JOB MONITORING TOOL FOR **RESOURCE UTILIZATION AWARENESS** AND **OBSERVABILITY** OF **HPC ADMINS & USERS**

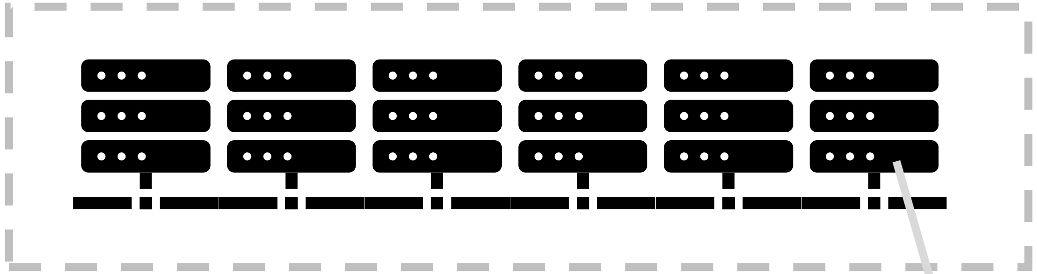
Qixing Xue

Advised by Fabricio Murai and Ermal Toto

Part of Interactive Qualifying Project (IQP)
Improving Computing Efficiency and Reducing Carbon Footprint for Turing Cluster

Icons and decorative images provided by Microsoft 365 creative content library

High Performance Computing (HPC)



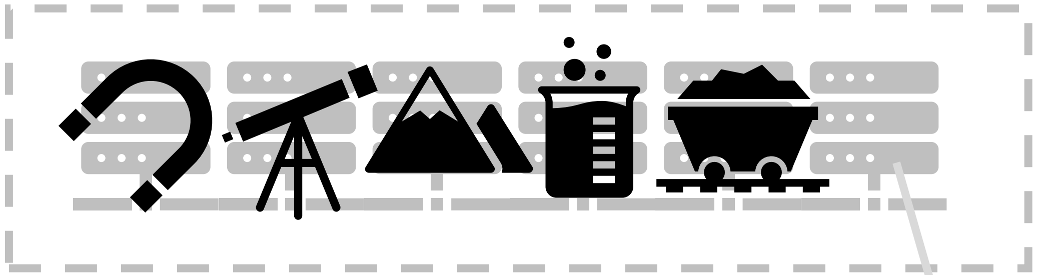
Cluster

- Hundreds & Thousands of CPU cores
- >960TB of storage backed up hourly
- >100Gbps Infiniband connection

Per Node

- Dozens to hundreds gigabytes of RAM
- Hardware accelerators
    - ~20 trillion floating point operations / sec
    - e.g. compute card NVIDIA A100

Attracting Usages in…



Physics

Astronomy

Geology

Chemistry

Material Science

Cluster

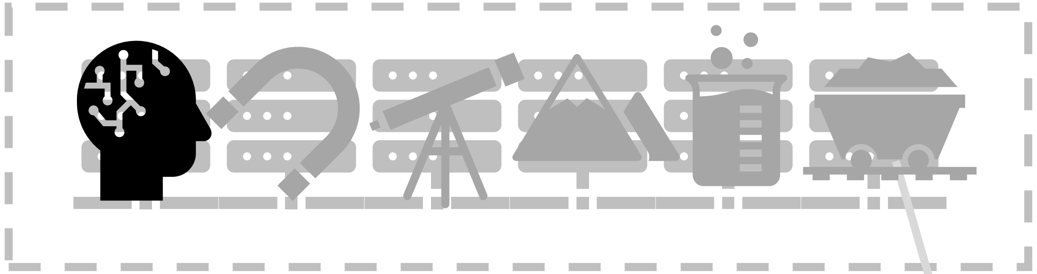
- Hundreds & Thousands of CPU cores
- >960TB of storage backed up hourly
- >100Gbps Infiniband connection

Per Node

- Dozens to hundreds gigabytes of RAM
- Hardware accelerators
  - ~20 trillion floating point operations / sec
  - e.g. compute card NVIDIA A100

* Icons are stereotypes of the subjects and does not represent actual research topics.

Well… Also this one



Machine learning

- NVIDIA V100 Compute card released 2 years before customer grade GPU RTX 2060 super
- V100 can be more than 30x faster than 2060[1] (in FP64 FLOP/s)

Cluster

- Hundreds & Thousands of CPU cores
- >960TB of storage backed up hourly
- >100Gbps Infiniband connection

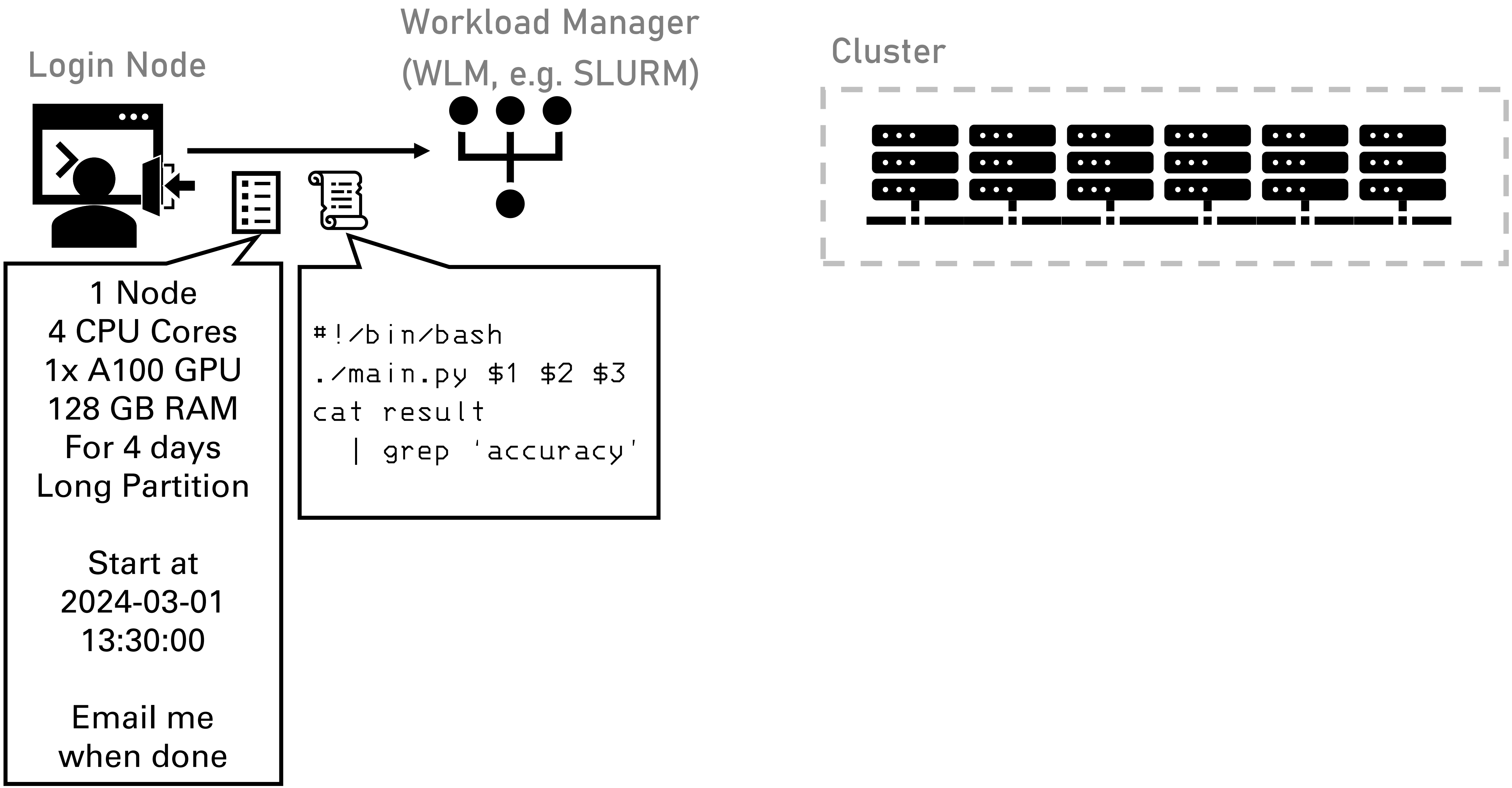
Per Node

- Dozens to hundreds gigabytes of RAM
- Hardware accelerators
  - ~20 trillion floating point operations / sec
  - e.g. compute card NVIDIA A100

* Icons are stereotypes of the subjects and does not represent actual research topics.

[1] **Svedin et. al., Benchmarking the Nvidia GPU Lineage: From Early K80 to Modern A100 with Asynchronous Memory Transfers (2021)**

Which computer in the cluster to use?



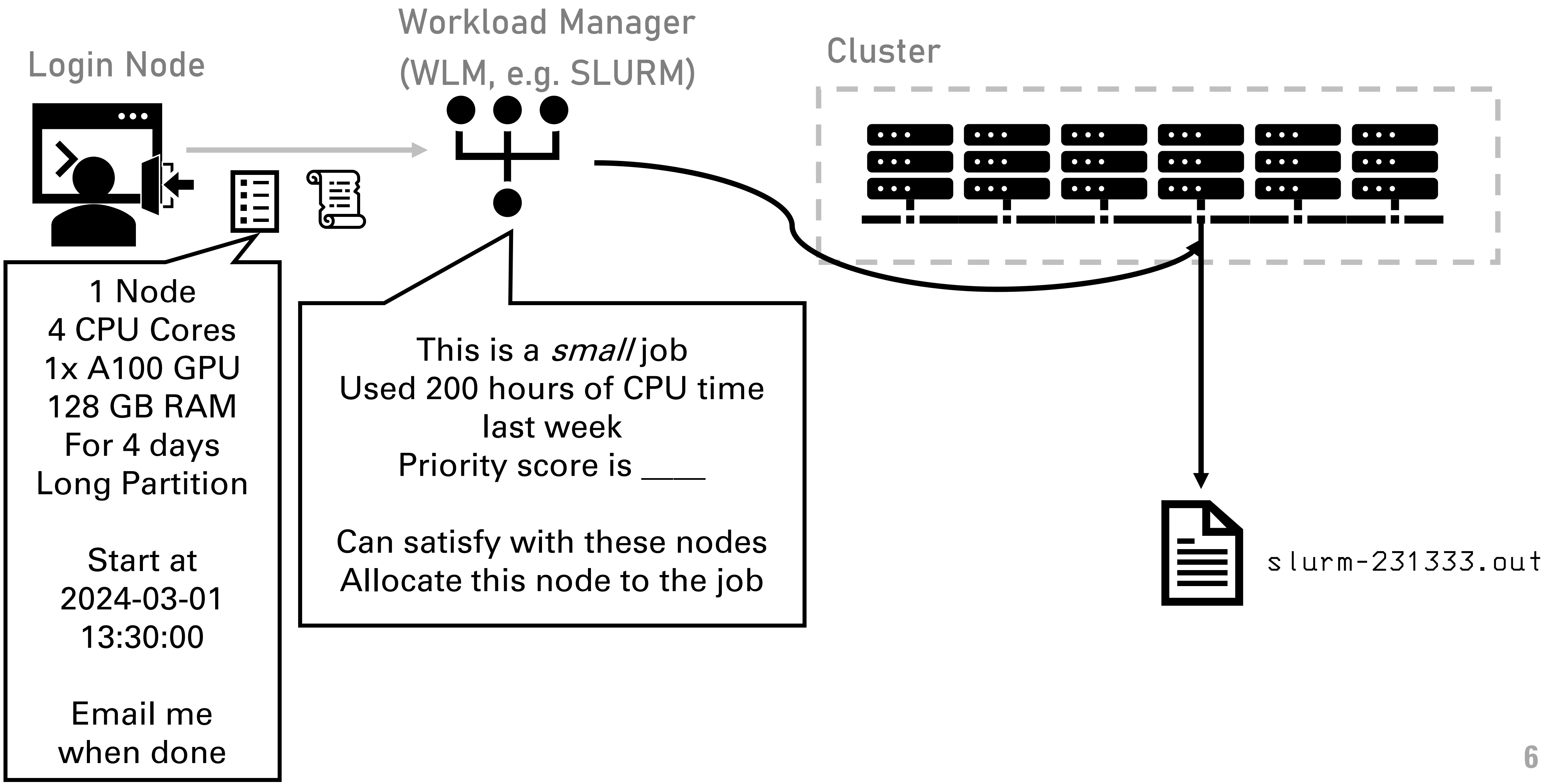
1 Node
4 CPU Cores
1x A100 GPU
128 GB RAM
For 4 days
Long Partition

