Near-Optimal Adaptive Control of a Large Grid Application

Det Buaklee
Greg Tracy        Mary Vernon    Steve Wright

Computer Science Department
University of Wisconsin - Madison
Talk Outline

- Condor
- Stochastic Optimization, ATR
- ATR Execution Time Analysis
- Model for Minimum Execution Time
- Results: Optimized ATR Performance
Condor

- Provides high throughput computation
- Manages a heterogeneous & dynamic pool
- MW layer supports Master-Worker applications
  - Submitting node is the “master” node
  - Condor dynamically allocates “worker” nodes
  - Worker nodes can drop out during computation (min,max)

<table>
<thead>
<tr>
<th>Application</th>
<th>MW Layer</th>
<th>Condor</th>
<th>PVM/TCP</th>
</tr>
</thead>
</table>

Communication Link
Stochastic Optimization

• Non-trivial ~ 10,000 lines + LP codes

• Optimization of a model with uncertain data
  – Large number of possible scenarios for the data
  – Arises in planning-under-uncertainty applications

• \( x \): vector of variables (unknowns)
  – aim to find the \( x \) that optimizes expected model performance over all the scenarios

• Objective function is an expectation \( Q(x) \)

\[
\min_x c^T x + Q(x) \text{ subject to } A x = b, \ x \geq 0
\]
Properties of Expectation $Q(x)$

- Probabilistic weighted sum over the objective for each individual scenario $\omega_i$, $i=1,2,...N$

$$Q(x) = \sum_{i=1}^{N} p_i Q(x; \omega_i)$$

- $N$ is number of scenarios evaluated
  - Maybe *sampled* from the full set of scenarios
  - Increase $N$ to improve the accuracy
For each Iteration

master

workers

N = 16 = number of scenarios evaluated
G = 4 = number of task groups
T = 8 = number of tasks per iteration
Goals

Given N and a set of workers:

- Compute (near)optimal adaptive values of B, G, T
  - Automated process
  - Fast/simple runtime computation

- Compare adaptive and non-adaptive B, G, and grouping/scheduling of tasks

Approach: LogP/LogGP/LoPC model
ATR in parallel

- Each task $i$ returns value of $\sum_i Q(x; \omega)$, and a subgradient (slope) for this partial sum.
- Sum over tasks to obtain complete function $Q(x)$ and its subgradient.

- At the end of each iteration, set new $x$ to be minimizer of the latest approximation to $Q(x)$.
Execution Time Analysis

Measure LogP/LogGP/LoPC model parameters

- \( L \) (network latency)
- \( o \) (message processing overhead)
- \( G \) (gap per byte - Bandwidth)
- \( P \) (number of Processors)

master execution time
worker execution time
communication time
### Execution Time Measurement

<table>
<thead>
<tr>
<th>G</th>
<th>T</th>
<th>Worker Execution Time (sec)</th>
<th>Master Time to Update Model Function ( m(x) ) (msec)</th>
<th>Master Time to Compute a New Iterate, ( x ) (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>( \text{avg} )</td>
<td>( \text{max} )</td>
</tr>
<tr>
<td>25</td>
<td>2</td>
<td>13.27</td>
<td>0.05</td>
<td>3.33</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>6.25</td>
<td>0.03</td>
<td>2.25</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>3.41</td>
<td>0.05</td>
<td>1.57</td>
</tr>
<tr>
<td>200</td>
<td>200</td>
<td>7.60</td>
<td>0.05</td>
<td>2.42</td>
</tr>
<tr>
<td>400</td>
<td>400</td>
<td>2.06</td>
<td>0.01</td>
<td>1.32</td>
</tr>
</tbody>
</table>

- One master and one worker experiment
- High variability
Worker Execution Times

- For a given planning problem $t_w$ is linear in
  - Number of scenarios evaluated
  - Processor speed

Total worker time = $n(t_w)^{\text{max}}$
Updating $m(x)$ after each task group ($G$) returns

- **Variability in execution time due to:**
  - Excessive default debug I/O
  - Interference from Condor administrative tasks
- **Eliminating both makes this execution time <1ms**
  i.e., negligible
Hard to make prediction for the next iterate
Same characteristic for all planning problem

SSN network design problem
20term problem
Generating new $x$ at the end of each iteration:

- Number of iterations ($n$) and time to compute $x$ for each iteration depends on $N$, $T$
- Given $N$, total master processing time ($t_M$) is fixed!

