Volunteer Computing in the Clouds

Artur Andrzejak\textsuperscript{1}, Derrick Kondo\textsuperscript{2}, Sangho Yi\textsuperscript{2}

\textsuperscript{1}Zuse Institute Berlin, but now at Institute for Infocomm Research (I2R), Singapore

\textsuperscript{2}INRIA Grenoble, France
Trade-offs

Cost ($)

Reliability

Cloud Computing

Performance

Cost ($)

Cost ($)

Cloud Computing

Reliability

High

Low

Supercomputers

Clusters

Volunteer Computing
Market-based Resource Allocation Systems

- Amazon Spot Instances
- “Spot” instance price varies dynamically
- Spot instance provided when user’s bid is greater than current price
- Spot instance terminated when user’s bid $\leq$ current price
- Amazon charges by the last price at each hour

### Synthetic Example:

#### Real Amazon Price Trace:

- Time (January 11-18, 2016) - eu-west-1.mux.m1.medium
- Time (January 11-18, 2016) - eu-west-1.mux.m2.xlarge
- Time (January 11-18, 2016) - eu-west-1.mux.m2.lARGE

cloudexchange.org [tim lossen]
Optimization Problem

• Given job with batch of parallel, independent, divisible tasks

• Deadline and budget constraints

• Objectives
  • Can the job be executed under budget and deadline constraints?
  • What is the bid price and instance type that minimizes the total monetary costs?
  • What is the distribution of monetary costs and execution times for a specific instance type and bid price?
Goal and Approach

• Formulate and show how to apply user decision model
• Characterize relationship between job execution time, monetary cost, reliability, bid price
• Compare costs of different instance types
Outline

• System model
• Decision model
• Simulations method and results
• Relation with BOINC
• Conclusion & Future work
User Parameters and Constraints

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_{inst}$</td>
<td>number of instances that process the work in parallel</td>
</tr>
<tr>
<td>$n_{max}$</td>
<td>upper bound on $n_{inst}$</td>
</tr>
<tr>
<td>$W$</td>
<td>total amount of work in the user’s job</td>
</tr>
<tr>
<td>$W_{inst}$</td>
<td>workload per instance $(W/n_{inst})$</td>
</tr>
<tr>
<td>$T$</td>
<td>task length, time to process $W_{inst}$ on a specific instance</td>
</tr>
<tr>
<td>$B$</td>
<td>budget per instance</td>
</tr>
<tr>
<td>$c_B$</td>
<td>user’s desired confidence in meeting budget $B$</td>
</tr>
<tr>
<td>$t_{dead}$</td>
<td>deadline on the user’s job</td>
</tr>
<tr>
<td>$c_{dead}$</td>
<td>desired confidence in meeting job’s deadline</td>
</tr>
<tr>
<td>$u_b$</td>
<td>user’s bid on a Spot Instance type</td>
</tr>
<tr>
<td>$I_{type}$</td>
<td>EC2 instance type</td>
</tr>
</tbody>
</table>

**Notation legend:**
- **Green:** Job parameters
- **Blue:** Job constraints
- **Red:** User decision variables
Random Variables of Model

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ET$</td>
<td>execution time of the job (clock time)</td>
</tr>
<tr>
<td>$AT$</td>
<td>availability time (total time in-bid)</td>
</tr>
<tr>
<td>$EP$</td>
<td>expected price, i.e. (cost per instance)/$AT$</td>
</tr>
<tr>
<td>$M$</td>
<td>monetary cost $AT \cdot EP$ per instance</td>
</tr>
<tr>
<td>$AR$</td>
<td>availability ratio $AT/ET$</td>
</tr>
<tr>
<td>$UR$</td>
<td>utilization ratio $T/ET$</td>
</tr>
</tbody>
</table>

- **Green**: performance
- **Red**: reliability
- **Blue**: monetary cost
Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Execution Model Example

Figure 1: Illustration of the execution model and computation of the random variables. (a) Parallel and independent bags of tasks. (b) The majority of tasks are submitted simultaneously and require different times to complete.

