A Lightweight GPU Monitoring Extension for Pegasus Kickstart

George Papadimitriou*, Ewa Deelman*
{georgpap, deelman}@isi.edu
*Information Sciences Institute, University of Southern California, Marina Del Rey, CA, USA

***Abstract*—This paper presents a lightweight tool to capture monitoring information from Nvidia GPUs. The tool is an extension of the Pegasus Kickstart wrapper that was originally designed for monitoring CPU-based workflow jobs.**

***Index Terms*—scientific workflows, Pegasus, workflow management systems, Nvidia, GPU, monitoring**

I. Introduction

Compute jobs in the Pegasus workflow management system (WMS) [1] are wrapped using a lightweight C executable called “pegasus-kickstart” (Kickstart) [2], [3] that captures runtime job performance and provenance data. The toolkit collects useful information about the execution of the wrapped task such as the environment setup, performance data and output logs. Kickstart is a very important component of the Panorama data collection architecture [4]. In Pegasus’ Panorama branch [5], Kickstart has been extended to include fine-grained monitoring capabilities that can pull resource usage statistics of workflow running tasks within a user-defined time interval. This information is then published to an AMQP [6] endpoint in JavaScript Object Notation (JSON) format so it can be ingested into a repository, saved to a storage system, or uploaded to an analysis framework (e.g., Elasticsearch [7]). Until now though, Kickstart could only collect statistics available in Linux’s procfs [8], ignoring other subsystems, such as graphics processing units (GPUs).

II. Approach

To extend Kickstart’s capabilities with monitoring support for Nvidia GPUs, we are leveraging Nvidia’s monitoring library (NVML) [9]. NVML offers a C-based API for monitoring and managing various states of Nvidia GPU devices. We have extended Kickstart (Figure 1) with a lightweight C wrapper for the NVML library that queries the state of all the GPU devices available on an execution host machine. Kickstart polls for new GPU statistics on a user-defined interval and populates JSON formatted events. Kickstart GPU polling supports multithreading and creates a new polling thread for each GPU device, which is essential when sampling the PCI-Express bus utilization.

A. Events

During a job’s execution there are three events produced by Kickstart containing information about the Nvidia GPUs.

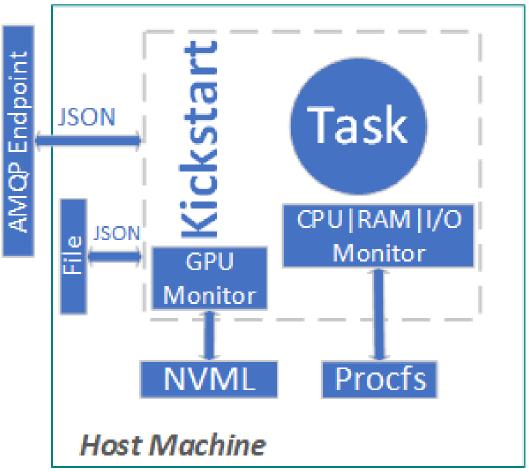

Figure 1. Kickstart online monitoring.

- **kickstart.inv.online.gpu.env:** This event is produced once at the beginning of the job and it contains information about the GPU environment (e.g., number of GPUs, driver version, type of GPUs etc.)
- **kickstart.inv.online.gpu.stats**: This event is produced throughout the execution and it contains a snapshot of the GPU counters at that given time (e.g., GPU utilization, memory usage, power consumption etc.)
- **kickstart.inv.online.gpu.stats.max**: This event is produced at the end of the execution, and it contains max values observed during the run (e.g., max GPU utilization, max GPU temperature etc.)

All of the events are easily correlated with workflow runs and their respective jobs, since they are annotated with workflow related attributes (e.g., workflow uuid, dag job id). In the case of the GPU statistics event, there are some optional fields that are controlled via environment variables. The full list and description of the fields available in the produced events can be found on GitHub [10].

III. How to use

Installation. This tool is available under Pegasus Panorama branch [5] and can be used independently of Pegasus. Precompiled versions of this branch can also be found on the Pegasus download server [11]. Even though this tool is distributed with the Pegasus WMS it is a standalone tool

```
export KICKSTART_MON_URL = \
       rabbitmq://[USERNAME:PASSWORD]@hostname[:port]/api/exchanges/[VIRTUAL_HOST]/[EXCHANGE_NAME]/publish
or
export KICKSTART_MON_URL = file://filename
pegasus-kickstart <args> -G -m 10 ./exec
```

Listing 1: Example invocation of GPU monitoring

and can be installed and used without using the rest of the system. On the Pegasus download server you will find the lightweight worker package that contains Kickstart and other essential Pegasus' tools (e.g., pegasus-transfer), which can be downloaded and installed independently.