Start at
2024-03-01
13:30:00

Email me
when done

```
#!/bin/bash
./main.py $1 $2 $3
cat result
  | grep 'accuracy'
```

Which computer in the cluster to use?



1 Node
4 CPU Cores
1x A100 GPU
128 GB RAM
For 4 days
Long Partition

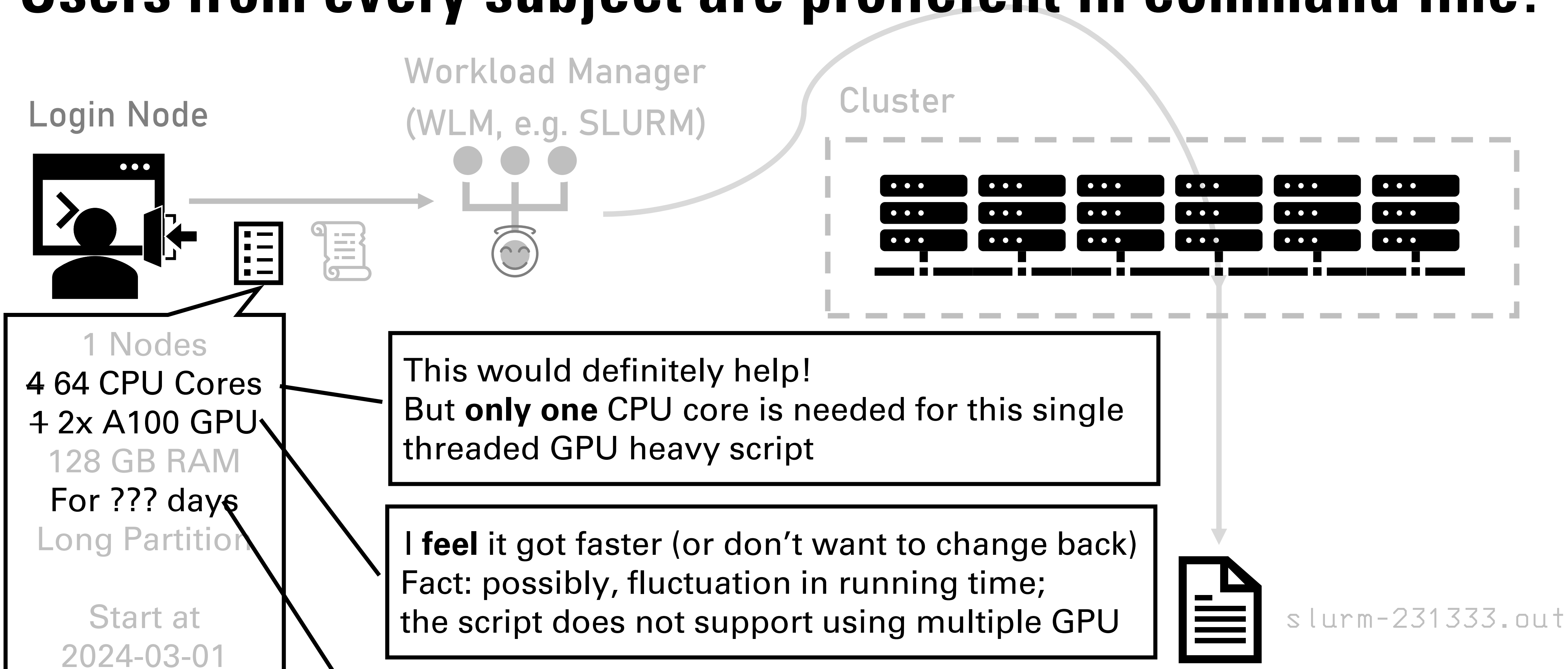
Start at
2024-03-01
13:30:00

Email me
when done

This is a *small* job
Used 200 hours of CPU time
last week
Priority score is ____

Can satisfy with these nodes
Allocate this node to the job

Users from every subject are proficient in command line!



1 Nodes
~~4~~ 64 CPU Cores
~~1~~ 2x A100 GPU
128 GB RAM
For ??? days
Long Partition

Start at
2024-03-01
13:30:00

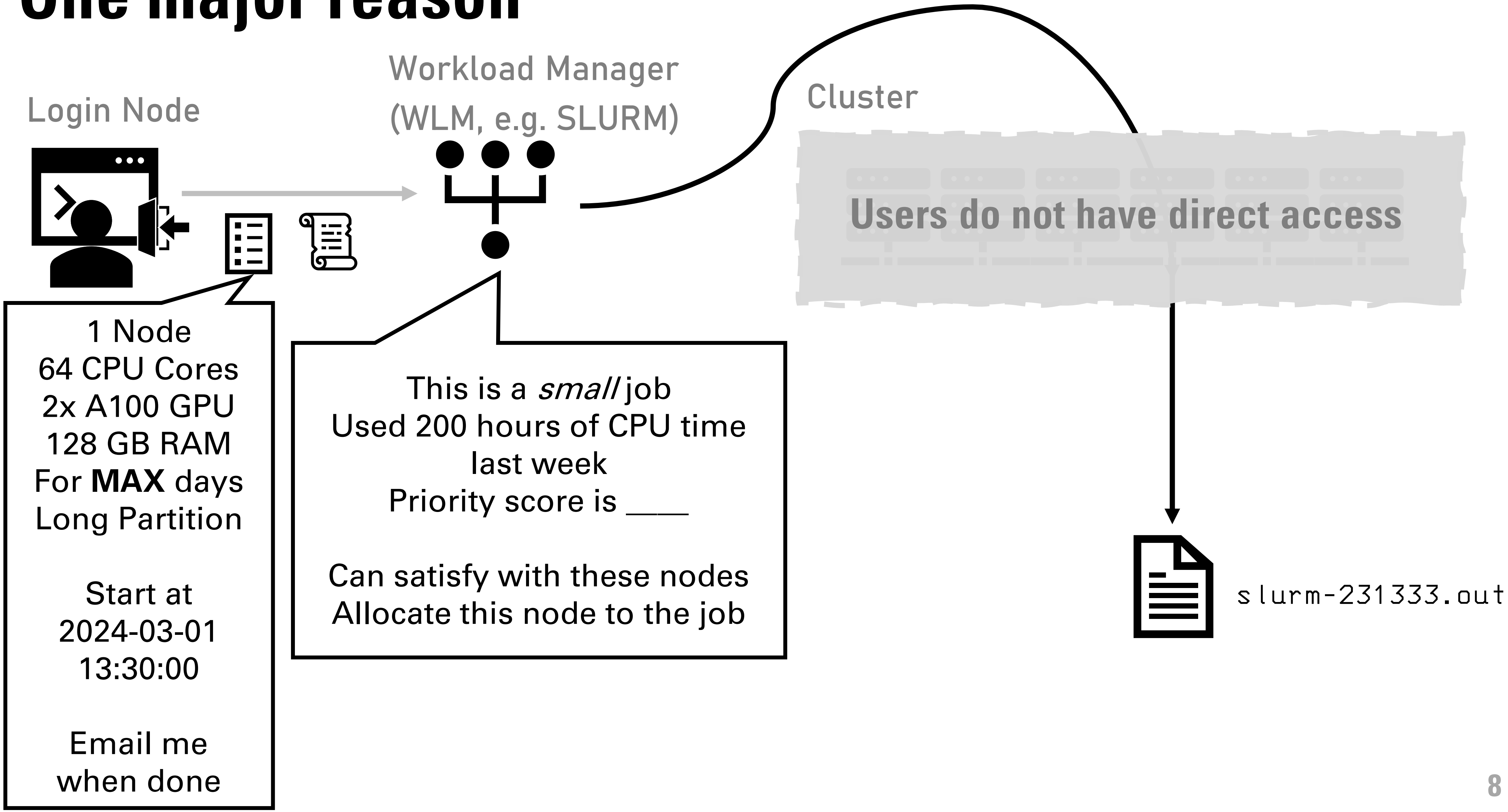
Email me
when done

This would definitely help!
But **only one** CPU core is needed for this single threaded GPU heavy script

I **feel** it got faster (or don’t want to change back)
Fact: possibly, fluctuation in running time;
the script does not support using multiple GPU

How long **should** one run take?
I don’t have 128 GB locally to firstly know…
(have no reference to say if it is actually running slow)

One major reason



Login Node

Workload Manager (WLM, e.g. SLURM)

Cluster

Users do not have direct access

1 Node
64 CPU Cores
2x A100 GPU
128 GB RAM
For **MAX** days
Long Partition

Start at
2024-03-01
13:30:00

Email me
when done

This is a *small* job
Used 200 hours of CPU time last week
Priority score is ____

Can satisfy with these nodes
Allocate this node to the job

`slurm-231333.out`

We have idle hardware! Underusing is fine…

It takes electricity…

Using **one more** CPU core has **minor impact** on heat generation

But **cooling** has to be running **constantly**

Takes up **30%** of power consumption[1]

