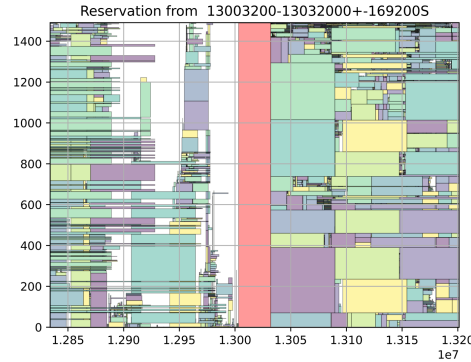

Reservation from 25963200-25992000+-169200S
1e7

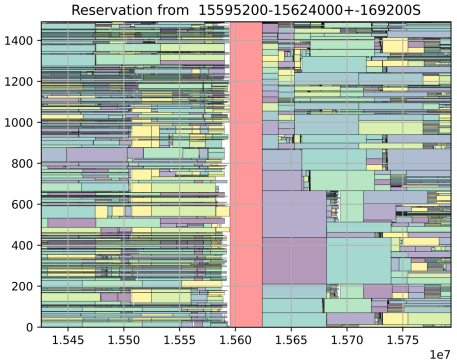


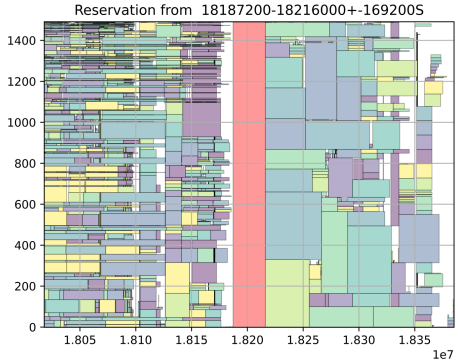


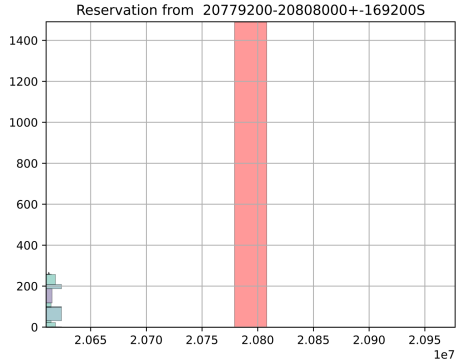


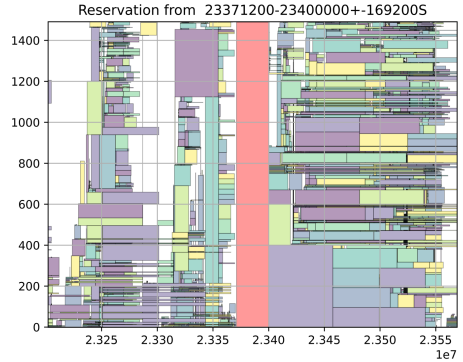


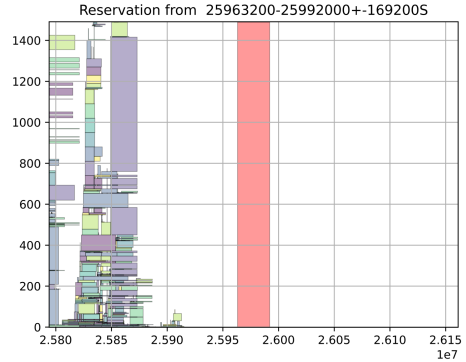


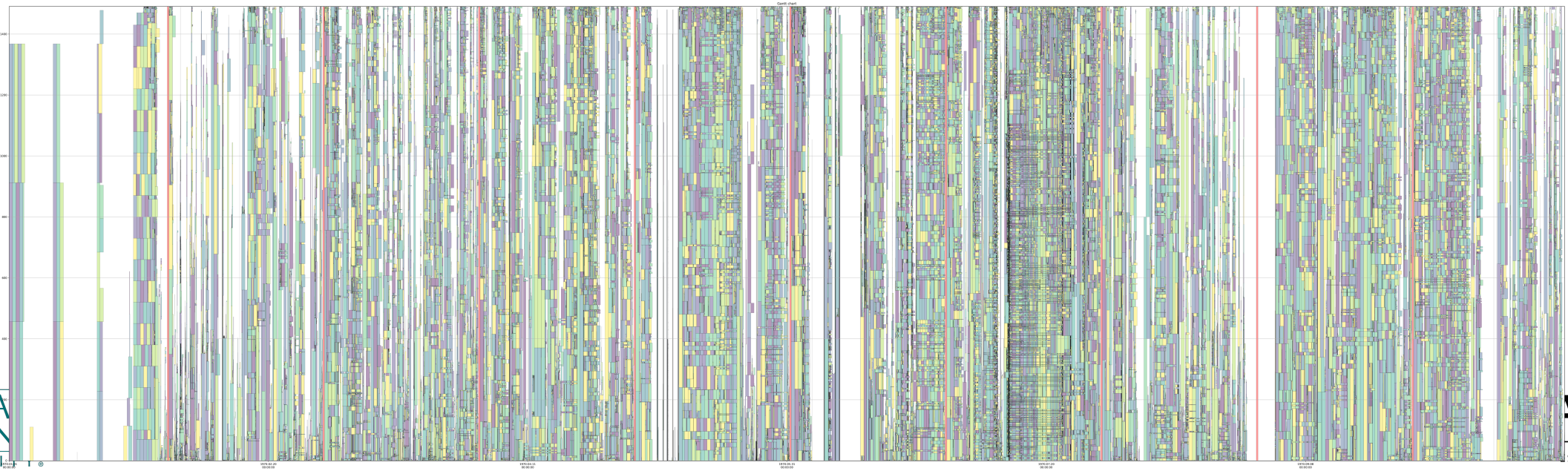












COASTAL CAROLINA UNIVERSITY

Alamos LABORATORY
EST.1943