Time (hour)

Price

Availability (5)

Failure (2)

Availability (3)

Useful computation (4)

Chpt (1)

Restart (1)

Useful comp. (2)

Notation

\( T = 6 \text{h} \)

\( ET = 10 \text{h} \)

\( AT = 5 + 3 = 8 \text{h} \)

\( EP = 1.4/8 = 0.175 \text{USD/h} \)

\( M = 3 \times 0.1 + 4 \times 0.2 + 1 \times 0.3 \)

\( = 1.4 \text{USD} \)

\( AR = 8/10 = 0.8 \)

\( UR = 6/10 = 0.6 \)

Table II: The specific Grid and Desktop Grid workload parameters. The table is divided into two parts: (1) the required instance types, (2) the amount of work requested, (3) the mean task length, and (4) the mean job deadline. The workload is reflective of the platform size, with desktop grids having on the order of tens to hundreds of thousands of resources.
Submission with job parameters, and time and budget constraints

Broker applying decision model

Feasible?

No, revise constraints

Yes, get bid to achieve lowest cost or execution time, then deploy.

Amazon EC2 Spot Market
Decision Model

• For a random variable, $X$, we write $X(y)$ for $x$ s.t. $\Pr(X < x) = y$.
  
  • E.g. $ET(0.50)$ is the median execution time

• Feasibility decisions
  
  • Deadline constraint achievable with confidence
    $c_{\text{dead}} \iff t_{\text{dead}} \geq ET(c_{\text{dead}})$
  
  • Budget constraint achievable with confidence
    $c_B \iff B \geq M(c_B)$

• Among the feasible cases, we choose the one with the smallest $M(c_B)$ or lowest execution time $ET(c_{\text{dead}})$
Outline

• System model
• Decision model
• Simulations method and results
• Relation with BOINC
• Conclusion & Future work
Simulation Method

- Determine distributions of model variables via price trace-driven simulation
- Prices: trace of Spot instance prices obtained from Amazon
- Workload model
  - W1: “Big”, based on Volunteer Computing, parameters derived from BOINC catalog
  - W2: “Small”, based on Grids, parameters derived from the Grid Workload Archive

<table>
<thead>
<tr>
<th>Workload</th>
<th>$I_{type}$</th>
<th>$n_{max}$</th>
<th>$W_{inst}$</th>
<th>$T$</th>
<th>$t_{dead}$</th>
<th>$c_{dead}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>2.5GHz</td>
<td>20,000</td>
<td>11.5</td>
<td>4.6h</td>
<td>9d</td>
<td>0.9</td>
</tr>
<tr>
<td>W2</td>
<td>2.5GHz</td>
<td>50</td>
<td>6.83</td>
<td>2.7h</td>
<td>17.9h</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Distribution of Execution Time and Costs (Instance Type A and Workload W1)

(b) when task length $T = 276$ minutes

$\Pr (ET <= 4800m) = 0.90$ with bid of 0.082

$\Pr (M <= 0.38) = 0.90$ with bid of 0.076

Possible range of bid prices $B$ and $c_B$

$t_{\text{dead}}, c_{\text{dead}}$: high-pass filter

$B, c_B$: low-pass filter

(b) $T = 276$
Relation to BOINC?

• Amazon does not provide any middleware for Spot instances

• BOINC is ideal as it handles nondeterministic failures, and ongoing work with VM integration would allow transparent checkpointing

• Use BOINC with decision model to be cost-aware

  • Cloud-enabled BOINC client or server?

  • Integrate with volunteers on the Internet, Grids etc?
Why not just use Internet volunteers?

- Reliability of Spot Instances is tunable (at a cost)
- Greater inter-node connectivity + higher bandwidth
  - ~1 Gbit among EC2 instances*.
  - ~100 Mbit down/55 Mbit up between EC2 and S3*
- Scientific data can be hosted on Amazon for free

Hybrid Use Case

- Scientist submit 10,000 jobs
- Last 7%* are stragglers and delay job completion
- Run last 700 jobs on Amazon Spot Instances in parallel all at once
  - Spot instance cost: ~$210 ± $20
  - Could be cheaper if use reliable host mechanism
- Tune reliability according to budget and time constraints of user

* Personal communication with Kevin Reed
Implementation Approach*

- Distinguish BOINC cloud nodes
  - Create accounts with special id
- Schedule on cloud nodes
  - Use matchmaking function is_wu_feasible_custom?
- Prioritize work units later in batch
  - Use feeder to prioritize by result_id or priority

* Thanks to David Anderson
Discussion Questions

• Would application scientists use hybrid volunteer computing / cloud platforms?

• Accounting model?

• Would volunteers use cloud platforms?

• Would hybrid system allow for new types of applications in terms of data intensity or message passing?
Plug

• EU project
  • European Desktop grid Initiative (EDGI)
  • Open 2-year post-doc in Lyon
Thank you