Job Brokering in Cycle-Stealing Systems with BoT’s Workloads: A Mean-Field Approach

Bruno Gaujal, Nicolas Gast
Bahman Javadi, Derrick Kondo, Jean-Marc Vincent
Job Brokering Problem

A Users

\[ a_t^1 Y_t \]

\[ a_t^2 Y_t \]

\[ a_t^d Y_t \]

\[ Y_t: \text{number of tasks sent between } t, t+1 \]

\[ a_t^i: \text{probability broker sends tasks to cluster } C_i \]

\[ P_i: \text{number of processors in cluster } C_i \text{ all of speed } u_d \]

Each processor j either available or not.

\[ B_{t+1}^i: \text{size of queue } i: (B_t^i - u_i X_t^i + a_t^i Y_t)^+ \]

Goal: minimize waiting time: \( \sum_i B_t^i \)
Approach and Key Result [Gaujal et Gast]

- Focus on limit behavior of a Markovian Decision Process (MDP)
- Show MDP converges to deterministic limit
  - Can compute optimal policy in deterministic system efficiently
  - Can apply policy in random system with good approximation of optimal policy
- Speed of convergence $O(\sqrt{\text{number of entities}})$
Experimental Problem

• Goal
  • Show effectiveness of mean-field approach to real cycle-stealing systems with real workloads
  • Workload: Bag of independent tasks
  • Resources: BOINC or KOALA
Approach

Markov Modelling

Experimental Validation

Less real

Simplest (Exponential)

Simulation

Emulation

Real

More real

Complex (Gamma, MMPP)
Issues

• Modelling
  • Workload, Resource availability
• Simulation and emulations
• Decide on or build framework
Workload Models [Iosup]

- Task
  - Task size (num procs)
  - Task runtime
  - Requested time
- BoT
  - Arrival Rate
  - BoT size
Resource Availability [Javadi]

- Availability intervals
- Unavailability intervals

BOINC

KOALA?

Gamma, hyperexponential
Log-normal, Gamma, perhaps MMPP
Experimentation Tool Requirements

- Simulation using traces of workloads (users, batches, tasks) and resource failures
- Emulation using traces of jobs and failures
- Emulation of local job schedulers (not always FIFO)
- Seemlessly go from simulation to emulation to real-world
- Must scale well (at least 30,000 resources)
Candidates

- Simulator
  - SimGrid
- DGsim
- Emulator
- Wrekhavoc
- EmuBOINC
- Simulator / Middleware
  - PyMW (+ Wrekhavoc + EmuBOINC)
## Comparison

<table>
<thead>
<tr>
<th>Tool</th>
<th>Experiment Type</th>
<th>Workload</th>
<th>Failures</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimGrid</td>
<td>Sim</td>
<td>Y</td>
<td>Y</td>
<td>work in prog</td>
</tr>
<tr>
<td>DGSim</td>
<td>Sim</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EmBOINC</td>
<td>Emu</td>
<td>high-tp</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>WrekHavoc</td>
<td>Emu</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PyMW</td>
<td>Sim, Real</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

Option: use PyMW (+ WrekHavoc + EmBOINC)
PyMW

- Middleware for Master-Worker applications
- Provides interfaces for different computation environments (simulation, multi-core, BOINC, Condor)
- Can switch from simulation to real execution by changing the interface (a single line of code)
- Simple user interface
  - Call at most 5 functions (Interface, PyMW(Interface), master.submit_task(), master.get_result(), master.get_status())
- Simple implementation (2500 lines of code, 1400 lines for 7 interfaces)
- Creator is collaborator and will hire as likely post-doc in MESCAL (different project)
Applying PyMW in Simulation

Generator-Driven Users

Broker: PyMW Master

Sim interface $C_1$

Sim interface $C_2$

Sim interface $C_d$

Generator-Driven Resource Availability

$u_1$

$u_2$

$u_d$
Applying PyMW in Simulation II

Trace-Driven Users

Broker: PyMW Master

Sim interface C₁

Sim interface C₂

Sim interface Cₙ

Trace-Driven Resource Availability
Applying PyMW in Emulation

Trace-driven users submitting simulated jobs

Broker: PyMW Master

Trace-Driven Resource Availability

BOINC interface C1 EmBOINC

BOINC interface C2 EmBOINC

BOINC interface Cd EmBOINC
Applying PyMW in Emulation

Trace-driven users submitting CPU-intensive jobs

Broker: PyMW Master

Trace-Driven Resource Availability

- BOINC interface $C_1$ (BOINC)
- BOINC interface $C_2$ (BOINC)
- BOINC interface $C_d$ (BOINC)

- Wrekhavoc/G5k
  - $u_1$
  - $u_2$
  - $u_d$
New Post-Doc

- PhD from Seoul University
- Starts October 26th
Other Points of Collaboration

- Cost-Benefit Analysis of Grids versus Clouds
- Characterizing and predicting idleness in Grids
Approach

Markov Modelling

Experimental Validation

Entities (servers, hosts, middleware)

Events

Less real

Simplest (Exponential)

Simulation

Generator-driven

Trace-driven

Emulation

More real

Complex (Gamma, MMPP)

Real

Real software, hardware