Trenza Survey Performance Collector

The Survey collector/analytics framework is a new generation, high-level, light-weight tool for HPC application performance metric collection. This was originally conceived to provide a broad collection and reporting tool with less impact than the more in-depth performance tools such as HPCToolKit and Open|Speedshop. Survey is a multi-platform Linux tool which targets collection of high-level performance metrics and analysis of applications running on both single node and large-scale platforms, including the Cray platforms.

The Survey collector is designed to work on sequential, MPI, OpenMP, and hybrid codes and directly leverages several interfaces available for tools inside current MPI implementations including: MPICH, MVAPICH, MPT, and OpenMPI. It supports multiple architectures and has been tested on machines based on Intel, AMD, ARM, and IBM P8/9 processors and integrated GPUs.

The Survey framework has the capability to utilize external collection tools to collect additional metrics that can cover additional aspects of the machine and environment (the NVIDIA-smi integration was the first). Application output can also be parsed to collect and monitor application specific data. https://trenza.gitlab.io/survey.io/

The **survey** collector features include:

- Is very lightweight with target goal of 1 % overhead
- Gathers multiple application performance metrics in one run
- Gathers job metadata that includes job, hardware, and system
- Gives a high-level performance overview (no mapping back to the source)
- Identify potential areas that you may want to use a more detailed tool such as Open|SpeedShop, HPCToolkit, or other tools to drill into specifics.
- Creates data files for application metrics (csv and json) and metadata (json)
    - For ingestion to local analysis frameworks
    - An extractor is available to pull specific metrics and metadata
- All raw per-thread of execution csv files are available after run (directory that contains per thread files)
- min, max, average output across threads of execution (including top-down for Intel)

survey performance metrics gathered include:

- *Summary job metadata- includes system and job data (*metadata..json)*

executable	*hardware*	*system*
• *cmd line* • *linked libraries* • *launch - start/end* • *# threads/ranks*	• *cpu* • *memory* • *file systems* • *HW components*	• *operating system* • *resource limits* • *environmental variables* • *slurm info (RM)*

- *Runtime collection (*report.json, *report.csv) (can be individually enabled/disabled)*

  - *Aggregate Metrics of all MPI ranks and OMP thread*
  - *Memory information, like high water mark, memory allocation and free calls, allocation sizes*
  - *Hardware counter information and derived metrics*
  - *Input/output information, I/O time, read / write times and byte counts*
  - *MPI information, MPI time and percent across the threads of execution*
  - *OpenMP information, serial time and time spent in OpenMP regions*

- *External collectors (added to *report.json, *report.csv)*

  - *NVIDIA-smi - provides CUDA GPU utilization information*
  - *Others can be developed*

The goal is to provide users with a high-level view of the performance of their application. Survey is targeting multiple use cases which include:

- SW developer or team –
  - User/team utilizing for performance assessment and comparison. There are options for Intel top-down analysis with additional integrations that can be done.
  - Data is in either csv or json format and a user/team can upload to an analysis tool of choice to further derive metrics of their own making.
  - This could also be easily integrated into regression test frameworks.
- Continuous Integration –
  - Can be integrated into a CI pipeline such as GitLab CI.
  - Would allow a pipeline to not only report back that it ran successfully but also report back performance metrics and metadata for that run.

    - Cross-architecture development would benefit from metric comparison.
    - A dashboard would be a future addition to this case (extractor functionality).
- System testing –
    - Since it is easily integrated and lightweight, the Survey tool could be integrated into machine acceptance tests, post DST, and other testing where performance metrics would be beneficial for comparison to identify system or software deployment issues.
- Monitoring system integration –
    - Integration into a system monitoring infrastructure would provide a view of system resource metrics and metadata from the applications perspective.

Trenza Survey is available with flexible subscription licensing based on the use case. Site subscriptions can be established based on the requirements and needs of the site. This was done as a balance between open source and providing support to develop a sustainable framework.

How it works:

The **survey** collector generates summary csv/json files from metric data based on the per rank/thread integrated performance collector. These are generated when the application that is being **surveyed** completes. How to run:

survey <run options> < how you run your application normally>

example, with MPI:
survey mpirun –np <number of ranks> <application><application arguments>

The **survey** primary output are the two json files that contain the metrics and metadata. The data provides a rich collection of data that can be extracted from based on the user (or use case) needs and be fed into other tracking or analysis tools. The output will consist of metadata and application runtime (report) json files. There is also a report csv file as another format. In addition, there is a csv directory that contains a structure with the per-thread csv files that will be available after the survey experiment completes.

external collectors - survey provides the capability to integrate collection tools that are external to the survey tool. The first example is the ability to kick off the nvidia-smi collection tool if you are running in a capable environment, extracting data from that tool, and integrate the data into the survey report json file. We expect that other vendor or community tools can be tapped on to provide additional valuable data for analysis.

A **survey_extractor** is available to pull specific metrics and metadata from the data collected. This can be used to add to SQLite databases, extract to specific formats and generate reports and graphs. We have rudimentary capabilities and are seeking additional feedback to better target use. Currently in progress:

- compare across files
- DB feed - for graphing and compare
- select specific data for extraction