Argonne NATIONAL LABORATORY

Characterization and **Identification** of HPC Applications at Leadership Computing Facility

International Conference on Supercomputing 2020

June 29 - July 2, 2020. Worldwide online event

ICS

acm Association for Computing Machinery

Zhengchun Liu[1,2], Ryan Lewis[3], Rajkumar Kettimuthu[1,2], Kevin Harms[1], Philip Carns[1], Nageswara Rao[4], Ian Foster[1,2] and Michael Papka[1,3].

[1]Argonne National Laboratory, [2]University of Chicago, [3]Northern Illinois University, [4]Oak Ridge National Laboratory

Presented by:
Zhengchun Liu,
Assistant Computer Scientist

Motivation

- Leadership computing facilities (A.K.A supercomputer center) collects various logs for debugging, troubleshooting, auditing purpose etc.
- Reuse logs for characterization of applications, i.e., without benchmarking.
- Closer look at real applications in production, to guide design of next generation computer via hardware-software co-design.
- Researchers and tool developers to build new, or optimize existing, frameworks and runtime to better assist actual users.
- …

Agenda

1. **Background**
   - ✓ Datasets
   - ✓ Terms

2. **Characterization / Observations**
   - ✓ Run time break down
   - ✓ Users' Job - composition(task) and size
   - ✓ File I/O
   - ✓ Fine grained hardware resource usage

3. **Identification**
   - ✓ Fingerprint
   - ✓ Identification

Background

Mira, 2012 - 2019, a 10 petaFLOPS IBM Blue Gene/Q system at ALCF.

Terms:

- **Application**: a program (executable) that runs at facility.
- **Job**: The computation on the computing resources allocated in response to a user request.
- **Task**: An application execution within a job. A job may comprise one or more tasks.
- **Process**: An MPI process/rank.
- **Communication**: Inter-process communication via MPI, whether inter-node or intra-node.

Datasets:

- **Cobalt** scheduler logs, users' job level
- **Control system**: task level logs
- **Darshan**: records file I/O behavior
- **RAS Event**: IBM's records of reliability, availability, and serviceability
- **Autoperf**: MPI information and limited hardware performance counter.

Parsed logs (csv files) are available at:
https://reports.alcf.anl.gov/data/

Agenda

1. **Background**
   - ☑ Datasets
   - ☑ Terms

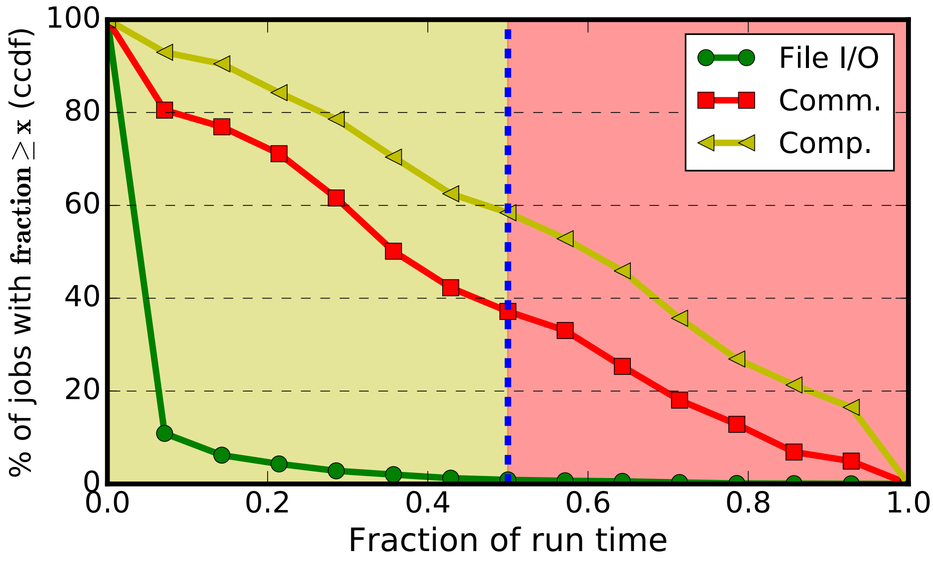
2. **Characterization / Observations**
   - ☑ Run time break down
   - ☑ Users' Job - composition(task) and size
   - ☑ File I/O
   - ☑ Fine grained hardware resource usage

