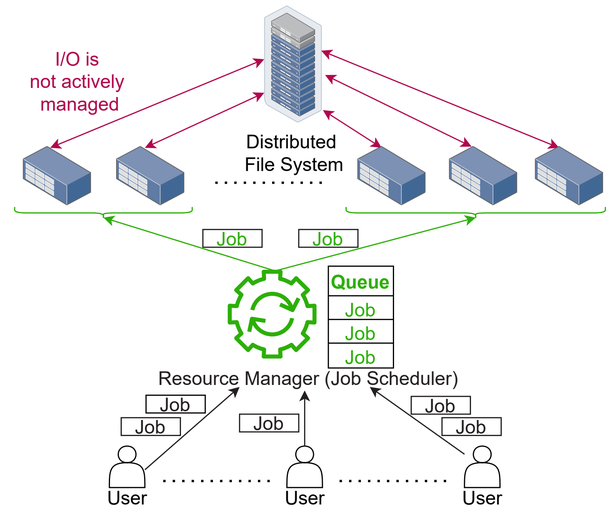
Project Wagomu: Elastic HPC Resource Management

Jonas Posner and Patrick Finnerty

University of Kassel (Germany) jonas.posner@uni-kassel.de | Kobe University (Japan) finnerty.patrick@fine.cs.kobe-u.ac.jp

Motivation

- Hardware resources of today's supercomputers are typically managed in a static way
  - Users submit their programs as jobs, specifying a fixed set of resources and time
  - A resource manager decides when to start which job on which nodes
- This node-oriented, static resource management needs to be addressed
  - ✗ Resource managers are inflexible in scheduling jobs
  - ✗ Programs cannot change their resources to meet their changing needs of different phases
  - ✗ Uncontrolled dynamic changes of resources affect program execution times in unpredictable ways
- ➪ **This project envisions *elastic resource management* via *Asynchronous Many-Task (AMT) programming***
  - ✓ New resources can be integrated into running jobs and resources can be released from running jobs
  - ✓ Resource change requests can be initiated from either the *resource manager side* and *the program side*



Program Side: Asynchronous Many-Task (AMT): Current Status [1, 2, 3, 4]

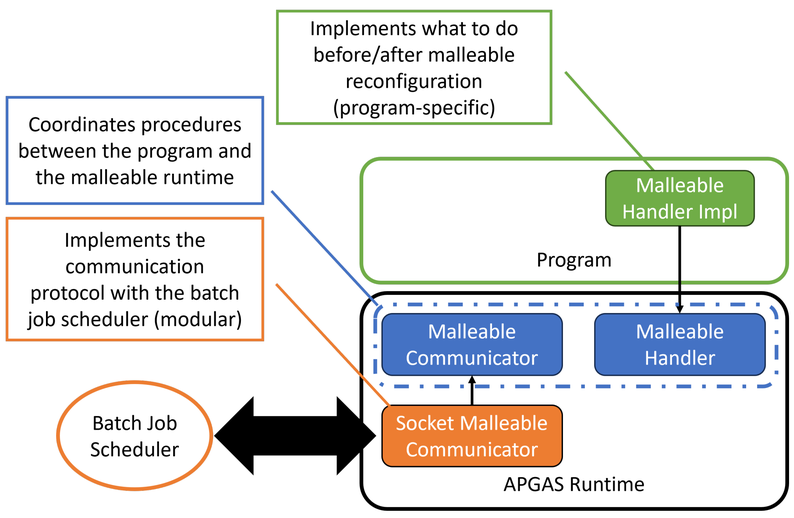


Figure 1: AMT: Elastic architecture [3]

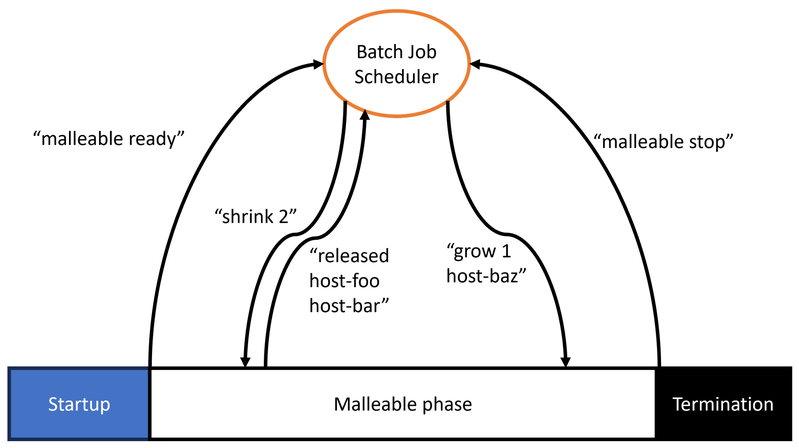


Figure 2: AMT: Lifecycle elastic program [3]

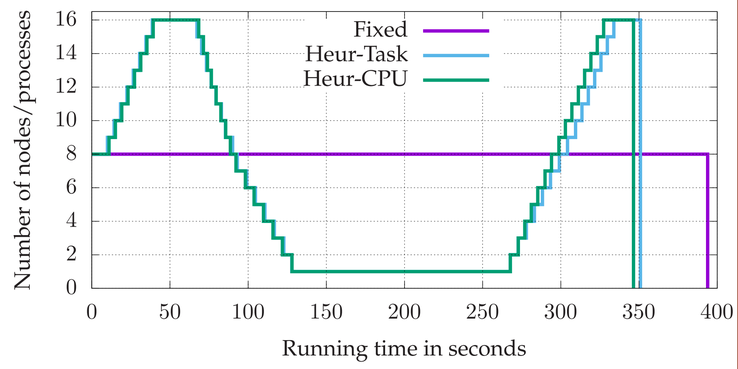


Figure 3: AMT: Evolving resource allotments [4]

