Exploring the Trade-off Space of Hierarchical Scheduling for Very Large HPC Centers

Stephen Herbein1,2, Dong H. Ahn2, Don Lipari3, Tom Scogland2, Kento Satō2, Jim Garlick2, Mark Grondona2, Becky Springmeyer2, Michela Tauffer1
1University of Delaware, 2Lawrence Livermore National Laboratory

Motivation

• On the path to a next-generation HPC center lies a significant increase in scale and diversity of resources
• Single, monolithic scheduling already lacks scalability and flexibility for today’s large HPC centers
• Hierarchical scheduling is becoming an attractive alternative
  • A lack of trade-off studies precludes the development of effective techniques

Goal

• Quantify the advantages and disadvantages of a hierarchical, multilevel scheduling scheme against a monolithic scheme
• In particular, explore the trade-off space between scheduling complexity and resource utilization

Hierarchical Scheduling under Flux

• Flux is a resource and job management system (RUMS) currently being developed to enable hierarchical, multi-level scheduling for large HPC centers [1]
• Its hierarchical scheduling rules:
  • Parent bounding rule – parent grants and confines the allocation of its children
  • Child empowerment rule – children are solely responsible for the most efficient use of their resources
• Major scalability and policy requirements
  • Provide higher levels of scheduler parallelism and thus scalability
  • Distribute load across the schedulers in this hierarchy
  • Provide the ability to impose a stricter policy enforcement

Realistic Hierarchical Workload Creation

• Under hierarchical scheduling, any job can instantiate a scheduler to schedule its sub-jobs
• A hierarchical workload is non-existent to be used for our trade-off exploration
• Use novel job-aggregation techniques to generate hierarchical workloads from real HPC workloads
  • Jobs submitted within a short window of time with similar characteristics are aggregated together into a larger job
  • This emulates an important mode in which users will use hierarchical schedulers under Flux

Scheduling Complexity Results

<table>
<thead>
<tr>
<th>Machine</th>
<th>Pre-Aggregation</th>
<th>Post-Aggregation</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rzmerl</td>
<td>6,737 jobs</td>
<td>5,688 jobs</td>
<td>1.18x</td>
</tr>
<tr>
<td>Rzzeus</td>
<td>25,309 jobs</td>
<td>7,073 jobs</td>
<td>3.58x</td>
</tr>
</tbody>
</table>

• Introducing only one additional level to scheduling leads to a reduction of 3.58x in scheduling complexity for Rzzeus workloads
  • The preliminary results show each additional level will become a significant reduction factor for scheduling complexity
• Rzmerl has a lower ratio since its submission rate is less
• Lower probability that a user submits jobs back-to-back

Resource Utilization Results

• MinRes results in virtually no idle time and thus no decrease in resource utilization compared to monolithic scheduling
  • Idleness caused by sub-jobs with varying sizes
• MinTime results in a large decrease in utilization compared to monolithic scheduling
  • Idleness caused by sub-jobs with varying runtimes

Conclusion and Future Work

Hierarchical scheduling offers a sizable reduction in scheduling complexity

• Job submission patterns suggest users can group together their jobs to take advantage of hierarchical scheduling
• Larger and busier HPC centers can reduce their scheduling complexity with hierarchical scheduling at deeper levels
• Sub-jobs with diverse time and resource requirements can leave center resources underutilized
• Our study motivates the development of dynamic scheduling as a way to complement hierarchical scheduling

References and Acknowledgements


The authors acknowledge advice and support from the Flux team and others at LLNL: Don Lipari, Tom Scogland, Kento Satō, Jim Garlick, Mark Grondona, and Becky Springmeyer. The authors are grateful to Adam Moody and Todd Gamblin for access to the LLNL cluster job logs.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.