



<https://hps.vi4io.org>

Zoya Masih

Introduction to Performance Analysis Tools

Outline

- 1 Introduction
- 2 Monitoring
- 3 Profiling
- 4 Tracing
- 5 Summary

Table of Contents

1 Introduction

2 Monitoring

3 Profiling

4 Tracing

5 Summary

From Concepts to Tools

Recap

- We introduced **monitoring, sampling, profiling, tracing**
- Tools differ in:
 - ▶ level of detail
 - ▶ overhead
 - ▶ interpretation of metrics
- Now: Get to know some tools

Objectives

Goal of this session

- Introduce common performance analysis tools
- Understand what each tool measures
- Learn when to use which tool

Tools differ in methodology, scope, and results

Table of Contents

1 Introduction

2 Monitoring

3 Profiling

4 Tracing

5 Summary

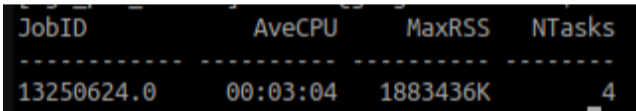
Understanding Slurm job steps: sacct

```
JobID      JobName  Partition  Account  AllocCPUS  State  ExitCode
-----
13251048   profile_m+ grete:sha+ 16        RUNNING    0:0
13251048.ba+ batch      16        RUNNING    0:0
13251048.ex+ extern     16        RUNNING    0:0
13251048.0 bash       16        COMPLETED 0:0
```

- Shows accounting information and lists job steps
- `jobid` : main job
- `jobid.batch` : batch script
- `jobid.extern` : external/system step
- `jobid.0` : actual workload step

Runtime job monitoring: sstat

```
sstat -jobs=your_job-id -format=var_1,var_2, ... , var_N
```

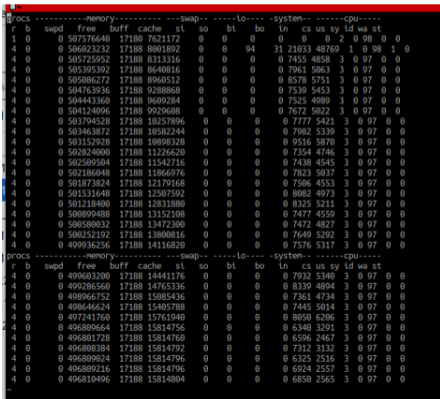


```
JobID           AveCPU           MaxRSS           NTasks
-----
13250624.0      00:03:04        1883436K         4
```