Running longer inefficiently lowers compute efficiency (FLOP/Watt)

Meanwhile, machine learning…

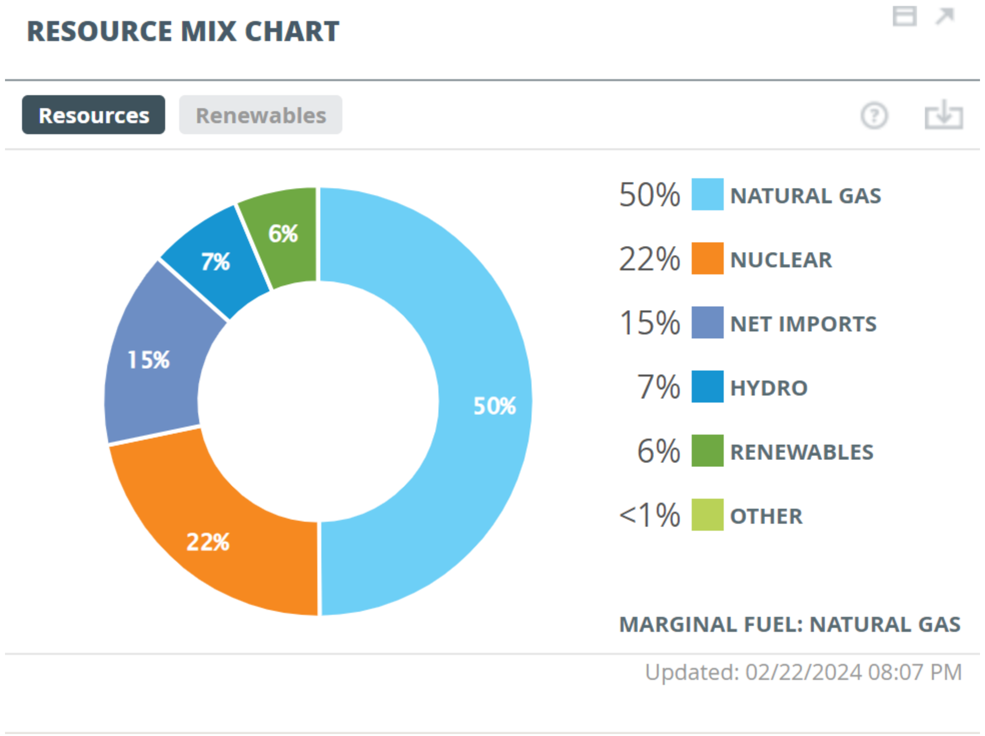
Further drives up power consumption

- Training GPT-3 consumes 1,287 MWh of Power
- Hyperparameter tuning is the major cause
  - Adjusting model structure for better accuracy
  - Experimental and frequently discards nonideal results
  - Can involve trying hundreds of parameters with some of them being float
- Use more iterations to buy margin decreasing accuracy

[1] Zhang et. al., A survey on data center cooling systems: Technology, power consumption modeling and control strategy optimization (2023)

We have powerful electricity plant!

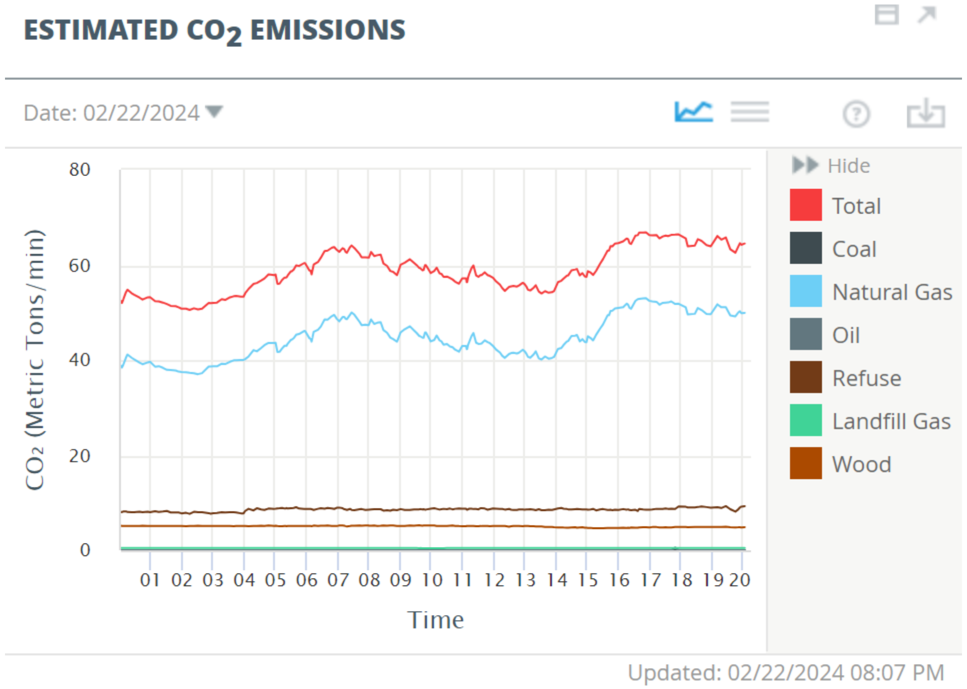
We are also using significant proportion of fossil fuel…


RESOURCE MIX CHART

Resources | Renewables

MARGINAL FUEL: NATURAL GAS

Updated: 02/22/2024 08:07 PM


ESTIMATED CO_2 EMISSIONS

Date: 02/22/2024

Updated: 02/22/2024 08:07 PM

Source: ISO New England, visited Feb 22, 2024 08:07 PM

We have powerful electricity plant!

We are also using significant proportion of fossil fuel…

Training natural language processing (NLP) model [1,*]

- **39 lbs** of CO_2 per training
- **78,468 lbs** with accounting hyperparameter tuning
  - Equivalent to double of regular American life

Data centers globally produce **100 megatonnes of CO_2**

Sustainability Risks…

Will continue to grow without intervention[2]

- For increasingly more compute needs

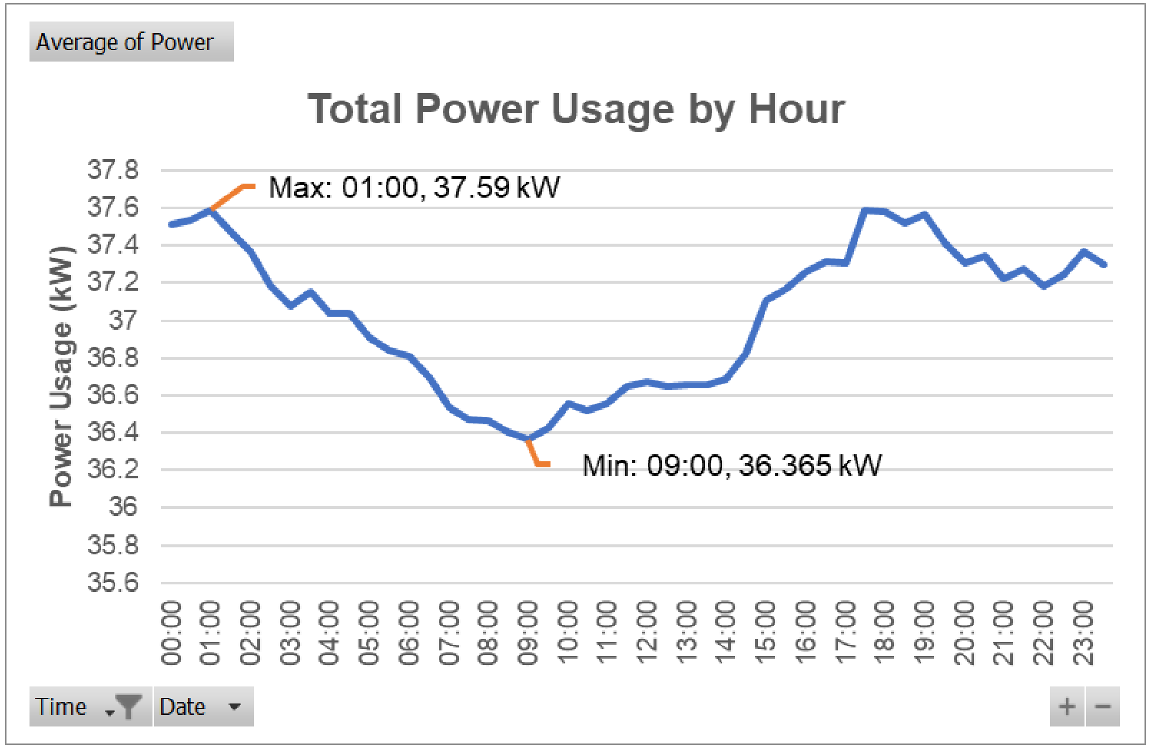
*** Using a model for semantic role labelling (shallow semantic parsing)**

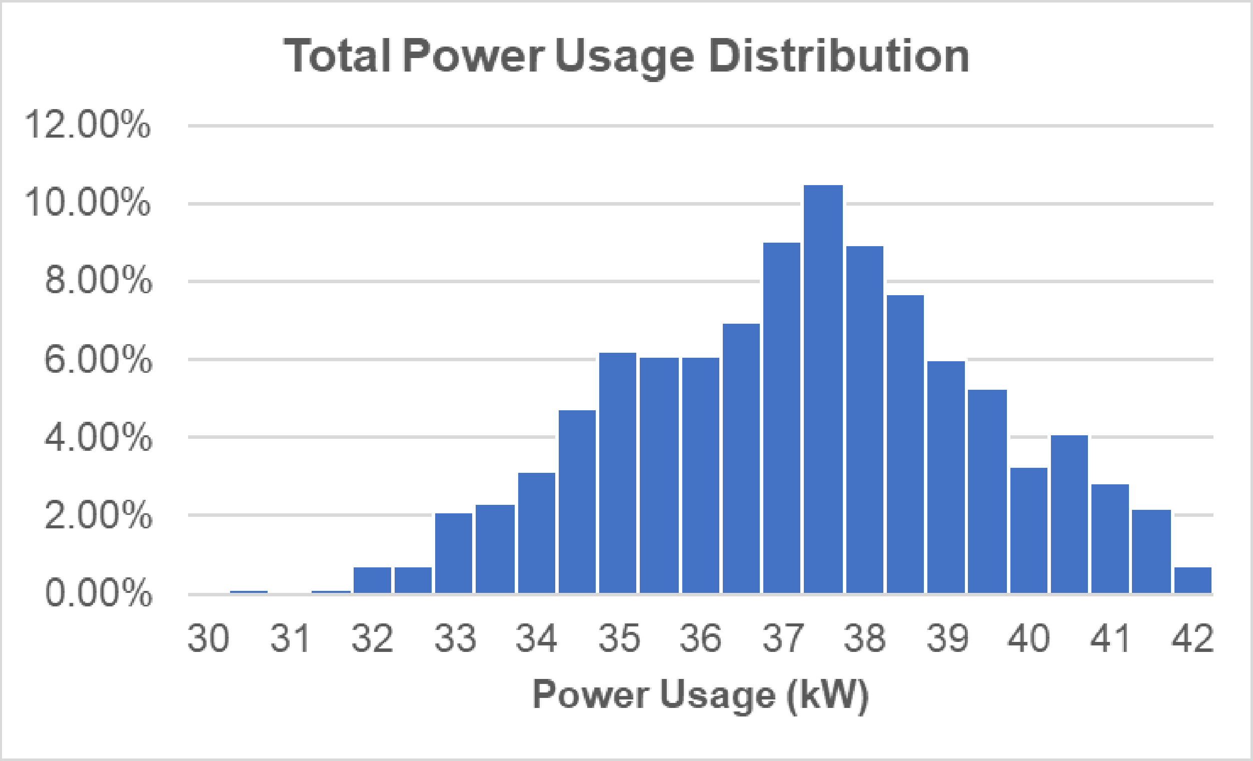
[1] Strubell et. al., Energy and Policy Considerations for Modern Deep Learning Research (2020)