More observations are available in the paper

3. **Identification**
   - ☑ Fingerprint
   - ☑ Identification

Characterization – Jobs run time break down

For each application: Time on communication is recorded in Autoperf and can be accumulated. Time on File I/O is recorded in Darshan. We then can break down the runtime into I/O, communication, and the rest it computing.

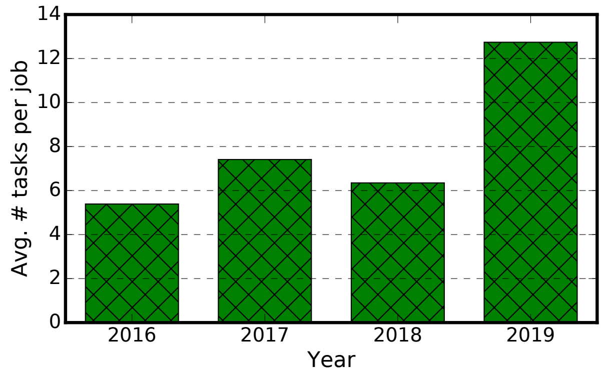


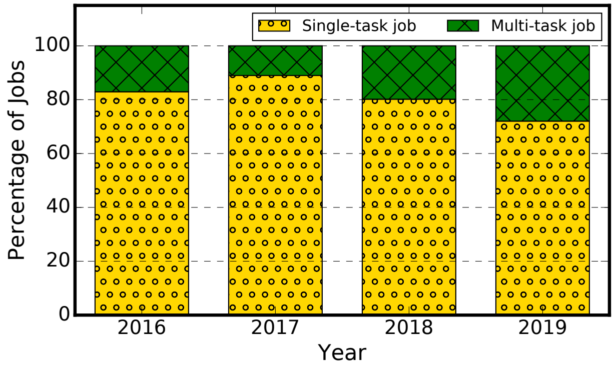
Observation:

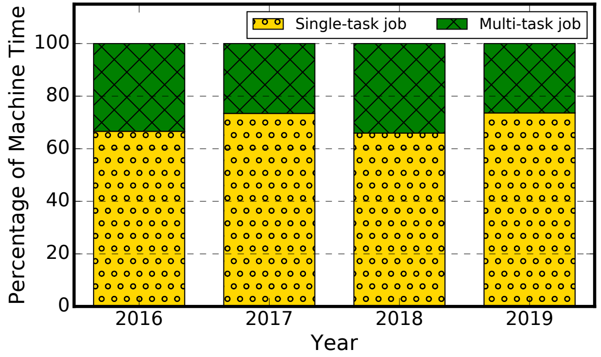
- Overall, about 40% of the tasks spent more than half of their machine time on MPI.
- For nearly 95% of the tasks, file I/O took less than 20% of their total machine time.
- Computing took the most of the machine time, e.g., more than 60% of the jobs spent at least half of their time on computing.

These findings show that computation is still the most time consuming operation of HPC applications.