Configuration. An example of using GPU-aware Kickstart with Pegasus is the "Predict Future Sales" workflow [12]. It has been configured to use the "–monitoring" flag during workflow generation. This flag instructs Pegasus Panorama to enable GPU monitoring for the jobs requesting GPUs, and orchestrates the data collection via an AMQP point.

To collect GPU traces using Kickstart as a standalone tool, one must set the following flags to "pegasus-kickstart":

- **-m** <interval>: enables online monitoring and collects traces at every <interval>
- **-G**: enables GPU monitoring (Note: this flag is considered only if the -m flag has been provided)

Finally, an environment variable sets the location where the statistics will be published (KICKSTART_MON_URL). Either file or AMQP endpoints can be specified. An example of a standalone invocation can be seen in Listing 1. For more we refer you to the "pegasus-kickstart" documentation [3].

IV. Related and Future Work

Nvidia offers tools for detailed profiling and analysis (e.g., NVIDIA Nsight Tools), which provide in depth analysis of GPU kernels and can aid in debugging and performance optimizations. However, these tools add extra overhead that cannot be tolerated in production and they don't integrate well with other third party tools (e.g., monitoring tools of workflow management systems). Additionally, HTCondor [13] in version 8.8.9 introduced GPU monitoring, but it only offers statistics about the avg. GPU utilization and maximum memory usage, and no tracing is supported. With Kickstart we are able to correlate GPU monitoring traces directly with workflow job executions.

We are currently working on extending the GPU monitoring feature to more devices such as AMD's ROCm GPUs.

Acknowledgments. This work is funded by DOE contracts #DE-SC0012636, #DE-SC0022328 and NSF contract #1664162.

References

[1] E. Deelman, K. Vahi, G. Juve, M. Rynge, S. Callaghan, P. J. Maechling, R. Mayani, W. Chen, R. Ferreira da Silva, M. Livny, and K. Wenger, "Pegasus: a workflow management system for science automation," *Future Generation Computer Systems*, vol. 46, pp. 17–35, 2015.

[2] G. Juve, B. Tovar, R. Ferreira da Silva, D. Krol, D. Thain, E. Deelman, W. Allcock, and M. Livny, "Practical resource monitoring for robust high throughput computing," in *Workshop on Monitoring and Analysis for High Performance Computing Systems Plus Applications*, 2015.

[3] SciTech, "Pegasus Kickstart Documentation," https://pegasus.isi.edu/documentation/manpages/pegasus-kickstart.html?highlight=kickstart.

[4] G. Papadimitriou, C. Wang, K. Vahi, R. Ferreira da Silva, A. Mandal, L. Zhengchun, R. Mayani, M. Rynge, M. Kiran, V. E. Lynch, R. Kettimuthu, E. Deelman, J. S. Vetter, and I. Foster, "End-to-end online performance data capture and analysis for scientific workflows," *Future Generation Computer Systems*, vol. 117, pp. 387–400, 2021.

[5] SciTech, "Pegasus panorama," https://github.com/pegasus-isi/pegasus/tree/panorama.

[6] Pivotal, "Rabbitmq," https://www.rabbitmq.com/.

[7] "ELK stack," https://www.elastic.co/elk-stack, 2018.

[8] L. Foundation, "procfs," https://www.kernel.org/doc/html/latest/filesystems/proc.html.

[9] Nvidia, "NVML," https://docs.nvidia.com/deploy/nvml-api/index.html.

[10] SciTech, "Kickstart gpu events," https://github.com/pegasus-isi/pegasus/blob/panorama/src/tools/pegasus-kickstart/nvidia/README.md.

[11] ——, "Pegasus download server," http://download.pegasus.isi.edu/pegasus.

[12] ——, "Predict Future Sales Workflow," https://github.com/pegasus-isi/predict-future-sales-workflow.

[13] D. Thain, T. Tannenbaum, and M. Livny, "Distributed computing in practice: the condor experience," *Concurrency and computation: practice and experience*, vol. 17, no. 2-4, pp. 323–356, 2005.