Optimize: $T$ is large, but not too large
Communication Costs

- Round trip time measurement
- Critical path contains one round trip time per iterate
- Round trip time $<<$ worker execution time
  for message sizes used in ATR (250–1200 bytes)
Effect of Basket Size

- More iterations (n) needed for larger B – approximately linear relationship of B and n
- Optimal B=1
Model Vocabulary

\( N \) number of scenarios in model

\( T \) number of tasks per iteration

\( G \) number of groups of scenarios (units of work)

\( B \) number of vectors \( x \) evaluated in parallel

\( t_M \) total master execution time

\( t_W \) individual worker execution time

\( n \) total number of iterations
Building the Model

Master, Worker, Communication Times

• Total master execution time
  – Variable for N, T, B
  – Include only time to generate new $x$

• Worker execution time per iteration:
  – Very low variation
  – Consistent from one iteration to another

• Insignificant contributions from:
  – Communication time
  – Master updating $Q(x)$
  – (if T not too large)

$$t_M + n(t_w)_{\text{max}}$$
# Model Validation for Homogenous Worker Pool

<table>
<thead>
<tr>
<th>Planning Problem</th>
<th>N</th>
<th>T</th>
<th>Compute New $x$ (sec)</th>
<th>Benchmark Average $t_W$ (sec)</th>
<th>Total Execution Time (min)</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>num it. (n)</td>
<td>Total ($t_M$)</td>
<td></td>
<td>Model</td>
</tr>
<tr>
<td>20-terms</td>
<td>5,000</td>
<td>200</td>
<td>597</td>
<td>2762</td>
<td>2.35</td>
<td>69.4</td>
</tr>
<tr>
<td>ssn</td>
<td>40,000</td>
<td>100</td>
<td>84</td>
<td>297</td>
<td>30.97</td>
<td>48.8</td>
</tr>
<tr>
<td>ssn</td>
<td>20,000</td>
<td>50</td>
<td>108</td>
<td>180</td>
<td>20.91</td>
<td>40.8</td>
</tr>
<tr>
<td>ssn</td>
<td>20,000</td>
<td>100</td>
<td>84</td>
<td>244</td>
<td>20.89</td>
<td>33.5</td>
</tr>
<tr>
<td>ssn</td>
<td>20,000</td>
<td>200</td>
<td>61</td>
<td>295</td>
<td>20.88</td>
<td>26.4</td>
</tr>
<tr>
<td>ssn</td>
<td>20,000</td>
<td>400</td>
<td>44</td>
<td>441</td>
<td>20.96</td>
<td>22.9</td>
</tr>
<tr>
<td>ssn</td>
<td>10,000</td>
<td>100</td>
<td>44</td>
<td>64</td>
<td>6.32</td>
<td>10.3</td>
</tr>
</tbody>
</table>

**Model**: $t_M + nt_W$
# Model Validation for Heterogeneous Worker Pool

## Table

<table>
<thead>
<tr>
<th>Computing new X (sec)</th>
<th>Worker Time (sec)</th>
<th>Non Adaptive Execution Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>avg.</td>
<td>$t_w^{\text{min}}$</td>
</tr>
<tr>
<td>$n$</td>
<td>$t_M$</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>50.37</td>
<td>7.04</td>
</tr>
<tr>
<td>70</td>
<td>50.02</td>
<td>7.03</td>
</tr>
<tr>
<td>58</td>
<td>35.8</td>
<td>6.62</td>
</tr>
<tr>
<td>42</td>
<td>60.71</td>
<td>2.86</td>
</tr>
<tr>
<td>38</td>
<td>53.18</td>
<td>2.76</td>
</tr>
<tr>
<td>36</td>
<td>46.77</td>
<td>2.86</td>
</tr>
<tr>
<td>36</td>
<td>61.3</td>
<td>2.11</td>
</tr>
</tbody>
</table>

## Model

$$\text{Model: } t_M + n(t_w)^{\text{max}}$$
Optimal Configuration for Homogenous Worker Pool

- **G** should be equal to number of available processors
- **T** should be large up to a point
- **B** should be set to 1

### Table: Original ATR Execution Time (T = 100, G = 25)

<table>
<thead>
<tr>
<th>Reduced Debug</th>
<th>Default Debug</th>
<th>Near-Optimize ATR Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>B=3</td>
<td>B=6</td>
<td>B=3</td>
</tr>
<tr>
<td>61 min</td>
<td>92 min</td>
<td>68 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B=6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>149 min</td>
</tr>
</tbody>
</table>