Characterization — job size



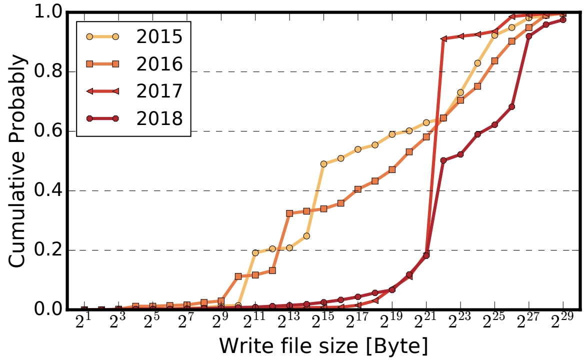


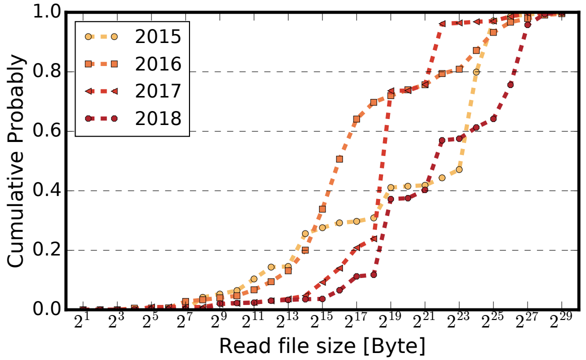


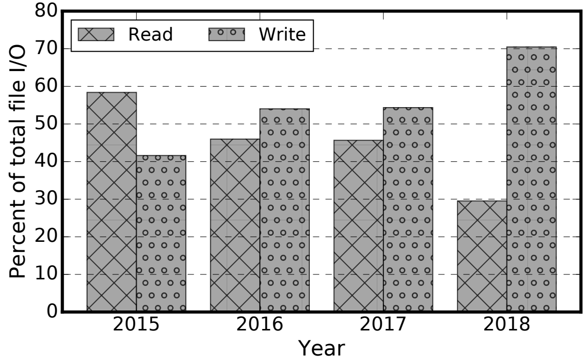
Observation: Multi-task jobs are common and account for a considerable share of machine time. It suggests that better system-level support for such jobs (e.g., via support for MPI task management features and for workflow languages such as Parsl and Swift) may be desirable.

Moreover, tool may be improved to support for dependency-aware task level scheduling, with the goal of increased backfilling.

Characterization – File I/O – file size







Observation: average size of files read and written on Mira is small, hindering efficient parallel I/O. This observation suggests more work is needed to educate users as to the benefits of larger files and/or to adapt storage and I/O systems to perform better for smaller files.

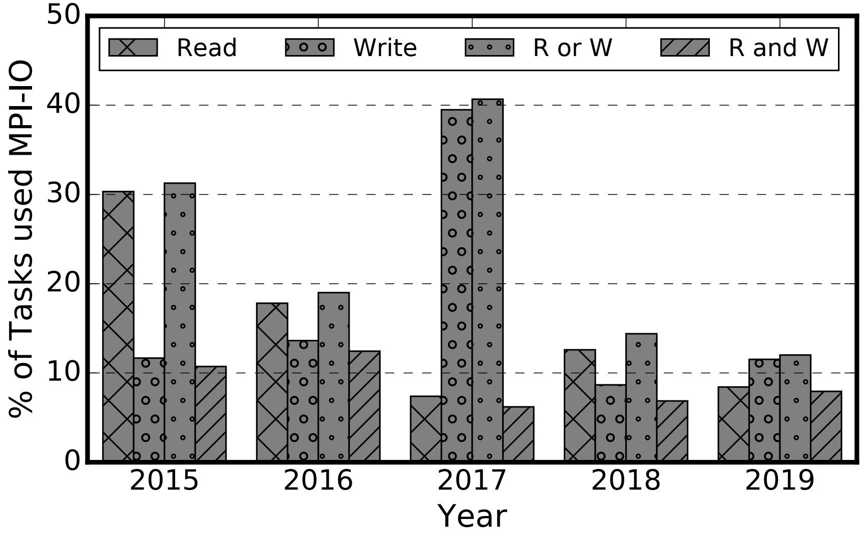
We see that from 2016 onwards the storage system is dominated by write workloads. In 2018, more than twice as many bytes were written than read.

Characterization – File I/O – Library

MPI-IO uses posix but Darshan captures posix without distinguishing raw posix from MPI-IO.