[2] Lannelongue et. al., Green Algorithms: Quantifying the Carbon Footprint of Computation (2021)

Ok that is such a big picture…

For Turing cluster here…





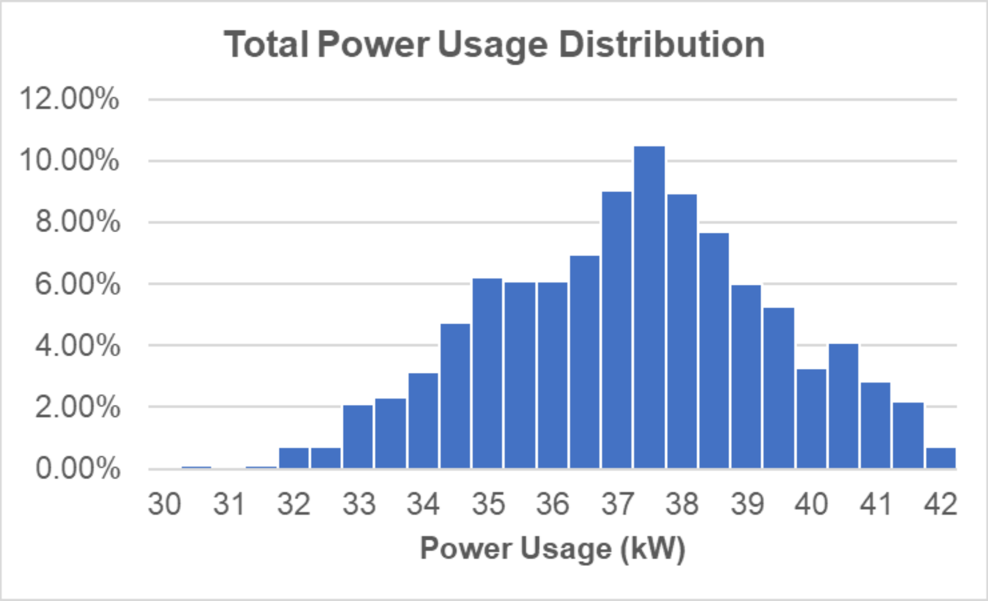
Statistic period Jan 18, 2024 to Feb 7, 2024

Let's do some math

95% Confidence Interval…
[36.89, 37.16] kW

Consumption each year…
[323.134, 325.564] MWh

Carbon Emission
2022 EPA data: natural gas 0.000485 tCO_2/kWh
[156.72,157.90] tCO_2 / year


Total Power Usage Distribution

Adjust for Cooling: divide by proportion of IT equipment (1-30%)
[223.89, 225.57] tCO_2 / year

~200 lifecycles of EV battery (material to product to recycle)[1]

[1] Kumar et. al., Life Cycle Assessment Based Environmental Footprint of a Battery Recycling Process (2022), p.p. 119-120
"216.2 kg CO2/kWh in the production phase, 94.2 kg CO2 eq/kWh in the use phase, and −17.18 kg CO2 eq/kWh in the recycling phase"

WHAT CAN WE DO?

Do less work to save the earth?

Literature Says…

Administrators

Need a monitoring tool [1] to…

Identify changes in usage pattern
may diverge from initial assumption

Locate improvement goal

Users

Need to…

Be educated on job scheduler usage
so to improve energy efficiency

[1] Allcock et. al., Challenges of HPC monitoring (2011)

Existing Tools…

Frequently concerns hardware health
rather than alerting problematic jobs

Rely on external data sources

Requires software deployment

e.g.

XDMoD
METRICS ON DEMAND

Widely used

Does not collect data by itself

Requires MySQL or MariaDB to work

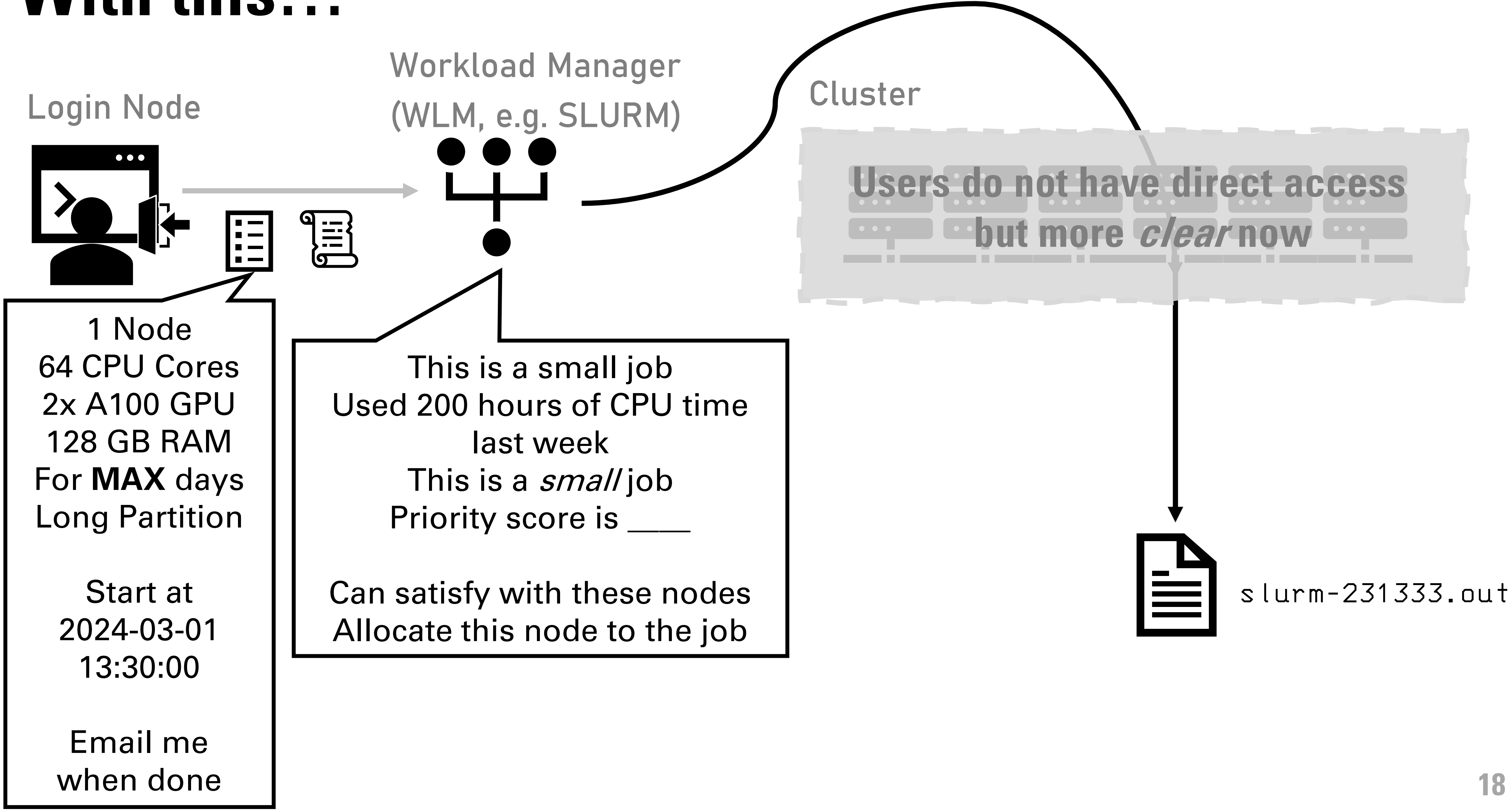
So the tool aims to

Increase observability of job steps’ resource utilization

Raise awareness of computing resource underutilization

Suggest resolutions for problems identified

With this…



Login Node

Workload Manager
(WLM, e.g. SLURM)

Cluster

Users do not have direct access
but more *clear* now

1 Node
64 CPU Cores
2x A100 GPU
128 GB RAM
For **MAX** days
Long Partition

Start at
2024-03-01
13:30:00

Email me
when done

This is a small job
Used 200 hours of CPU time
last week
This is a *small* job
Priority score is ____

Can satisfy with these nodes
Allocate this node to the job

`slurm-231333.out`

Therefore, the tool should

Collect statistics of processes under job allocations **and** related GPU driver readings

Generate evidence-based and actionable reports

Allow users and admins to **use independently**

Be extensible and tunable

This means we need…

A scraper to fetch **data**

A place to store it
Surely database – is this even a question?