- Works only while job is running
- Monitors job steps

Virtual Memory Statistics: vmstat

- Lightweight Linux monitoring tool
- Gives a quick system-wide view of CPU, memory, swap, and I/O activity



```
procs-----memory-----swap-----lo-----system-----cpu-----
r b swpd free buff cache sl so bi bo ln cs us sy ld wa st
1 0 0 507576640 17188 7621172 0 0 0 0 0 0 2 0 98 0 0
4 0 0 506923232 17188 8001892 0 0 0 0 0 0 0 0 0 0 0
4 0 0 505725952 17188 8313316 0 0 0 0 0 0 0 0 0 0 0
4 0 0 505395392 17188 8640816 0 0 0 0 0 0 0 0 0 0 0
4 0 0 505086272 17188 8960512 0 0 0 0 0 0 0 0 0 0 0
4 0 0 504763936 17188 9288868 0 0 0 0 0 0 0 0 0 0 0
4 0 0 504441360 17188 9609284 0 0 0 0 0 0 0 0 0 0 0
4 0 0 504124096 17188 9929608 0 0 0 0 0 0 0 0 0 0 0
4 0 0 503794528 17188 10257896 0 0 0 0 0 0 0 0 0 0 0
4 0 0 503463872 17188 10582244 0 0 0 0 0 0 0 0 0 0 0
4 0 0 503152928 17188 10898328 0 0 0 0 0 0 0 0 0 0 0
4 0 0 502824000 17188 11226620 0 0 0 0 0 0 0 0 0 0 0
4 0 0 502509504 17188 11542716 0 0 0 0 0 0 0 0 0 0 0
4 0 0 502186048 17188 11866976 0 0 0 0 0 0 0 0 0 0 0
4 0 0 501873824 17188 12179168 0 0 0 0 0 0 0 0 0 0 0
4 0 0 501531648 17188 12507592 0 0 0 0 0 0 0 0 0 0 0
4 0 0 501218400 17188 12831880 0 0 0 0 0 0 0 0 0 0 0
4 0 0 500899488 17188 13152108 0 0 0 0 0 0 0 0 0 0 0
4 0 0 500580032 17188 13472300 0 0 0 0 0 0 0 0 0 0 0
4 0 0 500252192 17188 13800816 0 0 0 0 0 0 0 0 0 0 0
4 0 0 499936256 17188 14116820 0 0 0 0 0 0 0 0 0 0 0
procs-----memory-----swap-----lo-----system-----cpu-----
r b swpd free buff cache sl so bi bo ln cs us sy ld wa st
4 0 0 499603200 17188 14441176 0 0 0 0 0 0 0 0 0 0 0
4 0 0 499286560 17188 14765336 0 0 0 0 0 0 0 0 0 0 0
4 0 0 498966752 17188 15085436 0 0 0 0 0 0 0 0 0 0 0
4 0 0 498646624 17188 15405788 0 0 0 0 0 0 0 0 0 0 0
4 0 0 497241760 17188 15761940 0 0 0 0 0 0 0 0 0 0 0
4 0 0 496899664 17188 15814756 0 0 0 0 0 0 0 0 0 0 0
4 0 0 496501728 17188 15814760 0 0 0 0 0 0 0 0 0 0 0
4 0 0 496098384 17188 15814792 0 0 0 0 0 0 0 0 0 0 0
4 0 0 495899024 17188 15814796 0 0 0 0 0 0 0 0 0 0 0
4 0 0 495889216 17188 15814796 0 0 0 0 0 0 0 0 0 0 0
4 0 0 495810496 17188 15814804 0 0 0 0 0 0 0 0 0 0 0
```

Virtual Memory Statistics: vmstat

Useful columns

- `us, sy, id, wa` : CPU state
- `free, buff, cache` : memory
- `bi, bo` : block I/O
- `si, so` : swap activity

Quick interpretation

- High `id` → CPU mostly idle
- High `wa` → waiting for I/O
- Nonzero `si/so` → swap pressure