We mark tasks that perform MPI-IO based I/O operations (reads or writes), for at least one file, as MPI-IO aware.

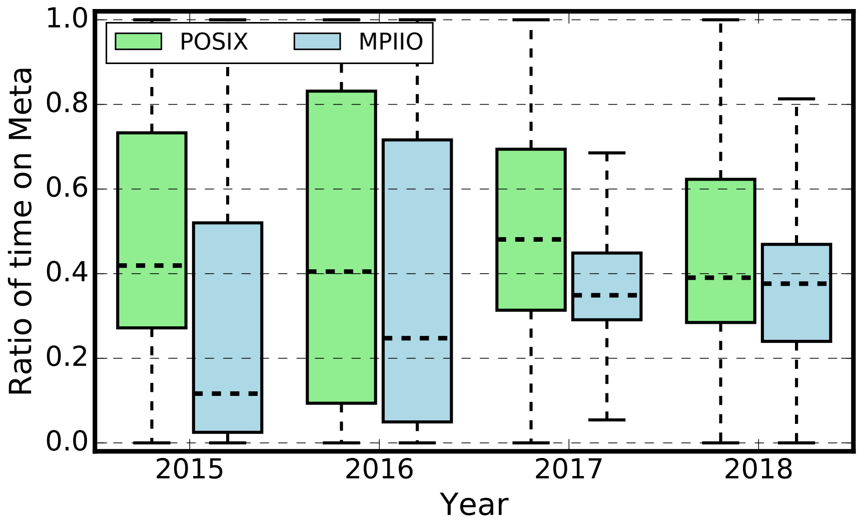
If no MPI-IO operations are used to read/write files, we mark those tasks/applications as basic posix only.



Observation: basic posix is much more widely used than the high-level parallel MPI-IO. Although not all POSIX-using applications achieve poor I/O throughput, this observation motivates investigations of why higher-level parallel I/O libraries are not more widely used.

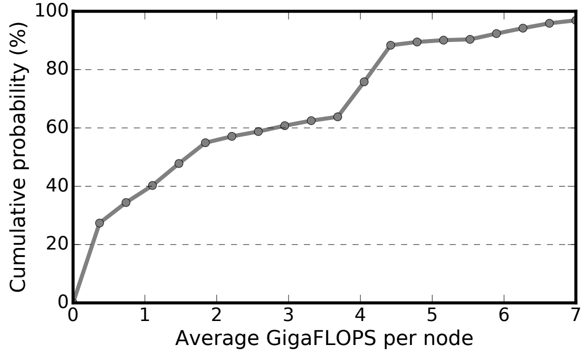
Characterization – File I/O – metadata operations

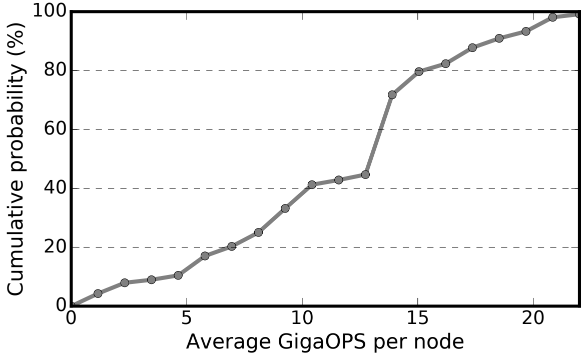
For each file read or written, we extracted the total time as well as the time spent on operations related to metadata from Darshan logs. We then computed the fraction of time on metadata operations for each file. →

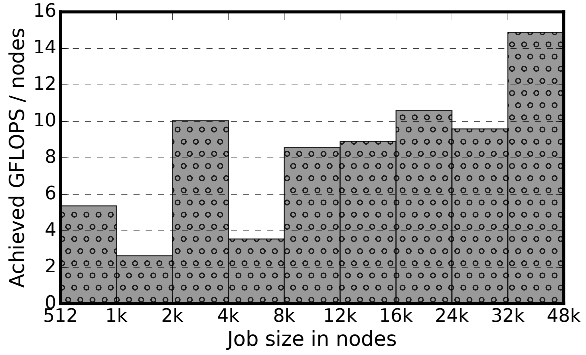


Observation: Metadata operations account for a significant portion of total I/O time, due to the fact that the majority of files are small.
Where feasible, system administrators should optimize their system configuration for smaller files(e.g., small files on Metadata server).

