Fully Hierarchical Scheduling: Paving the Way to Exascale Workloads

Stephen Herbein
University of Delaware
sherbein@udel.edu

Dong H. Ahn (advisor)
Lawrence Livermore National Laboratory
ahn1@llnl.gov

Michela Taufer (advisor)
University of Delaware
taufer@udel.edu

ABSTRACT

Exascale workloads, such as uncertainty quantification (UQ), represent an order of magnitude increase in both scheduling scale and complexity. Batch schedulers with their decades-old, centralized scheduling model will fail to address the needs of these new workloads. To address these upcoming challenges, we claim that HPC schedulers must transition from the centralized to the fully hierarchical scheduling model. In this work, we assess the impact of the fully hierarchical model on both a synthetic stress test and a real-world UQ workload. We observe over a 100x increase in scheduler scalability on the synthetic stress test and a 37% decrease in the runtime of real-world UQ workloads under the fully hierarchical model. Our empirical results demonstrate that the fully hierarchical scheduling model can overcome the limitations of existing schedulers to meet the needs of UQ and other exascale workloads.

1 INTRODUCTION

The landscape of HPC workloads is quickly changing. Exascale science will include high-throughput workloads currently uncommon to HPC, such as uncertainty quantification (UQ). Uncertainty quantification helps computational scientists characterize and reduce the uncertainty in large-scale simulations by running an ensemble of simulation jobs, where each job in the ensemble has a different set of input parameters. UQ ensembles typically consist of 1,000 to 100,000 jobs but can contain as many as 100,000,000. The larger ensembles can produce up to exaFLOPS of computing load and generate exabytes of data. Running these ensembles presents many challenges due to the limitations of existing HPC batch schedulers (see Table 1 - Column 1). Existing scientific workflow systems, such as the Lawrence Livermore National Laboratory (LLNL) UQ Pipeline (UQP), provide workarounds for common HPC scheduler limitations (see Table 1 - Column 2). These workarounds come with side effects, as described in Table 1 - Column 3. The community is in need of a general solution that goes beyond the workarounds listed in Table 1. We present such a solution in this work by leveraging the fully hierarchical scheduling model.

2 FULLY HIERARCHICAL SCHEDULING UNDER FLUX

Fully hierarchical scheduling is a new HPC scheduling model aimed at addressing next-generation scheduling challenges [1]. The fully hierarchical model applies a divide-and-conquer approach to scheduling, allowing for the distribution of scheduling work across an arbitrarily deep hierarchy of schedulers. This divide-and-conquer approach is applicable at both the system and application levels. This parallelization addresses the challenges of UQ by increasing the scheduler's scalability and enabling more efficient management of job ensembles.

Supercomputing'17, November 2017, Denver, Colorado USA
2017. ACM ISBN .
https://doi.org/

Schedulers' Limitations	Workflow System's Workaround	Side Effects
Max number of jobs	Throttle submissions	Decreased job throughput
Limited job throughput	Aggregate jobs	Increased workload runtime
Lack of job/ensemble status & control API	Track individual jobs' status through files	I/O bottleneck
Lack of programmable failure detection	Inspect failures manually	Unnecessary job resubmissions

Table 1: The limitations of existing schedulers, the common workarounds provided by workflow systems such as the UQP, and the negative side effects of those workarounds.

3 RESULTS

To demonstrate how a fully hierarchical scheduler can overcome the limitations of existing centralized schedulers, we evaluate each scheduler on both a synthetic stress test and a real-world UQP workload.

3.1 Case Study: Synthetic Stress Test

Existing centralized schedulers are limited in the total number of jobs they can manage at a single time. When handling large number of jobs, centralized schedulers exhaust their local resources (e.g., RAM) [3]. As a workaround to this limitation, many workflow systems, such as UQP, throttle their job submissions, artificially reducing the workload's job throughput. The fully hierarchical model distributes local resource requirements across multiple schedulers, enabling it to handle larger numbers of jobs.

To evaluate the effectiveness of our model on a large number of jobs, we first design a synthetic stress test that isolates the overheads associated with scheduling and launching many, short running jobs. Specifically, our stress test consists of many jobs that execute the command 'sleep 0'. Using this stress test, we then compare the centralized and the fully hierarchical models on a 32 node cluster. For the centralized scheduler, there is only one scheduler instance that schedules every job. For the fully hierarchical scheduler, we create a three-level hierarchy and distribute the jobs equally among 1024 schedulers, one on each core. We vary the number of jobs per scheduler between 1 and 2,048, resulting in a total number of jobs between 1 and 1,048,576.