Collect statistics of processes under **job allocations and** related GPU driver readings

Somehow spawn scrapers onto compute nodes
(by user or do it on our own)

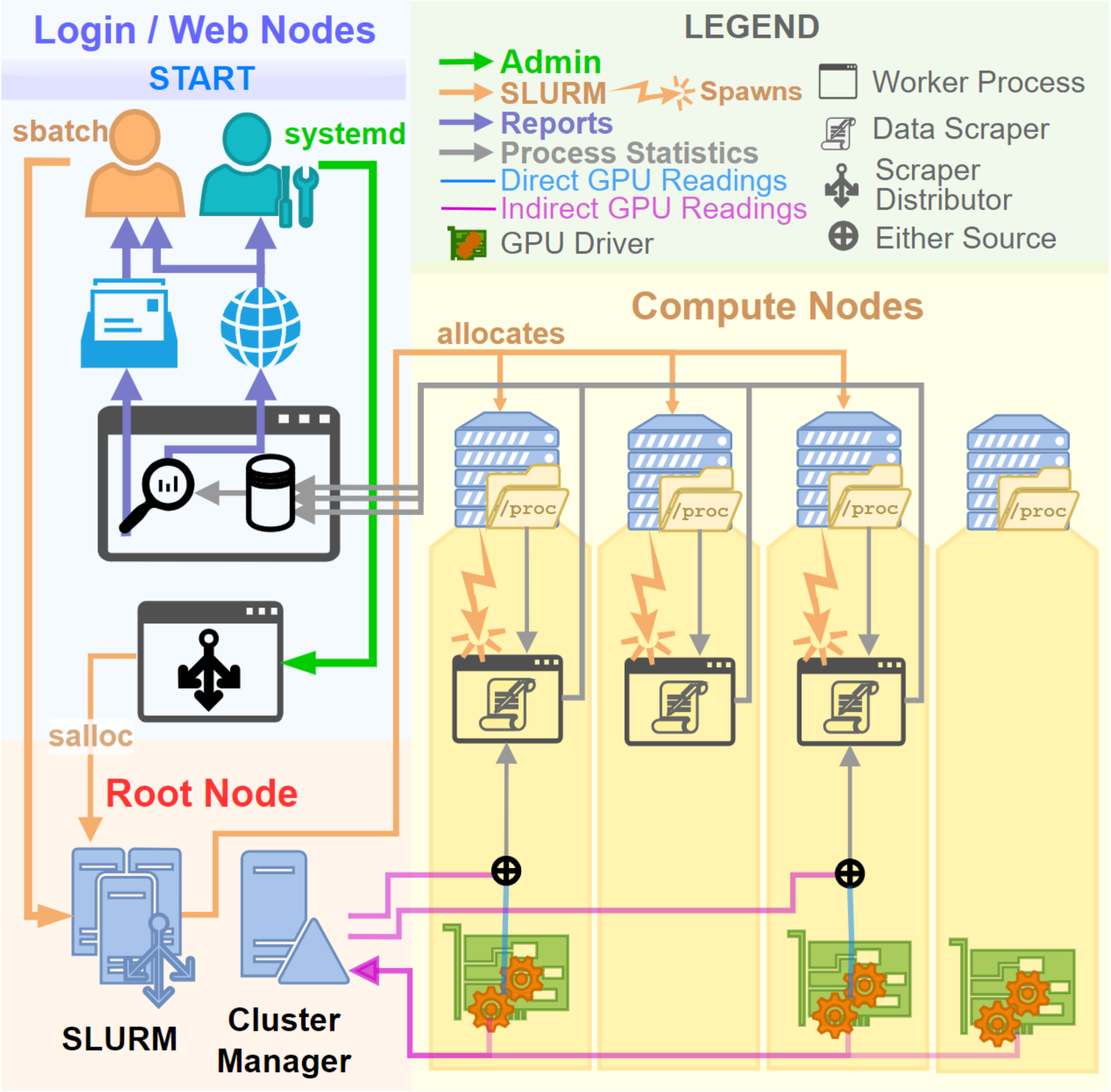
Generate evidence-based and actionable reports

In what format? Pushed to or pulled by users?
How can we do more with data in the reports?

Allow users and admins to **use independently**

Be extensible and tunable

HOW TO APPROACH THIS…



IMPLEMENTATION

Isn’t database an easy decision?

Eliminate need of software deployment
SQLite as DBMS
Reduces barrier of being used by regular users

However, for network file system (NFS)
SQLite can only lock by file page for concurrency

A lightweight custom database server serializes database operations
Also imports data from SLURM and **invokes analyzer**

Why not use existing data providers?

These data providers **are not designed** for collecting data for **this purpose**

- **Low granularity** for describing job characteristics
  - E.g. Bright Cluster Manager refreshes data every **2 minutes** and are mainly consists of **hardware status info**
- High penalty for RPC calls
  - E.g. **Multiple hierarchy** of SLURM makes an RPC take **seconds** to be responded

Collect statistics from `/proc` and `/sys/fs/cgroup`
Also collects **GPU readings**
Send back to server daemon in unified units

How scrapers ever get to run?

Users modify job submission script
to run watcher and scraper in background

OR… If one would like to **sample the cluster**

Why not record everything?
This inflates the database way too fast…
Regardless its frequency and intensity are **tunable**

How to prevent affecting other jobs?
Simply ask SLURM to fairly schedule us a core!

Sample in what order?
Prefer amount or fairness?

Scraper Distributor

Key idea

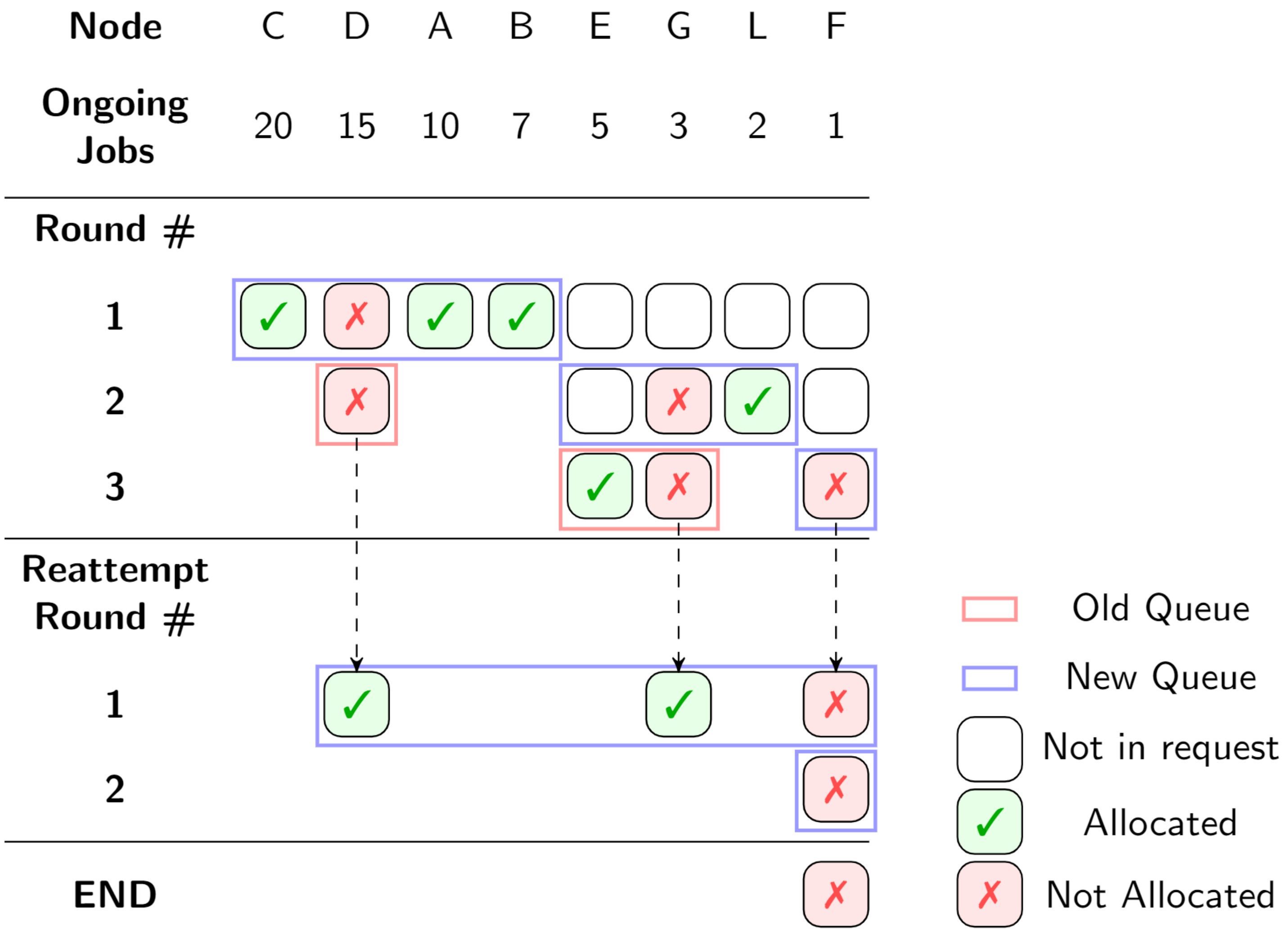
balanced sampling

Prefer nodes with more ongoing jobs

- know about more jobs

Try every available nodes at least twice

- avoid node differences



Node	C	D	A	B	E	G	L	F
Ongoing Jobs	20	15	10	7	5	3	2	1
Round #								
1	✓	✗	✓	✓	□	□	□	□
2		✗			□	✗	✓	□
3					✓	✗		✗
Reattempt Round #								
1		✓				✓		✗
2								✗
END								✗

▭	Old Queue
▭	New Queue
□	Not in request
✓	Allocated
✗	Not Allocated

Now we got the data! What to do?

Identify active users

Build time series showing, e.g.

- Changes in resource usage
- Intensity of kernel activities

Perform queries on the data

- Derive metric values and flag problems

From that on…How to report findings?

In HTML format for users to…

- receive via email
- view on-demand through website

In JSON format for use by integrations

- e.g. pivot-table like views

Design goals…

- Include both summary and on-click details
- Identify problems and suggest actionable solutions

Why compatibility of restrictive HTML email?

Web servers are **hard** for **regular cluster users** to set up

- Installation, authentication, access control (firewall)...

Leverage existing setup for SLURM to send notifications

Push content to users and **actively alert underutilizations**

- Important when being deployed by administrators
- They can also respond simply by clicking reply button
  - Context is naturally included in reply email

Why not send as **attachments**?

- Users feel **insecure** about opening them at the first place
- **Discourages** users from reading and seeing the content

Can users get anything for effort fixing these?