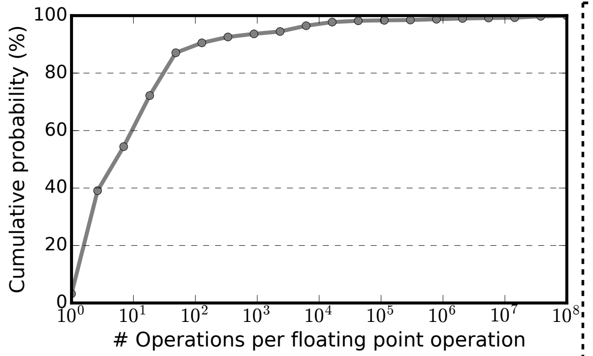
Characterization — fine grained hardware resource (FLOPs)







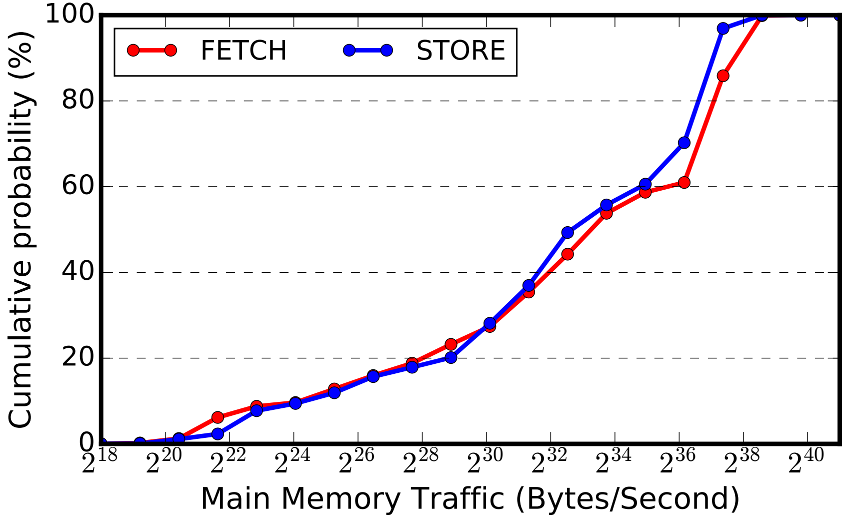
Observation: Most applications achieved pretty low (~2% of peak) FLOPS although some power (4% of annual hours) users(e.g., HACC) achieved 70% of peak. Possible explanation is that performance aware users turned off Autoperf because of its overhead concern.↑ Counter-intuitively, large applications achieved better FLOPS, probably users put more effort into optimizing their application at scale.