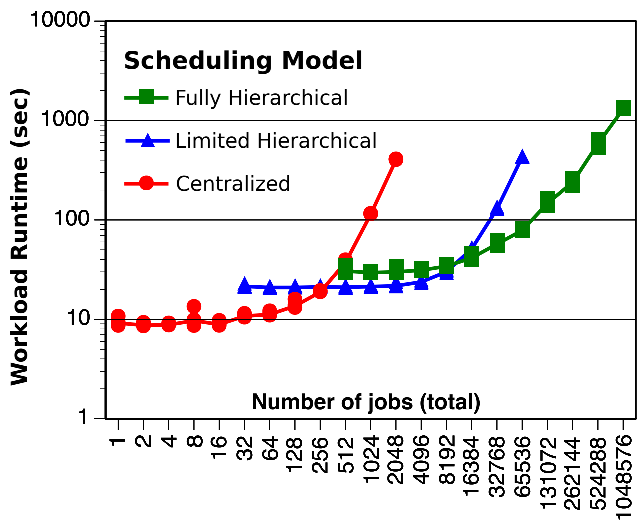

Figure 1: Workload runtime (in seconds) for the centralized and fully hierarchical schedulers running a synthetic HTC workload on a 32-node cluster with different numbers of total jobs.

Figure 1 shows that the fully hierarchical model can launch over 250,000 jobs in the same time it takes centralized scheduling to launch only 2,048 jobs. This confirms that the fully hierarchical model can handle a substantially larger number of jobs, which removes the need for job submission throttling and improves the overall job throughput of the system.

3.2 Case Study: Uncertainty Quantification Pipeline (UQP) Workflow

Existing schedulers lack the job throughput necessary for exascale workloads. To overcome this limitation, many workflow systems, like UQP, combine their individual jobs into fewer aggregate jobs. Under the existing centralized model, sub-jobs within an aggregated job are executed serially, increasing the workflow's runtime. Under the fully hierarchical model, each aggregated job is managed by its own full-feature scheduler, allowing sub-jobs to run concurrently. To examine the benefits of this, we execute the same UQP workflow used in previous UQP studies [2] on a 16 node cluster under both a centralized scheduler (Slurm) and a fully hierarchical scheduler (Flux).

Figure 2 shows up to a 37% faster workflow runtime when switching from a centralized scheduler to a fully hierarchical scheduler due to improved job throughput and resource utilization. This confirms that aggregated jobs can benefit from the full-featured scheduler that the fully hierarchical model provides to every allocation.

4 CONCLUSION AND FUTURE WORK

Current HPC scheduling models have already been outpaced by the challenges posed by the UQP and other workflow systems. With a push towards exascale, the gap between existing schedulers and the workloads they must manage will only grow. The fully hierarchical model provides a single, general solution to many of the challenges presented by exascale HPC workloads. In this work, we have shown that it can increase both the scheduler's maximum number of jobs and the scheduler's job throughput. In future work, we plan to integrate Flux's ensemble status & control API into the UQP to simplify the submission and tracking of the job ensembles. We also plan to develop a programmable failure detection mechanism within Flux to reduce unnecessary resubmissions and simplify error handling for users.

ACKNOWLEDGMENT

The authors acknowledge the advice and support from the both Flux and UQP team at LLNL: Tamara Dahlgren, David Domyancic, Don Lipari, Tom Scogland, Jim Garlick, Mark Grondona, Becky Springmeyer, Chris Morrone, and Al Chu. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-ABS-735978).

REFERENCES

[1] Dong H. Ahn et al. 2014. Flux: A Next-Generation Resource Management Framework for Large HPC Centers. In *Proc. of the 10th International Workshop on Scheduling and Resource Management for Parallel and Distributed System (SRMPDS).*

[2] Tamara L. Dahlgren et al. 2015. Poster: Scaling Uncertainty Quantification Studies to Millions of Jobs. In *Proc. of the 27th ACM/IEEE International Conference for High Performance Computing and Communications Conference (SC).*

[3] John Gyllenhaal et al. 2014. Enabling High Job Throughput for Uncertainty Quantification on BG/Q. In *IBM HPC Systems Scientific Computing User Group (ScicomP'14).* Chicago, IL.

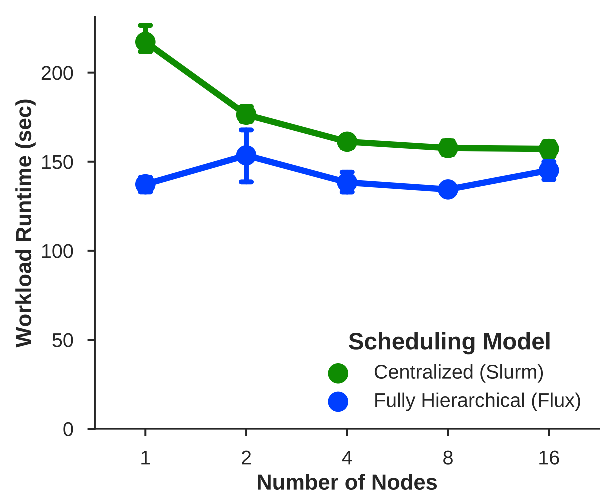

Figure 2: Workflow runtime (in seconds) for the centralized and fully hierarchical schedulers running a real-world UQP workflow on up to 16 nodes.