Speedup Versus Efficiency in Parallel Systems

DEREK L. EAGER, JOHN ZAHORJAN, AND EDWARD D. LAZOWSKA

2-22-2011

University of Utah
Abstract

- **Speedup**
  - Can be achieved by executing independent subtasks in parallel

- **Efficiency**
  - Along with an increase in speedup comes a decrease in efficiency
  - due to factors such as contention, communication, and software structure.

- **This paper investigates,**
  - the tradeoff between speedup and efficiency
  - the extent to which this tradeoff is determined by the average parallelism of the software system
  - both speedup and efficiency can simultaneously be poor
  - The incremental benefit and cost of allocating additional processors
Definitions, and sections

- **Definitions**
  - Speedup: \( S(n) = \frac{T_1}{T_n} \)
  - Efficiency: \( E(n) = \frac{S(n)}{n} \)

- **Section II: models of parallel software and its execution**
  - Definitions of average parallelism
  - the number of available processors \( n \) and the average parallelism of the software structure \( A \) provide complementary hardware and software upper bounds on speedup

- **Section III: lower bounds on speedup and efficiency**
  - in terms of \( n \) and \( A \)
  - Try to answer to questions of speedup vs. efficiency

- **Section IV: the incremental cost/benefit of adding processors**

- **Section V: the “knee” of the execution time-efficiency profile**
The System Model and assumptions

The System Model

Assumptions

- Multi-program context is considered
- Overheads such as those due to interconnection network topologies, memory contention, and locking are represented as fixed cost in this model.
The Average Parallelism Measure

- **Definitions**
  - the average number of processors that are busy during the execution time of the software system in question, given an unbounded number of available processors
  - the speedup, given an unbounded number of available processors
  - the ratio of the total service required by the computation to the length of a longest path in the subtask graph
  - the intersection point of the hardware bound and the software bound on speedup

- **Hardware bound and software bound on speedup**

![Graph showing upper bounds and actual speedup for the graph in Fig. 1.](image)
Theorem 1 and its applications

- Let $A$: average parallelism, $n$: number of processors, and $I$: idle time.

$$S(n) = \frac{T_1}{T_n} = \frac{nAT_{\infty}}{AT_{\infty} + I(n)}$$, where $AT_{\infty}$ is total busy time.

And, $I(n) \leq T_{\infty}(n-1)$, considering the longest path is executed by one proc.

By using both $S(n)$ and $I(n)$, we have

$$S(n) \geq \frac{nA}{n + A -1},$$
$$E(n) \geq \frac{A}{n + A -1}$$

- Applications

  - How “Bad” Can Speedup and Efficiency Simultaneously Become?
    - Answer: $E(n) + S(n)/A > 1$

  - To Achieve a Given Speedup, What Efficiency Penalty Must be Paid?
    - Answer: $E(n) \geq (A - S(n))/(A - 1)$
Incremental cost and benefit of adding processors

- **Speedup**
  - When $n=1$, right equation is same to previous one
  - When $k=\infty$, the inequality
    \[
    \max(1, \frac{A}{n}) \leq \frac{S(\infty)}{S(n)} \leq \min\left(1 + (A-1) \frac{k-1}{kn-1}, k\right)
    \]
  - When $n \ll A$, lower bound speedup close to linear in the number of processors
    ex) $n = A/9$, 2 times of $n \rightarrow 80\%$ speedup increase
  - When $n > A \gg 1$, upper bound speedup close to 1
    ex) $n = 4A$, $\rightarrow 25\%$ speedup increase

- **Efficiency**
  - Efficiency is related to speedup in a way of
    \[
    E(n) = \frac{S(n)}{n}
    \]
    \[
    \frac{E(\infty)}{E(n)} \leq \min\left(1 + (A-1) \frac{k-1}{kn-1}, 1\right)
    \]
The Knee of the execution-time efficiency profile

- “knee”: the point where the benefit per unit cost is maximize
  - The point of “maximum power”
  - The system goal is to achieve efficient usage of each processor, while taking into account the cost to users in the form of increased task execution times.

![Execution time-efficiency profile](image)

Fig. 4. Execution time-efficiency profile corresponding to Fig. 1.
The exact location of the knee and bounds on it

- The location of the knee
  - $N$: the number of processors that yields the knee
  - $P_m$: the proportion of time that $m$ processors are simultaneously busy
  - $M_{\text{max}}$: maximum parallelism

\[
  n = \frac{\sum_{m=1}^{m_{\text{max}}} p_m m}{\sum_{m=1}^{n} p_m} \quad \text{or} \quad \frac{\sum_{m=n+1}^{m_{\text{max}}} p_m m}{\sum_{m=1}^{n} p_m} \leq n \leq \frac{\sum_{m=1}^{n-1} p_m m}{\sum_{m=1}^{n} p_m}
\]

- Bounds on the location of the knee
  - $K$: The number of processors that yields the knee

\[
  \frac{A}{2} \leq K \leq 2A - 1.
\]
Conclusion

- Speedup and efficiency cannot simultaneously be low, regardless of scheduling discipline or software structure.

- The result bounds the efficiency cost and speedup benefit possible by altering the number of allocated processors.

- The location of “knee” is well approximated by the average parallelism.

- The result bounds on the speedup and efficiency values that are attained at the knee.