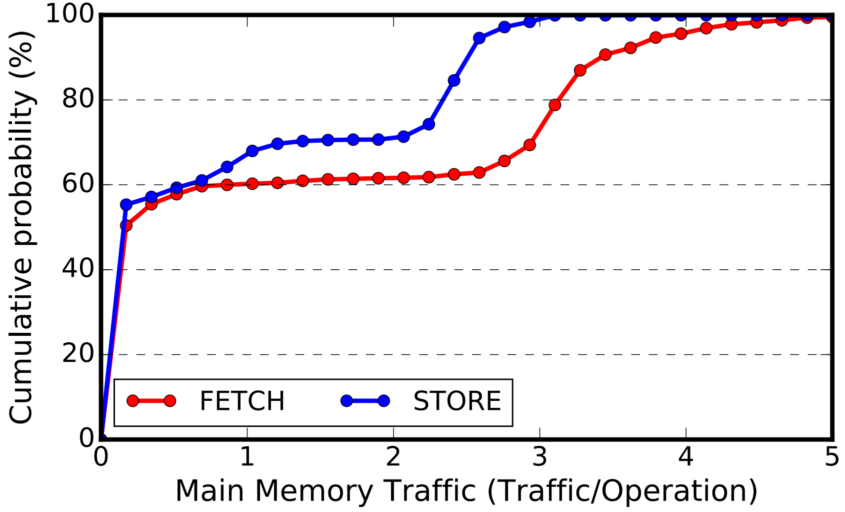


→at least **10** auxiliary operations are performed for each floating point operation.

TOP500 project uses FLOPS achieved by the LINPACK benchmark (high percentage of floating-point operations) as the metric to rank supercomputers, the balance of OPS and FLOPS may be a better metric to quantify supercomputer capability.

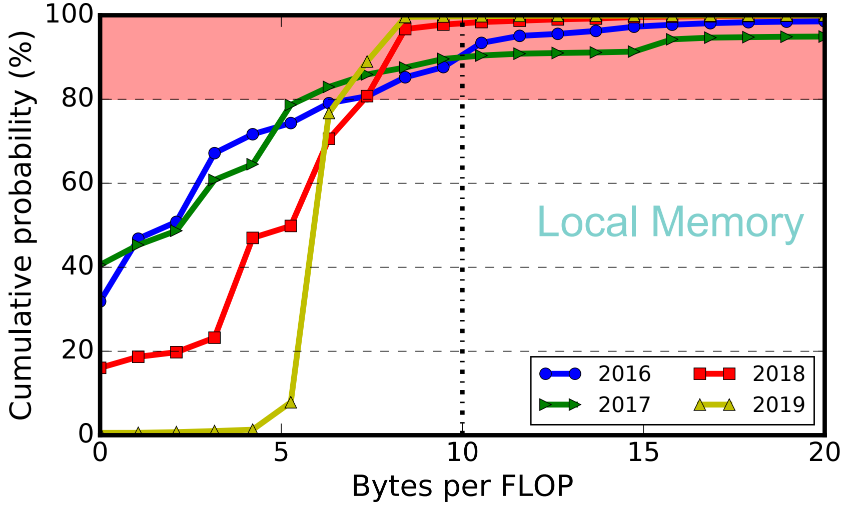
Characterization – fine grained hardware resource (RAM)

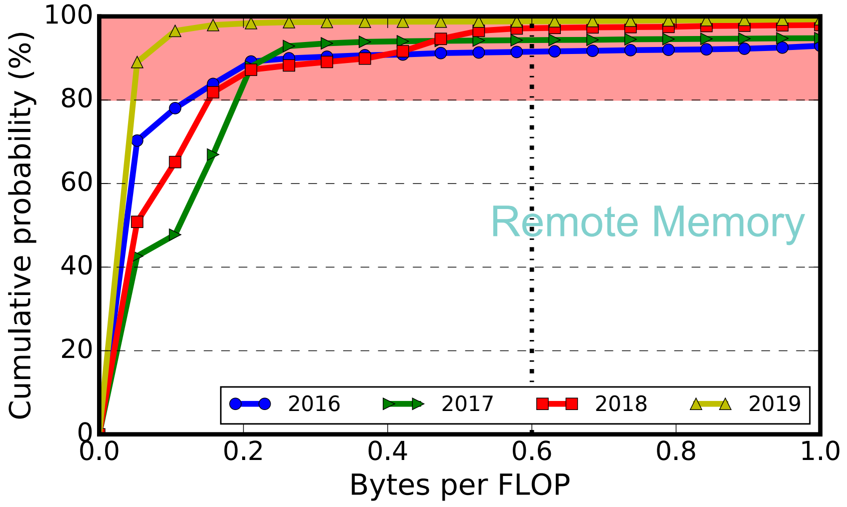




Observation: When compared with FLOPS achieved, we find that main memory throughput is much closer to its peak. This observation reveals how worrisome the “memory wall"(i.e., the growing disparity of speed between CPU and memory outside the CPU chip) is for applications on HPC systems.

