Speedup for Multi-Level Parallel Computing

Shanjiang Tang, Bu-Sung Lee, Bingsheng He

School of Computer Engineering
Nanyang Technological University

21st May 2012
OutLine

• Background & Motivation
• Multi-Level Parallel Speedup
• Evaluation
• Conclusion
Multi-Level Computing Architecture and Paradigm
Multi-Level Computing Architecture and Paradigm

- MPI+OpenMP
- MPI+CUDA
- MPI+OpenMP+CUDA

…..
Multi-Level Parallel Computing Model
Parallel Speedup

• Definition

\[ Speedup = \frac{Sequential\text{ExecutionTime}}{Parallel\text{ExecutionTime}} \]

• Classification

➢ Absolute Speedup

\[ Speedup = \frac{BestSequential\text{ALGExecutionTime}}{Parallel\text{ALGExecutionTime}} \]

➢ Relative Speedup

\[ Speedup = \frac{Parallel\text{ALGSequentialExecutionTime}}{Parallel\text{ALGExecutionTime}} \]
Relative Speedup Model

- **Fixed-size Speedup**
  - **Amdahl’s Law**
    \[
    \text{Speedup} = \frac{\text{sequentialTime}}{\text{parallelTime}} = \frac{1}{1 - \alpha + \frac{\alpha}{p}}
    \]
    where $\alpha$ is parallel fraction workload of the program, $p$ is the number of processors.

- **Fixed-time Speedup**
  - **Gustafson’s Law**
    \[
    \text{Speedup} = \frac{\text{sequentialTime}}{\text{parallelTime}} = \frac{1 - \alpha + \alpha p}{1 - \alpha + \alpha} = 1 - \alpha + \alpha p
    \]
Motivation Example—NAS Benchmark (MPI+OpenMP)
Motivation Example—NAS Benchmark (MPI+OpenMP)

Amdahl’s Law is **UNSuitable** for Multi-Level Parallel Computing
Outline

- Background & Motivation
- Multi-Level Parallel Speedup
- Evaluation
- Conclusion
E-Amdahl’s Law

- Awareness of Different Grained-Level Parallelism

\[
sp(i) = \begin{cases} 
\frac{1}{1 - f(m) + \frac{f(m)}{p(m)}} & (i = m) \\
\frac{1}{1 - f(i) + \frac{f(i)}{p(i)sp(i+1)}} & (1 \leq i < m)
\end{cases}
\]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(m)</td>
<td>The number of nested parallelism levels. ((m \geq 1))</td>
</tr>
<tr>
<td>(p(i))</td>
<td>The number of parallel processing elements in the (i^{th}) level. ((p(i) \geq 1))</td>
</tr>
<tr>
<td>(f(i))</td>
<td>The portion of workload in the (i^{th}) level that can be parallelized. ((0 \leq f(i) \leq 1))</td>
</tr>
<tr>
<td>(sp(i))</td>
<td>The multi-level speedup for the (i^{th}) level.</td>
</tr>
</tbody>
</table>
E-Amdahl’s Law

- Two-Level Parallelism Speedup Model (MPI+OpenMP)

\[
s_p(\alpha, \beta, p, t) = \frac{1}{\alpha(1 - \beta + \frac{\beta}{t})}
\]

where

- \(\alpha\) is the parallel fraction of coarse-grained (MPI-level) parallelism.
- \(\beta\) is the parallel fraction of fine-grained (OpenMP-level) parallelism.
- \(p\) is the number of processes spawned.
- \(t\) is the number of threads spawned per process.
E-Gustafson’s Law

- Awareness of Different Grained-Level Parallelism

\[ sp(i) = \begin{cases} 
1 - f(m) + f(m)p(m) & (i = m) \\
1 - f(i) + f(i)p(i)sp(i+1) & (1 \leq i < m)
\end{cases} \]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(m)</td>
<td>The number of nested parallelism levels. ((m \geq 1))</td>
</tr>
<tr>
<td>(p(i))</td>
<td>The number of parallel processing elements in the (i^{th}) level. ((p(i) \geq 1))</td>
</tr>
<tr>
<td>(f(i))</td>
<td>The portion of workload in the (i^{th}) level that can be parallelized. ((0 \leq f(i) \leq 1)).</td>
</tr>
<tr>
<td>(sp(i))</td>
<td>The multi-level speedup for the (i^{th}) level.</td>
</tr>
</tbody>
</table>
OutLine