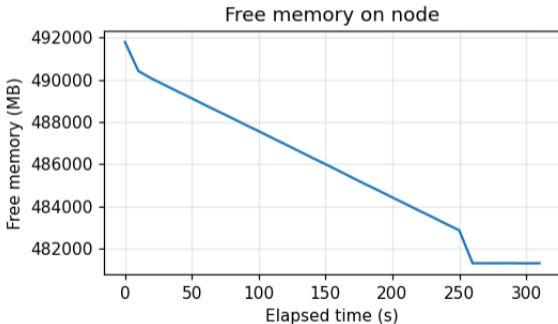
```

procs-----memory-----swap-----lo-----system-----cpu-----
 r b swpd free buff cache st so bi bo in cs us sy id wa st
1 0 0 507576640 17180 7621172 0 0 0 0 0 0 0 2 0 98 0 0
4 0 0 506023232 17188 8001892 0 0 0 94 31 21033 48769 1 0 98 1 0
4 0 0 505725952 17188 8313316 0 0 0 0 0 0 0 7455 4858 3 0 97 0 0
4 0 0 505395392 17188 8640816 0 0 0 0 0 0 0 7961 5063 3 0 97 0 0
4 0 0 505086272 17188 8960512 0 0 0 0 0 0 0 8578 5751 3 0 97 0 0
4 0 0 504763936 17188 9288068 0 0 0 0 0 0 0 7530 5453 3 0 97 0 0
4 0 0 504443360 17188 9609284 0 0 0 0 0 0 0 7525 4989 3 0 97 0 0
4 0 0 504124096 17188 9929608 0 0 0 0 0 0 0 7672 5022 3 0 97 0 0
4 0 0 503794528 17188 10257896 0 0 0 0 0 0 0 7777 5421 3 0 97 0 0
4 0 0 503463872 17188 10582244 0 0 0 0 0 0 0 7982 5339 3 0 97 0 0
4 0 0 503152928 17188 10898328 0 0 0 0 0 0 0 9516 5870 3 0 97 0 0
4 0 0 502824000 17188 11226620 0 0 0 0 0 0 0 7354 4746 3 0 97 0 0
4 0 0 502509504 17188 11542716 0 0 0 0 0 0 0 7438 4545 3 0 97 0 0
4 0 0 502186048 17188 11866976 0 0 0 0 0 0 0 7823 5037 3 0 97 0 0
4 0 0 501873824 17188 12179168 0 0 0 0 0 0 0 7506 4553 3 0 97 0 0
4 0 0 501531648 17188 12507592 0 0 0 0 0 0 0 8082 4073 3 0 97 0 0
4 0 0 501218400 17188 12831880 0 0 0 0 0 0 0 8325 5211 3 0 97 0 0
4 0 0 500899488 17188 13152108 0 0 0 0 0 0 0 7477 4559 3 0 97 0 0
4 0 0 500580032 17188 13472300 0 0 0 0 0 0 0 7472 4827 3 0 97 0 0
4 0 0 500252192 17188 13800816 0 0 0 0 0 0 0 7649 5292 3 0 97 0 0
4 0 0 499936256 17188 14116820 0 0 0 0 0 0 0 7576 5317 3 0 97 0 0
procs-----memory-----swap-----lo-----system-----cpu-----
 r b swpd free buff cache st so bi bo in cs us sy id wa st
4 0 0 499603200 17188 14441176 0 0 0 0 0 0 0 7932 5340 3 0 97 0 0
4 0 0 499286560 17188 14765336 0 0 0 0 0 0 0 8339 4894 3 0 97 0 0
4 0 0 498966752 17188 15085436 0 0 0 0 0 0 0 7361 4734 3 0 97 0 0
4 0 0 498646624 17188 15405788 0 0 0 0 0 0 0 7445 5014 3 0 97 0 0
4 0 0 497241760 17188 15761940 0 0 0 0 0 0 0 8050 6206 3 0 97 0 0
4 0 0 496809664 17188 15814756 0 0 0 0 0 0 0 6340 3291 3 0 97 0 0
4 0 0 496801728 17188 15814760 0 0 0 0 0 0 0 6596 2467 3 0 97 0 0
4 0 0 496808384 17188 15814792 0 0 0 0 0 0 0 7312 3132 3 0 97 0 0
4 0 0 496809024 17188 15814796 0 0 0 0 0 0 0 6325 2516 3 0 97 0 0
4 0 0 496809216 17188 15814796 0 0 0 0 0 0 0 6924 2557 3 0 97 0 0
4 0 0 496810496 17188 15814804 0 0 0 0 0 0 0 6850 2565 3 0 97 0 0

```

Virtual Memory Statistics: vmstat

- Parsed `vmstat` log and reconstructed time series
- Visualized free memory over time using Python (`matplotlib`)



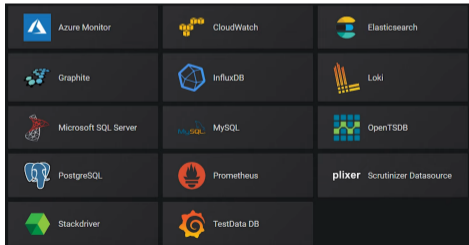
nvidia-smi

- `nvidia-smi`
- Lightweight GPU monitoring tool
- GPU utilization (%) & Memory usage
- Running processes

```
Fri Apr 3 11:40:25 2026
-----
| NVIDIA-SMI 570.211.01                  Driver Version: 570.211.01      CUDA Version: 12.0      |
|-----|-----|-----|-----|-----|-----|-----|-----|
| GPU  Name      Persistence-N | Bus-Id          Disp.A    Volatile Uncorr. ECC  |
| Fan  Temp      Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M.  | | | | | |
|---|---|---|---|---|---|---|---|
| 0   NVIDIA A100-SXM4-80GB | On              00000000:2F:00:0 Off |           512SMiB / 81920MiB | 0%      Default  |
| N/A  22C      P0              72W / 500W     |                       |           Disabled  |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 1   NVIDIA A100-SXM4-80GB | On              00000000:30:00:0 Off |           4MiB / 81920MiB | 0%      Default  |
| N/A  20C      P0              61W / 500W     |                       |           Disabled  |
|-----|-----|-----|-----|-----|-----|-----|-----|
| Processes:                               |
| GPU   GI    CI          PID    Type   Process name          GPU Memory |
| ID   ID   ID           ID          |                   Usage         |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 0   N/A  N/A         1879316  C     ...                  ./bin/python          1276MiB |
| 0   N/A  N/A         1879317  C     ...                  ./bin/python          1276MiB |
| 0   N/A  N/A         1879318  C     ...                  ./bin/python          1276MiB |
| 0   N/A  N/A         1879319  C     ...                  ./bin/python          1276MiB |
```