Characterization – Memory Bytes-per-FLOP





Observation: The bytes exchanged with other processes per FLOP, and the bytes accessed from local memory per FLOP, decrease year over year. This trend is in line with that of supercomputers' hardware bytes-per-FLOP capability.

These numbers are informative to the design of supercomputer as the applications' bytes-per-FLOP characterization reflects the actual demand.

Agenda

1. **Background**
   - ✓ Datasets
   - ✓ Terms

2. **Characterization / Observations**
   - ✓ Run time break down
   - ✓ Users' Job - composition(task) and size
   - ✓ File I/O
   - ✓ Fine grained hardware resource usage

More observations are available in the paper

3. **Identification**
   - ✓ Fingerprint
   - ✓ Identification

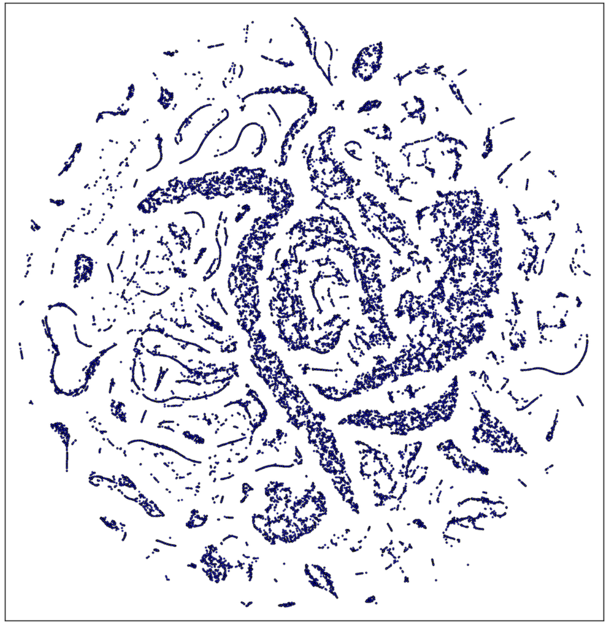
Identification – fingerprint

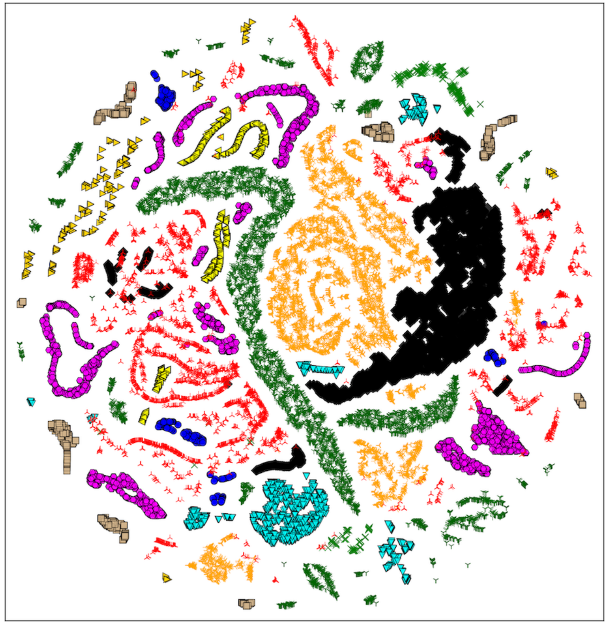
Based on our analysis and understanding, we next engineer and extract features from logs to represent applications in order to study the commonality of HPC applications.