Less time spent waiting…

- **in queue**, for reduced amount of resource required
- **for results**, for fixing underuses of allocated resources

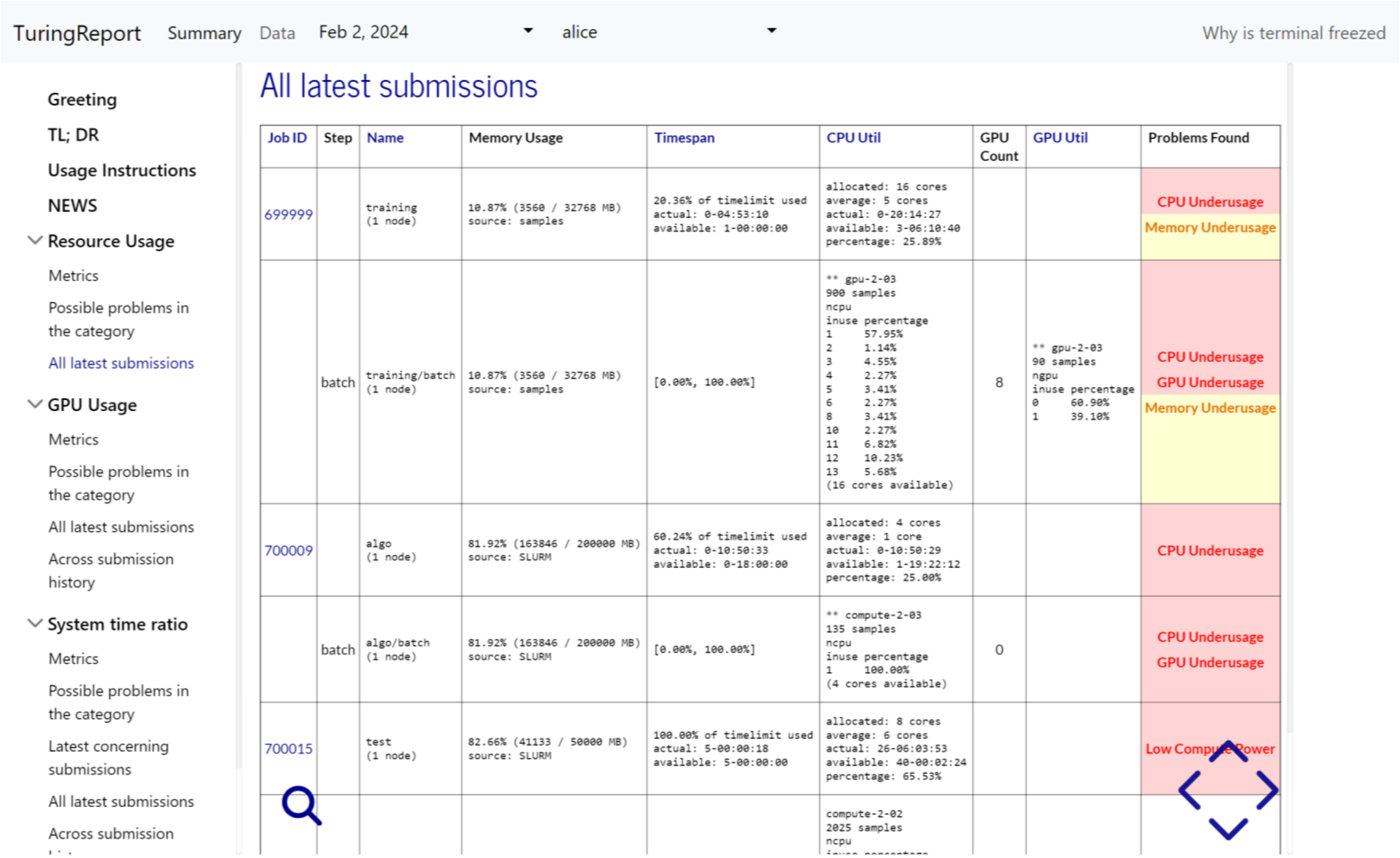
Resulting from making sure users…

- get what they need
- use what they get

WHAT THE USER WILL SEE

Reporting website

- **Same content** but **far more interactive** than HTML emails
- Allow **popups** and **page updates**
- Users can check these **on demand**
- **HTML emails** are still a **crucial piece**

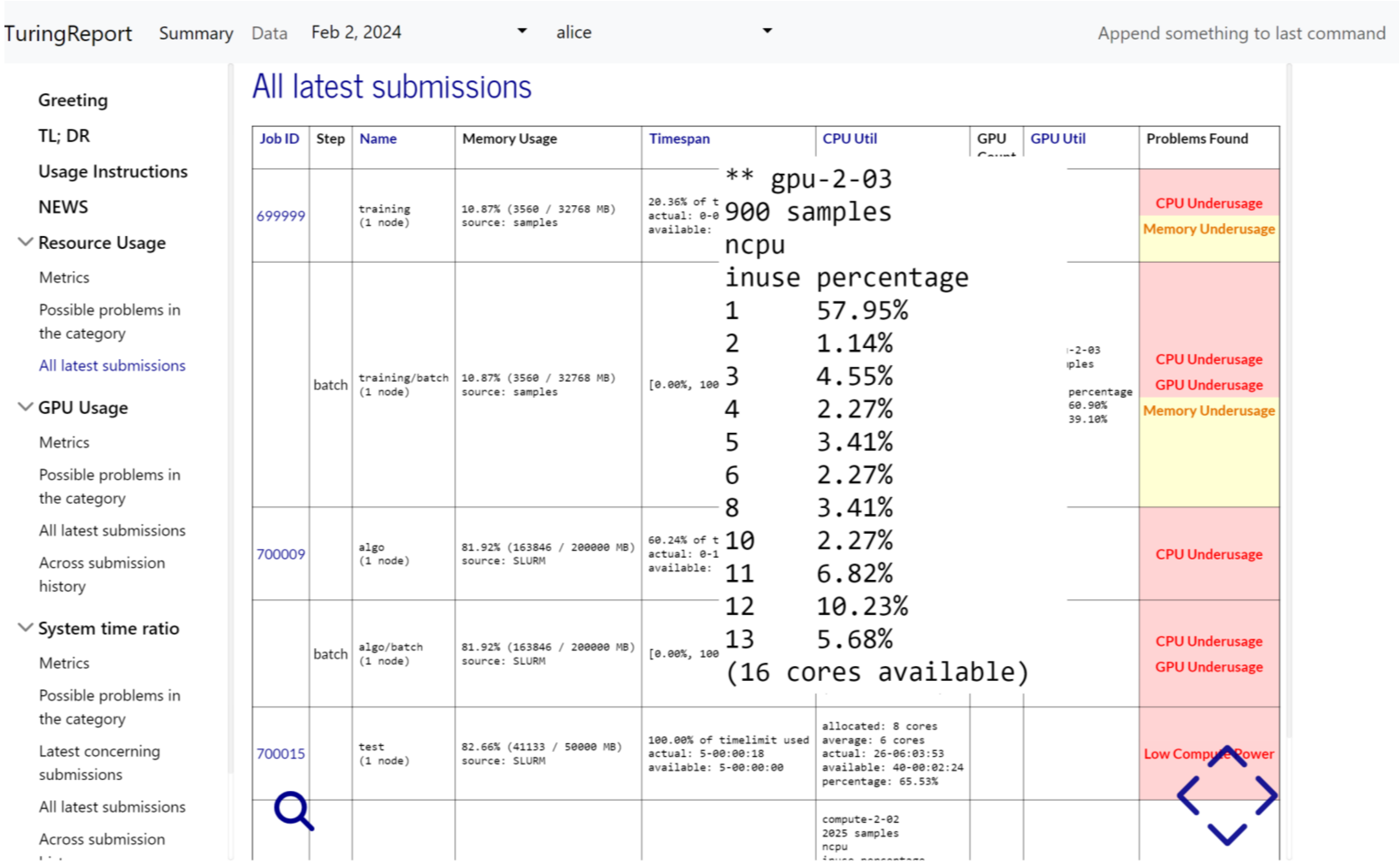


All latest submissions

Job ID	Step	Name	Memory Usage	Timespan	CPU Util	GPU Count	GPU Util	Problems Found
699999		training (1 node)	10.87% (3560 / 32768 MB) source: samples	20.36% of timelimit used actual: 0-04:53:10 available: 1-00:00:00	allocated: 16 cores average: 5 cores actual: 0-20:14:27 available: 3-06:10:40 percentage: 25.89%			CPU Underusage Memory Underusage
	batch	training/batch (1 node)	10.87% (3560 / 32768 MB) source: samples	[0.00%, 100.00%]	** gpu-2-03 900 samples ncpu inuse percentage 1 57.95% 2 1.14% 3 4.55% 4 2.27% 5 3.41% 6 2.27% 8 3.41% 10 2.27% 11 6.82% 12 10.23% 13 5.68% (16 cores available)	8	** gpu-2-03 90 samples ngpu inuse percentage 0 60.90% 1 39.10%	CPU Underusage GPU Underusage Memory Underusage
700009		algo (1 node)	81.92% (163846 / 200000 MB) source: SLURM	60.24% of timelimit used actual: 0-10:50:33 available: 0-18:00:00	allocated: 4 cores average: 1 core actual: 0-10:50:29 available: 1-19:22:12 percentage: 25.00%			CPU Underusage
	batch	algo/batch (1 node)	81.92% (163846 / 200000 MB) source: SLURM	[0.00%, 100.00%]	** compute-2-03 135 samples ncpu inuse percentage 1 100.00% (4 cores available)	0		CPU Underusage GPU Underusage
700015		test (1 node)	82.66% (41133 / 50000 MB) source: SLURM	100.00% of timelimit used actual: 5-00:00:18 available: 5-00:00:00	allocated: 8 cores average: 6 cores actual: 26-06:03:53 available: 40-00:02:24 percentage: 65.53%			Low Compute Power
					compute-2-02 2025 samples ncpu			

Reporting website

- **Same content** but **far more interactive** than HTML emails
- Allow **popups** and **page updates**
- Users can check these **on demand**
- **HTML emails** are still a **crucial piece**



Greeting
TL; DR
Usage Instructions
NEWS
Resource Usage
Metrics
Possible problems in the category
All latest submissions
GPU Usage
Metrics
Possible problems in the category
All latest submissions
Across submission history
System time ratio
Metrics
Possible problems in the category
Latest concerning submissions
All latest submissions
Across submission

