Parallel Task Scheduling in Heterogeneous Computing Environments

Christopher Blandin and Dylstan Machovec
Post Doctoral Research Assistant: Dr. Bhavesh Khemka
Advisor: Professor H. J. Siegel

Introduction

- In High Performance Computing (HPC), large scale parallel applications are used to solve various scientific problems.
- Often HPC systems are heterogeneous environments with different types of processors.
- Due to their scale, scientific applications often require numerous nodes.
- Mapping parallel applications to these nodes is an NP-hard problem.
- Heuristics are often used to find solutions.
- Our environment is comprised of a set of heterogeneous clusters.
- Each cluster has homogeneous nodes.
- Each task operates within a cluster and does not share nodes.

Problem Statement

Task 2
Task 3
Task 4
Task 5
Task 6
Task 7
Task 8

Dynamic Heuristics

- We implemented a number of heuristics to solve this problem for dynamically arriving tasks.
- These heuristics are given the following:
  - Tasks that have arrived up until that time
  - Execution time for each task within each cluster
  - Number of cores needed for execution
  - Value function of each task
- The resources that a task uses are the product of:
  - The number of cores occupied by the task
  - The execution time of that task

Conservative Backfilling

- Assigns tasks in a first-come-first-serve (FCFS) order such that they are the earliest possible available start time.
- If the task cannot start execution immediately, then its earliest possible start time is reserved.
- Future tasks cannot violate this reservation.

Conservative Multiple Queues

- Tasks are separated into three queues based on the average amount of resources that they use.
- Tasks in each queue are arranged in FCFS order.
- Tasks are assigned using conservative backfilling where the ordering is based on cycling through the queues in a round-robin fashion.

Max Value

- Determines the value each task can earn for the earliest assignment of nodes in each cluster.
- Schedules the task with the maximum value until all tasks have been scheduled.
- If a task cannot start execution immediately, then a permanent reservation is made.

Max Value-Per-Resource (VPR)

- Same as Max Value except VPR is the criterion.

Concept of Place-Holders

- Temporary reservations that restrict the scheduling of tasks for one heuristic iteration.
- We implemented variations of Max Value and Max VPR with place-holders to provide more flexibility for later-arriving tasks.

Genetic Algorithm

- A global search technique that mimics evolution.
- A postmortem static approach to mapping tasks to clusters.
- A better comparison than our mathematical upper bound of maximum possible value earned.
- Steps of a genetic algorithm:
  - For N generations
    - Perform crossover operation
    - Perform mutation operation
    - Perform selection operation
    - Repeat

Chromosome

- Each chromosome represents a mapping of tasks.
- The chromosome’s fitness is the mapping’s value.
- Genes in chromosomes represent tasks and contain a cluster ID.
- The ordering of genes in a chromosome is the order in which tasks are assigned to the clusters.
- Seeded with offline versions of our heuristics.

Crossover

- Choose two random chromosomes.
- Choose two points to crossover and swap genes between the two points.
- The center section contains the remaining genes from the parent.

Mutation

- For each chromosome that is selected for mutation, check each gene to mutate.
- For each gene that is selected for mutation, we both swap order with a random gene and change the cluster assignment.

Selection

- The best chromosome is always in next generation’s population (elitism).
- Select chromosomes for next generation’s population using a rank-based roulette wheel.

\[
 f(\text{rank}) = 2 - SP + 2(SP - 1) \times \frac{\text{rank}}{N - 1}, \quad 2 \geq SP \geq 1
\]

Dynamic Simulation Results

- Arrival trace information from the Curie supercomputer in France from Dror Feitelson’s Parallel Workloads Archive.
- Traces are used to generate the task arrival patterns and execution times for one cluster in our simulations.
- Execution times of other clusters were generated using the Coefficient of Variation (CV) method from prior work.
- Value functions were initially generated using a gamma distribution such that there is a weak correlation between average task execution time and value earned by a task.
- Increase or decrease the oversubscription of the system by adjusting the total number of cores in the system.
- The same inputs are used for both the dynamic heuristics and the genetic algorithm.

The dynamic results will appear in…”