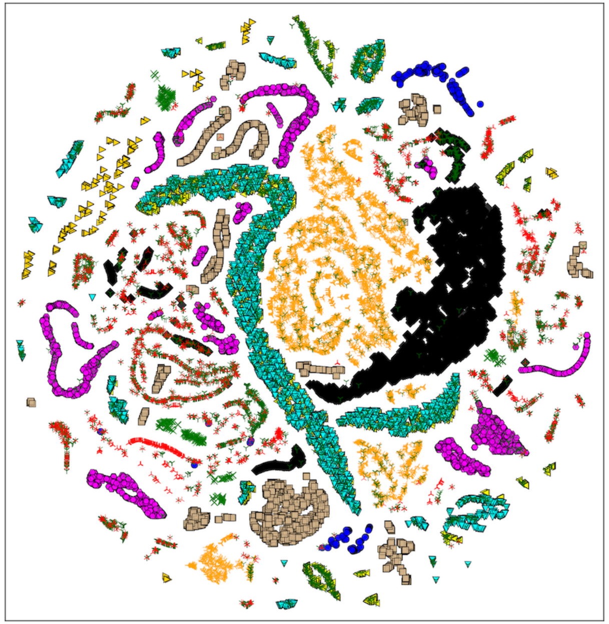
- Fraction of runtime used for inter-process communication(i.e., time spent on MPI routines);
- Fraction of runtime used for MPI-IO;
- Achieved operations per second;
- Achieved floating point operations per second;
- Number of processes per node;
- Average RAM fetch per CPU cycle; and
- Average RAM store per CPU cycle.

All are interpretable features.

Identification – finger print

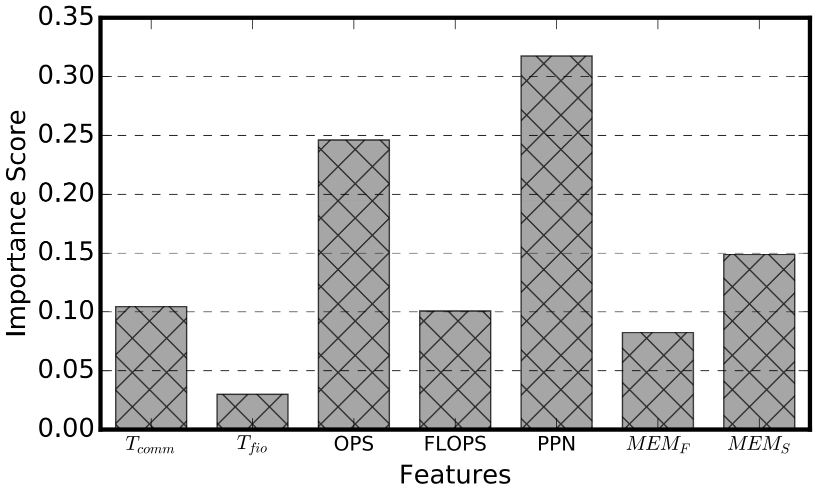

T-SNE 2D embedding


Colored by executable name


Colored by username

Identification – ml-based identification model

- 127,585 Autoperf sampled tasks
- Top 20 applications (covered 95%) plus others, totals 21 labels
- 70% for training and 30% for testing
- XGboost model with max_depth=10
- 5-fold cross-validation
- Our testing accuracy is 99.5% (half mistakes are the “other” label)



Summary: Tasks can easily be grouped and identified using our interpretable feature representation. Statistically, it means that the proposed features can explain the various behaviors and characteristics of different applications.

Conclusion and Future work

- ☑ We characterized applications by co-analyzing 5 production log-sets and draw several insightful observations.
- ☑ We believe our analysis can help researchers, tool developers, resource providers, end users, and funding agencies from difference perspective.
- ☑ Based our observations and understanding, we proposed interpretable representation (i.e., fingerprint) of application runs.
- ☑ A machine learning model using our representation can accurately identify application.

- ☐ Classifying application, based on the run time breakdown and fingerprint, as computing-, communication- or I/O intensive for smart scheduling.
- ☐ Hardware performance counter ONLY (light weight) based identification.

THANKS!