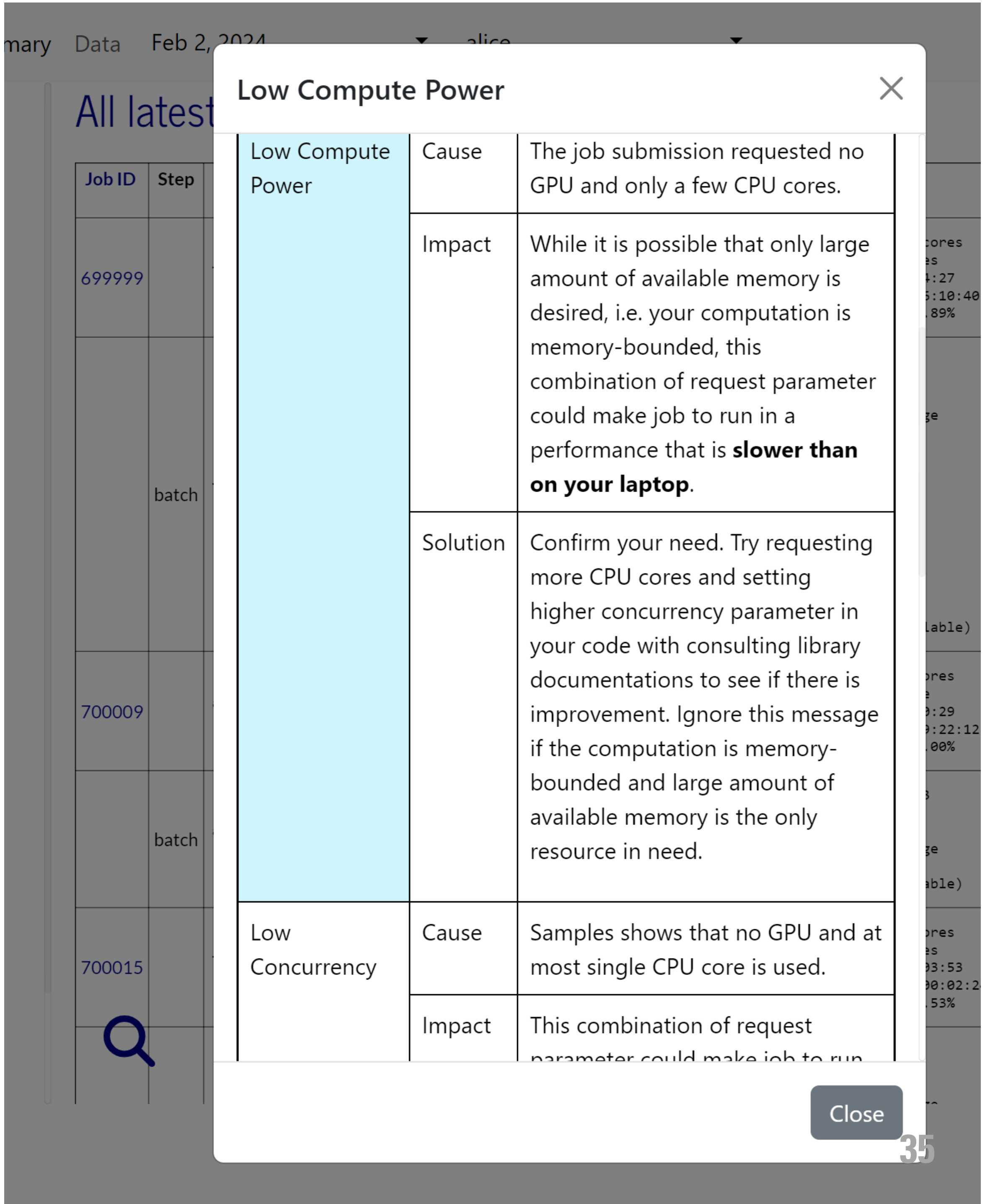
All latest submissions

Job ID	Step	Name	Memory Usage	Timespan	CPU Util	GPU Count	GPU Util	Problems Found
699999		training (1 node)	10.87% (3560 / 32768 MB) source: samples	20.36% of t actual: 0-0 available:				CPU Underusage Memory Underusage
	batch	training/batch (1 node)	10.87% (3560 / 32768 MB) source: samples	[0.00%, 100			-2-03 ples percentage 60.90% 39.10%	CPU Underusage GPU Underusage Memory Underusage
700009		algo (1 node)	81.92% (163846 / 200000 MB) source: SLURM	60.24% of t actual: 0-1 available:				CPU Underusage
	batch	algo/batch (1 node)	81.92% (163846 / 200000 MB) source: SLURM	[0.00%, 100				CPU Underusage GPU Underusage
700015		test (1 node)	82.66% (41133 / 50000 MB) source: SLURM	100.00% of timelimit used actual: 5-00:00:18 available: 5-00:00:00	allocated: 8 cores average: 6 cores actual: 26-06:03:53 available: 40-00:02:24 percentage: 65.53%			Low Compute Power
					compute-2-02 2025 samples ncpu			

```
** gpu-2-03
900 samples
ncpu
inuse percentage
1     57.95%
2     1.14%
3     4.55%
4     2.27%
5     3.41%
6     2.27%
8     3.41%
10    2.27%
11    6.82%
12    10.23%
13    5.68%
(16 cores available)
```

Reporting website

- **Same content** but **far more interactive** than HTML emails
- Allow **popups** and **page updates**
- Users can check these **on demand**
- **HTML emails** are still a **crucial piece**



Low Compute Power

Low Compute Power	Cause	The job submission requested no GPU and only a few CPU cores.
	Impact	While it is possible that only large amount of available memory is desired, i.e. your computation is memory-bounded, this combination of request parameter could make job to run in a performance that is **slower than on your laptop**.
	Solution	Confirm your need. Try requesting more CPU cores and setting higher concurrency parameter in your code with consulting library documentations to see if there is improvement. Ignore this message if the computation is memory-bounded and large amount of available memory is the only resource in need.
Low Concurrency	Cause	Samples shows that no GPU and at most single CPU core is used.
	Impact	This combination of request parameter could make job to run

What are being analyzed: Resource Usage

User assumes that the program…

- Can utilize GPU, or multiple GPUs
- Can collaborate across nodes
- Can use as many CPU cores as possible
- Needs a lot of RAM to work

But in fact **not**

- as putting them in use require changing more than just allocation request

They are **unaware** of this
as they cannot see it (**lack of observability**)

Ok we just want users to use less, right?

It is **consuming electricity** whenever cluster is on

- Disk array, Cooling, **U**ninterrupted **P**ower **S**upply

Power efficiency lowers when the job runs for longer

- As portion of facility power consumption goes up

Underusing cluster is also a problem

- A **nicely implemented** scientific software requires **correct use** of SLURM to operate expectedly
  - Users may forget to specify CPU cores
  - They may mistype `#SBATCH -c 32` as `#SBTACH -c 32`
    - **I just did this the day before writing this slide**
  - Both causes a small default to be used and therefore runs **slower than a laptop**, while more **power consuming**
  - Users wait longer than needed to obtain results

Why does GPU have a separate analysis

It is **more limited** than CPU cores and memory

Selections are more varied than CPUs

- T4? A100? H100? 40G? 80G?

Detecting jobs that, **after connected to GPU driver**…

- have **no utilization** at all
- or with **low utilization**
  - Not utilizing GPU well
  - or a **lower spec** one **already satisfies the need**
- Have a long period of zero utilization
  - The job can possibly be split into **GPU part and pure CPU part**
  - Can have some computation done **while waiting for GPU**
  - These changes help further saturating utilization

We are not magician

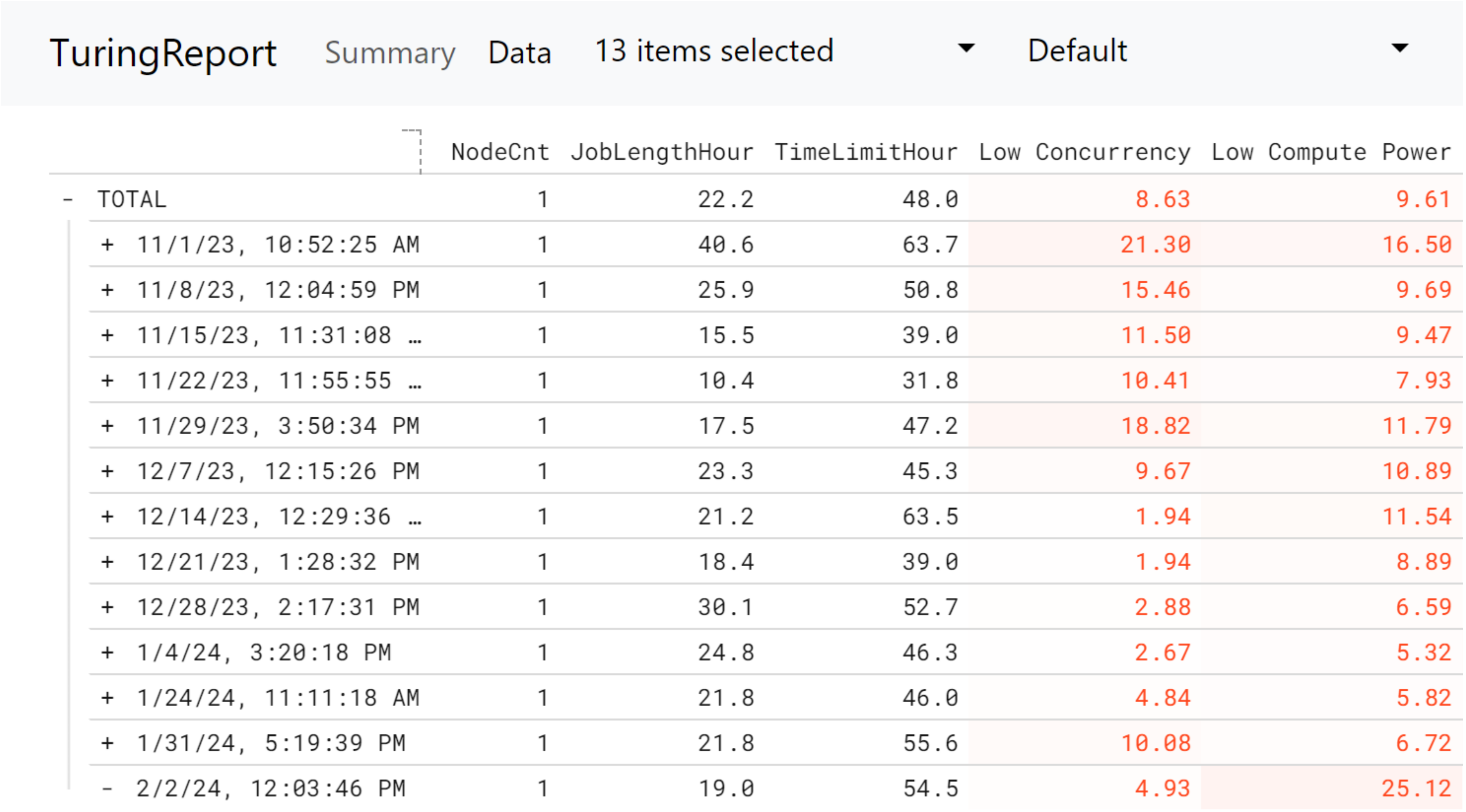
Impossible to tell every possible problem!

Indicative analyses added to alert **anomalies**.