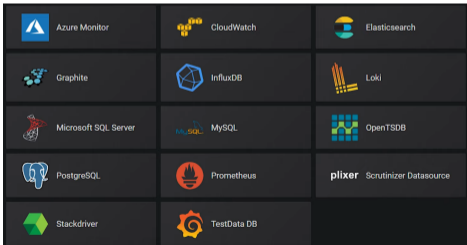
Grafana

- Visualization platform
- Metrics collected from systems



Grafana

- Visualization platform
- Metrics collected from systems



- CPU, memory, I/O usage
- network activity
- multi-node behavior
- cluster monitoring
- real-time + historical analysis



Table of Contents

1 Introduction

2 Monitoring

3 Profiling

4 Tracing

5 Summary

CPU and Memory Profiling: LIKWID

- Hardware counter-based profiling
- Measures CPU performance counters, Cache behavior and mem bandwidth
- **Strength:**
 - ▶ Lightweight, fast
 - ▶ No recompilation required
- **Limitation:**
 - ▶ Its usefulness on clusters depends entirely on admin configuration. Without proper permissions, you cannot rely on it

CPU and Memory Profiling: LIKWID

■ How to use:

- ▶ `module spider likwid`
- ▶ `module load likwid/5.3.0`
- ▶ `likwid-perfctr -C 0 -g CLOCK ./a.out`

```
-----  
CPU name:      AMD EPYC 7343 16-Core Processor  
CPU type:      AMD K19 (Zen3) architecture  
CPU clock:     3.28 GHz  
ERROR - [./src/includes/perfmon_perfevent.h:perfmon_setupCountersThread_perfevent:1435] Permission denied  
Setup of event ACTUAL_CPU_CLOCK on CPU 2 failed: Permission denied  
ERROR - [./src/includes/perfmon_perfevent.h:perfmon_setupCountersThread_perfevent:1435] Permission denied  
Setup of event MAX_CPU_CLOCK on CPU 2 failed: Permission denied  
-----  
Failed to execute command: ./a.out  
-----  
Group 1: CLOCK  
-----  
|      Event      | Counter | HwThread 2 |  
-----  
| ACTUAL_CPU_CLOCK | FIXC1   | 0           |  
| MAX_CPU_CLOCK    | FIXC2   | 0           |  
| RETIRED_INSTRUCTIONS | PMCB    | 249774      |  
| CPU_CLOCKS_UNHALTED | PMC1    | 311471      |  
| RAPL_PKG_ENERGY   | PMR1    | 0           |  
-----  
-----  
|      Metric      | HwThread 2 |  
-----  
| Runtime (RDTS) [s] | 0.0028 |  
| Runtime unhaltd [s] | 0.0001 |  
| Clock [MHz]        | -       |  
| CPI                | 1.2478 |  
| Energy [J]         | 0       |  
| Power [W]          | 0       |  
-----
```

GPU Profiling: Nsight

- NVIDIA profiling and analysis tool
- Provides detailed insight into GPU execution
- Kernel execution timeline
- GPU utilization over time
- Memory transfers (CPU ↔ GPU)

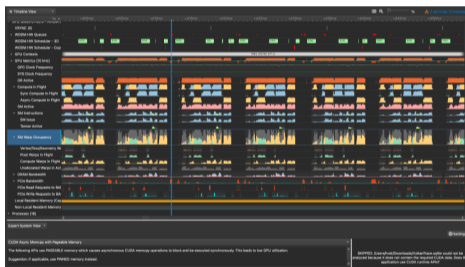


Figure:
<https://developer.nvidia.com>

IO Profiling Tool: Darshan

```
zoya@zoya-Latitude-7420:~$ darshan-parser zmasih_io500_id15155848-497182_3-8-55773-7894066996975896480_1.darshan
# darshan log version: 3.41
# compression method: ZLIB
# exe: ./io500 myconfig.ini
# uid: 711947
# jobid: 15155848
# start_time: 1678285773
# start_time_ascii: Wed Mar  8 15:29:33 2023
# end_time: 1678314671
# end_time_ascii: Wed Mar  8 23:31:11 2023
# nprocs: 10
# run time: 28897.3787
# metadata: lib_ver = 3.4.1
# metadata: h = romio_no_indep_rw=true;cb_nodes=4
```