Resource Manager Side: Simulations: Current Status [1, 3, 5]

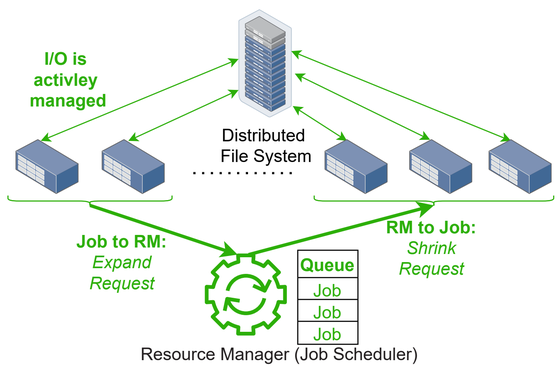


Figure 4: Supercomputer: Elastic architecture

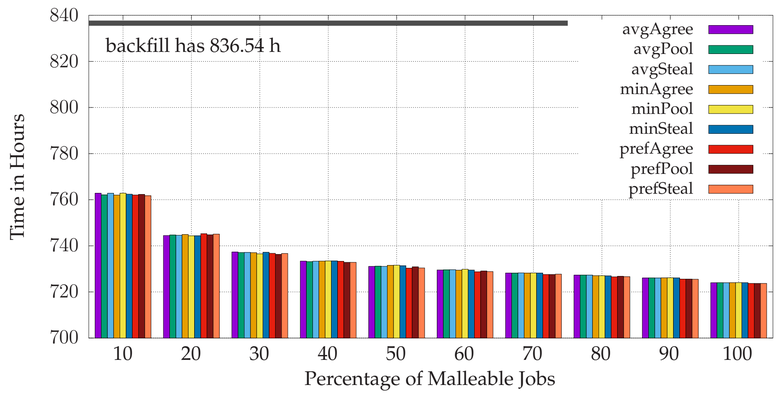


Figure 5: Simulations: Overall completion time [5]

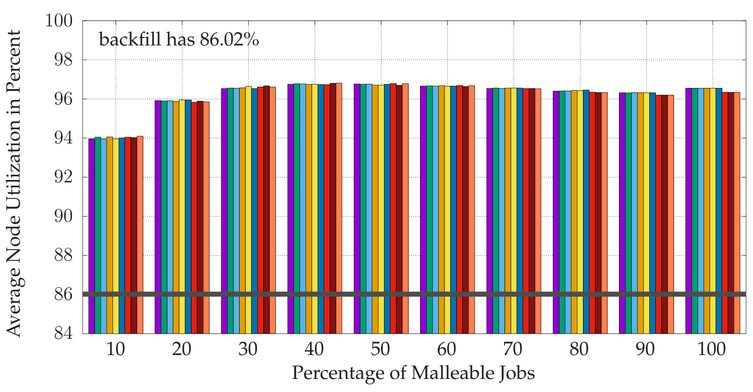


Figure 6: Simulations: Average node utilization [5]

Road Map

- ➪ Elastic Asynchronous Many-Task (AMT) runtime system
  - *Status*: Prototyped with APGAS for Java [2, 4]
  - *Next Step*: Move to C++ using MPI Sessions
- ➪ Elastic job scheduling algorithms
  - *Status*: Evaluated via simulations [5]
  - *Next Step*: Use real supercomputer log traces
- ➪ Interaction between AMT runtime and resource manager
  - *Status*: Prototyped with Java [3]
  - *Next Step*: Extend Slurm supporting elasticity interactions

Project Information

- Aggregation of several (future) projects in Germany and Japan
- Main Project Members:
  - Jonas Posner, University of Kassel, Germany
  - Patrick Finnerty, Kobe University, Japan

1. *Funding Source*: Central Research Fund (ZFF)
   - *Runtime*: 2022 – 2023
2. *Funding Source*: Deutsche Forschungsgemeinschaft (DFG)
   - *Runtime*: 2024 – 2027
   - ? *Status*: Under review

Methodology

- Development of innovative algorithms to enable resource elasticity
- Implementation of prototypical AMT runtimes in Java and C++
- Evaluations via both simulations and real-world experiments
- Improvement and combination of the techniques step-by-step

References

[1] J. Posner and P. Finnerty. *Project Wagomu*. 2024. URL: https://github.com/ProjectWagomu.

[2] P. Finnerty, R. Takaoka, T. Kanzaki, and J. Posner. "Malleable APGAS Programs and their Support in Batch Job Schedulers". In: *Euro-Par AMTE*. 2023. DOI: 10.1007/978-3-031-48803-0_8.

[3] P. Finnerty, J. Posner, J. Bürger, L. Takaoka, and T. Kanzaki. "On the Performance of Malleable APGAS Programs and Batch Job Schedulers". In: *SNCS*. 2024. DOI: 10.1007/s42979-024-02641-7.

[4] J. Posner, R. Goebel, and P. Finnerty. "Evolving APGAS Programs: Automatic and Transparent Resources Adjustments at Runtime". In: *WAMTA*. 2024. *To appear*.

[5] J. Posner, F. Hupfeld, and P. Finnerty. "Enhancing Supercomputer Performance with Malleable Job Scheduling Strategies". In: *Euro-Par PECS*. 2023. DOI: 10.1007/978-3-031-48803-0_14.