- E.g. **High ratio** of kernel time to user time
  - As time spent in kernel does not help progress actual computation
- Prompt user for case-by-case profiling support to increase program efficacy

Reporting website: admin uses

- Multi-level pivot table like view
- More **clear**
- **Highlights problems**

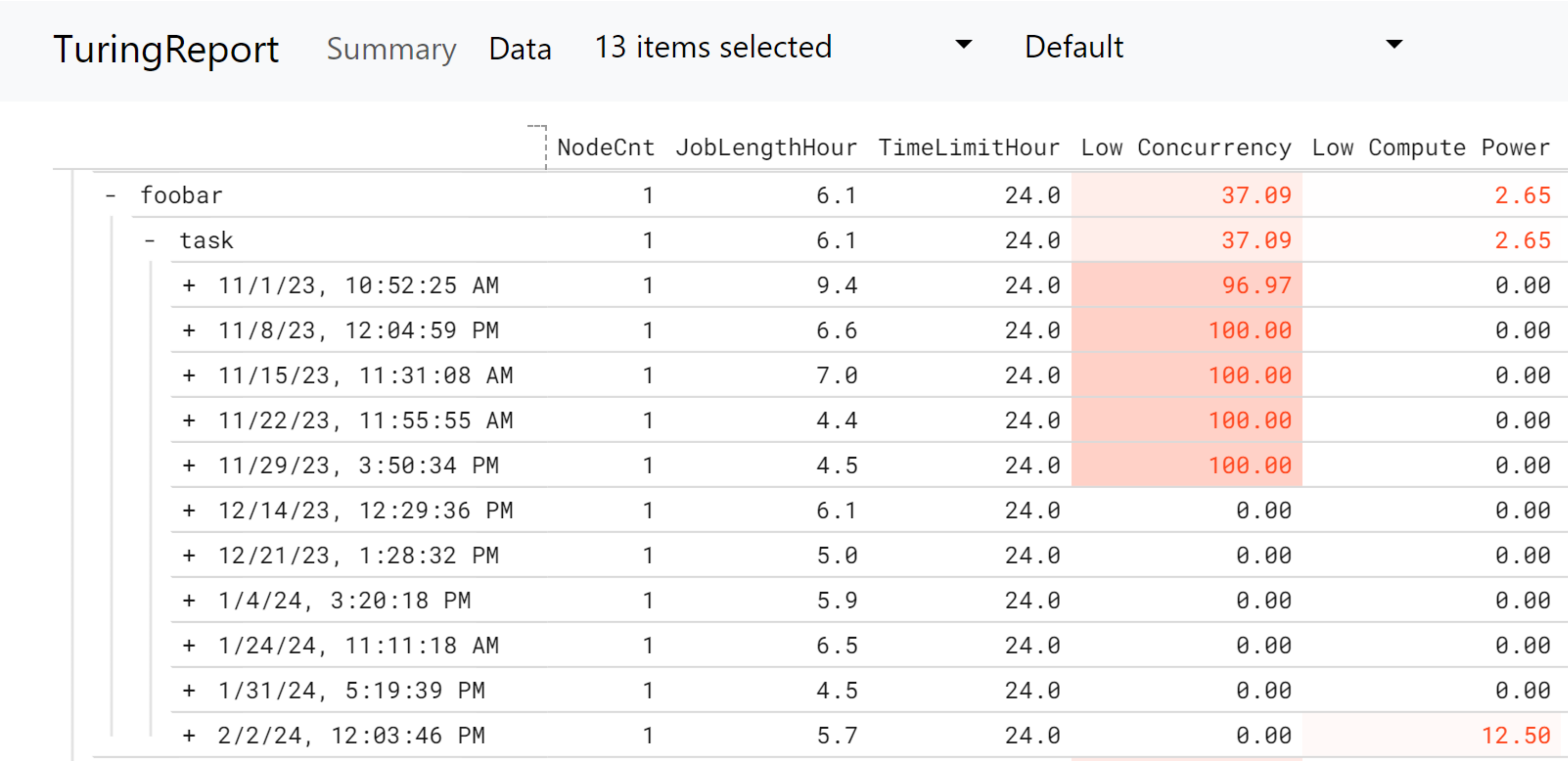


TuringReport Summary Data 13 items selected ▾ Default ▾

	NodeCnt	JobLengthHour	TimeLimitHour	Low Concurrency	Low Compute Power
- TOTAL	1	22.2	48.0	8.63	9.61
+ 11/1/23, 10:52:25 AM	1	40.6	63.7	21.30	16.50
+ 11/8/23, 12:04:59 PM	1	25.9	50.8	15.46	9.69
+ 11/15/23, 11:31:08 …	1	15.5	39.0	11.50	9.47
+ 11/22/23, 11:55:55 …	1	10.4	31.8	10.41	7.93
+ 11/29/23, 3:50:34 PM	1	17.5	47.2	18.82	11.79
+ 12/7/23, 12:15:26 PM	1	23.3	45.3	9.67	10.89
+ 12/14/23, 12:29:36 …	1	21.2	63.5	1.94	11.54
+ 12/21/23, 1:28:32 PM	1	18.4	39.0	1.94	8.89
+ 12/28/23, 2:17:31 PM	1	30.1	52.7	2.88	6.59
+ 1/4/24, 3:20:18 PM	1	24.8	46.3	2.67	5.32
+ 1/24/24, 11:11:18 AM	1	21.8	46.0	4.84	5.82
+ 1/31/24, 5:19:39 PM	1	21.8	55.6	10.08	6.72
- 2/2/24, 12:03:46 PM	1	19.0	54.5	4.93	25.12

Pivot table view: user uses

- Shows changes in problems for same family of jobs across time

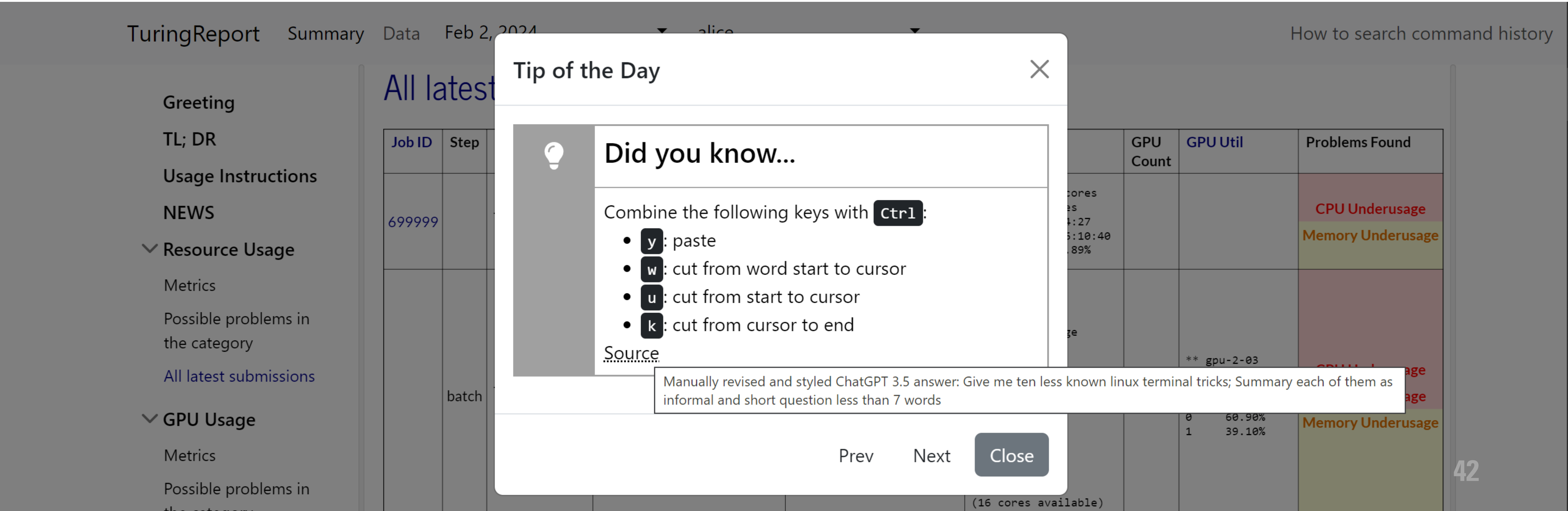


TuringReport Summary Data 13 items selected ▾ Default ▾

	NodeCnt	JobLengthHour	TimeLimitHour	Low Concurrency	Low Compute Power
- foobar	1	6.1	24.0	37.09	2.65
- task	1	6.1	24.0	37.09	2.65
+ 11/1/23, 10:52:25 AM	1	9.4	24.0	96.97	0.00
+ 11/8/23, 12:04:59 PM	1	6.6	24.0	100.00	0.00
+ 11/15/23, 11:31:08 AM	1	7.0	24.0	100.00	0.00
+ 11/22/23, 11:55:55 AM	1	4.4	24.0	100.00	0.00
+ 11/29/23, 3:50:34 PM	1	4.5	24.0	100.00	0.00
+ 12/14/23, 12:29:36 PM	1	6.1	24.0	0.00	0.00
+ 12/21/23, 1:28:32 PM	1	5.0	24.0	0.00	0.00
+ 1/4/24, 3:20:18 PM	1	5.9	24.0	0.00	0.00
+ 1/24/24, 11:11:18 AM	1	6.5	24.0	0.00	0.00
+ 1/31/24, 5:19:39 PM	1	4.5	24.0	0.00	0.00
+ 2/2/24, 12:03:46 PM	1	5.7	24.0	0.00	12.50

User education features

- Shows teasers at top right corner
- Helps users to be more productive and avoid confusions
  - Why is saving with `Ctrl+S` freezing my terminal?
    - possible work loss if terminal is just killed!
  - `Ctrl+Z` says `[1]+ Stopped`, am I good to go?



Tip of the Day

Did you know...

Combine the following keys with `Ctrl`:

- `y`: paste
- `w`: cut from word start to cursor
- `u`: cut from start to cursor
- `k`: cut from cursor to end

Source

Manually revised and styled ChatGPT 3.5 answer: Give me ten less known linux terminal tricks; Summary each of them as informal and short question less than 7 words

```
** gpu-2-03
0      60.90%
1      39.10%
```

Extensibility

- Vital for **adaptability** of different **scenarios** and **use cases**
- Designed for having **capability of**. . .
  - **Adding columns** to database and recording new metrics
    - ... with **existing** migrating and scraping **framework**
  - **Modifying** analysis rules or **creating** new ones
    - ... by simply providing **queries and textual descriptions** to be included in reports
- Customizing **post-processing** or **scheduled tasks** on results
  - Result tarballs containing both **HTML reports** and **raw values** in JSON
  - **Wrapper** prepares working directory and does cleanup work
  - Watcher creates **notification file on tarball updates**
- Extensions have **abundant examples** near sites of change

Some Possible Improvements

- Generate suggestive SLURM arguments as a boilerplate
- Import hierarchy information from SLURM for advisors to see the resource utilization status of their students
- Connect pivot table view with report view to jump to details
- Immediately send user emails when serious misuses observed
  - E.g. dozens of cores allocated but only one core is being used for hours
- Further ease in extending scrapers
  - E.g. as a config of where and how to fetch those data

THANKS!