• Background & Motivation
• Multi-Level Parallel Speedup
• Evaluation
• Conclusion
Experiment Setup

- **Platform and Configuration**
  - A Linux cluster consisting of eight computing nodes each with two quad-core chips
  - Configuration: One thread per CPU core

- **Benchmarks**
  NAS Parallel Benchmark (NPB) Multi-Zone (MZ) Version:
  - BT-MZ (Unbalanced Workload Partitioning)
  - SP-MZ (balanced Workload Partitioning)
  - LU-MZ (balanced Workload Partitioning)
Performance Prediction

(a) Experimental result of BT-MZ (class W).
(b) Estimated result of BT-MZ.
$\alpha = 0.9771, \beta = 0.5522$
(c) Comparison result of (a) and (b).

(d) Experimental result of SP-MZ (class A).
(e) Estimated result of SP-MZ.
$\alpha = 0.9790, \beta = 0.7263$
(f) Comparison result of (d) and (e).

(g) Experimental result of LU-MZ (class A).
(h) Estimated result of LU-MZ.
$\alpha = 0.9892, \beta = 0.8161$
(i) Comparison result of (g) and (h).
Prediction Result Comparison

Fig. 8: Experimental and estimated speedups of NPB-MZ for different combinations of $p \times t$ under a given total number of 8 processors. The speedup based on *Amdahl’s Law* is estimated with the formula $\frac{1}{1 - \alpha + \frac{\alpha}{p \times t}}$. The speedup based on *E-Amdahl’s Law* is estimated by using Formula (17).
OutLine

- Background & Motivation
- Multi-Level Parallel Speedup
- Evaluation
- Conclusion
Conclusion

• Traditional speedup models are unsuitable for multi-level parallelism
  – Unable to be awareness of different granularities of parallelism for multi-level parallel computing.

• Multi-level Parallelism Model
  – A guidance model for multi-level optimization.
  – A prediction model for multi-level parallelism.
Thank You!

Question?
Algorithm 1 Argument Estimation for $\alpha, \beta$.

1: Execute the program $k$ times in the multi-level parallel execution mode with the chosen parameters $(p_1, t_1), (p_2, t_2), \ldots, (p_k, t_k)$ respectively. Then it gets $(p_1, t_1, s_{p_1}), (p_2, t_2, s_{p_2}), \ldots, (p_k, t_k, s_{p_k})$.

2: Choose all possible combinations of two arrays $(p_i, t_i, s_{p_i})$ and $(p_j, t_j, s_{p_j})$ to figure out the value $(\alpha_s, \beta_s)$ based on Equation (17), where $s$ denotes the $s^{th}$ combination and $1 \leq i, j \leq k$ & $i \neq j$.

3: Check all possible pairs of value $(\alpha_s, \beta_s)$ to guarantee that the pair of estimated values is valid (i.e. $0 \leq \alpha_s \leq 1, 0 \leq \beta_s \leq 1$). Otherwise, discards it.

4: Collect all valid pairs of $(\alpha_i, \beta_i), (i = 1, 2, \ldots, k')$, where $k'$ represents the number of valid pairs. Remove the noise pairs by clustering with the guard condition: $|\alpha_i - \alpha_j| < \varepsilon$ & $|\beta_i - \beta_j| < \varepsilon$.

5: The exact value of $\alpha, \beta$ can thereby be estimated with the formula:

\[
\begin{align*}
\hat{\alpha} &= \frac{1}{\hat{k}} \sum_{i=1}^{\hat{k}} \alpha_i \\
\hat{\beta} &= \frac{1}{\hat{k}} \sum_{i=1}^{\hat{k}} \beta_i,
\end{align*}
\]

where $\hat{k}$ is the number of clustered pairs.
Speedup Under E-Amdahl’s Law
Speedup Under E-Gustafson’s Law