- Measures access patterns and Read/write behavior
- DXT: extended tracing
- **Strength:**
 - ▶ Very low overhead
 - ▶ System-wide deployment
- **Limitation:**
 - ▶ inability in handling large-scale I/O patterns, especially on Python workloads

IO Profiling Tool: Darshan

Example insights

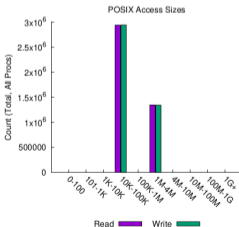
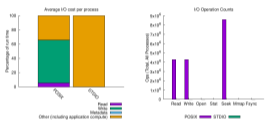
- Total I/O and bandwidth
- Read vs. write time distribution
- Operation counts (read/write/seek)

Interpretation

- Write dominates runtime
- Many small I/O operations
- Sequential access pattern

io500 (3/8/2023) 1 of 3

jobid: 15155848 uid: 711947 nprocs: 10 runtime: 28897.3787 seconds

I/O performance estimate (at the POSIX layer): transferred 5636192.9 MiB at 259.24 MiB/s
I/O performance estimate (at the STDIO layer): transferred 0.0 MiB at 0.05 MiB/s

Dask Dashboard (Visualization Profiling)

- Dask is a flexible **Python library** for parallel computing
- In distributed hardware, particularly for data science and ML
- Dask Dashboard is an interactive monitoring and **profiling framework**
- Dask-specific (not general-purpose HPC tool)



Dask Dashboard

- Task execution and progress,
- CPU utilization and occupancy
- Memory usage per worker
- Data transfer between workers

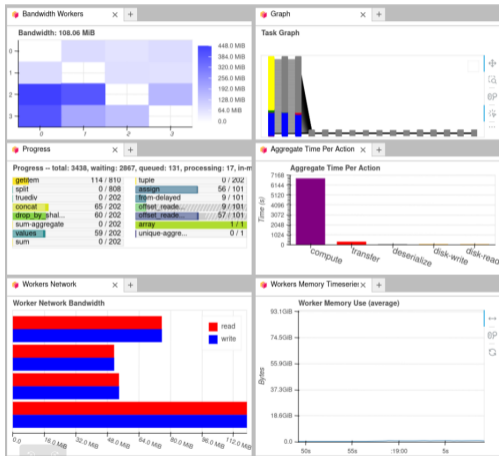


Table of Contents

1 Introduction

2 Monitoring

3 Profiling

4 Tracing

5 Summary

Score-P

- A comprehensive profiling and tracing tool for HPC applications
- **Strength:**
 - ▶ flexible visualization
 - ▶ strong for large-scale traces
- **Limitation:**
 - ▶ not supporting python multiprocessing and tracking `importlib.reload()`
 - ▶ only supports MPI or SHMEM.
Any other multiprocessing approach cannot be traced

We will discuss this tool in more details in the next session

DFTracer

- Integration & Visualization: DF analyzer & Perfetto UI
- Measures compute, data loading, I/O, communication
- **Strength:**
 - ▶ specific design for ML pipelines
 - ▶ fine-grained categorization
 - ▶ can handle python multiprocessing
 - ▶ high control via decorators. You explicitly choose what to trace
- **Limitations**
 - ▶ need to: identify important functions, and risk of missing important regions
 - ▶ inconsistent instrumentation across experiments
 - ▶ lack of documentations

Table of Contents

- 1 Introduction
- 2 Monitoring
- 3 Profiling
- 4 Tracing
- 5 Summary**

Summary

- Tools implement different analysis methods
- No single tool is sufficient
- Never rely on the metrics blindly!
- You need to know what the statistics show for real
- Choose based on:
 - ▶ level of detail needed
 - ▶ overhead tolerance
 - ▶ target bottleneck (CPU, I/O, GPU)
- Performance analysis = **tool + interpretation**