- **3x – 6x faster!**
Heterogeneous task assignment

**master node’s worker queue**

- **benchmark:** 9 9 10 13 15 20 20 20
- **$E_w$:**
  - 27 27 20 26 30 40 20 20

**master node’s job queue per iteration**

- 1 2 3 4 5 6 7 8
## Adaptive task assignment

<table>
<thead>
<tr>
<th>Worker Time ($t_w$) (sec)</th>
<th>Original task assignment</th>
<th>Adaptive task assignment</th>
<th>Estimated Speedup (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Execution Time (min)</td>
<td>Execution time (min)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Model</td>
<td>Measured</td>
<td>Model</td>
</tr>
<tr>
<td>avg</td>
<td>min</td>
<td>max</td>
<td>Model</td>
</tr>
<tr>
<td></td>
<td>4.02</td>
<td>1.69</td>
<td>8.9</td>
</tr>
<tr>
<td></td>
<td>7.76</td>
<td>2.58</td>
<td>19.4</td>
</tr>
<tr>
<td></td>
<td>2.25</td>
<td>1.20</td>
<td>9.5</td>
</tr>
<tr>
<td></td>
<td>2.84</td>
<td>0.83</td>
<td>9.6</td>
</tr>
<tr>
<td></td>
<td>20.33</td>
<td>8.06</td>
<td>98.6</td>
</tr>
</tbody>
</table>

- Heterogeneous & dynamic worker pool
- Better utilization of worker node
Conclusion

• Analysis of Grid Application Execution Time
• Construct, Validate a Simple Performance Model
• Create an Adaptive Control scheme guided by our Performance Model
• Optimal adaptive parameter gives large speedup (3x-6x) over original ATR code
• Adaptive task assignment gives 15-55% speedup over original policy, for optimal parameter values
Future Work

- Apply the model to larger data sets
- Apply the model to more complex objectives such as controlling processor utilization
- Apply this model to other grid applications
Acknowledgments

- Jeff Linderoth (ATR)
- Jichuan Chang (MW)
- condor-admin@cs.wisc.edu
Question!?
Stochastic Optimization Example

- First month data: Demand 10 units, Price $1.00/unit, Storage cost $0.05/unit
- Possible second month scenarios:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Prob.</th>
<th>Demand</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>0.50</td>
<td>10.0</td>
<td>1.00</td>
</tr>
<tr>
<td>Warm</td>
<td>0.30</td>
<td>08.0</td>
<td>0.85</td>
</tr>
<tr>
<td>Cold</td>
<td>0.15</td>
<td>14.0</td>
<td>1.50</td>
</tr>
<tr>
<td>Very cold</td>
<td>0.05</td>
<td>27.0</td>
<td>1.80</td>
</tr>
</tbody>
</table>
ATR

- “Asynchronous Trust-Region” algorithm for minimizing $Q(x)$ subject to the constraints
- Iterative fork-join synchronization structure
- Unpredictable number of iteration to converge
- Adjustable task parameter
- 15,000 lines of code
Even More Parallelism!

- Possibly generate new $x$ before all $Q(x; \omega_i)$ return!
- Now only have partial info about $Q(x)$, so expect lower quality estimates of $x$
- Example:

$$Q(x)$$

![Diagram showing master and workers with $Q(x_1; \omega_3)$ and $Q(x_1; \omega_2)$]
ATR Vocabulary

N number of scenarios in the model (possible values for the uncertain data)
   e.g., \( N = 5,000 \) or \( N = 40,000 \)

G number of groups of scenarios (units of work)
   e.g., \( G = 50 \) or \( G = 100 \)

T number of tasks in iteration
   e.g., \( T = 200 \) or \( T = 1,000 \)

B number of variables \( x \) evaluated in parallel
   e.g., \( B = 5 \)
Adaptive Control Algorithm

- Sorting the worker list based on benchmarks
  - Benchmark = execution time of a sample task group on this worker
  - Indicates the expected time needed for this worker to complete one task group

- For each worker \( w \), define
  \[
  E_w = (\# \text{ currently assigned tasks}_w + 1) \times \text{benchmark}_w
  \]

- New task will be assigned to the worker with lowest \( E_w \)