CSE 664 Definitions

By:

Dr. Subra Ganesan
Professor, CSE Department, Oakland University
Rochester, MI 48309. USA.
Degree of Parallelism (DOP)

• During execution of Program, at different times, different number of processors are used.
• DOP profiles depends on: Algorithm, resource utilization, maximum number of processors, memory size
Degree of Parallelism (DOP) contd...

Assume number of processors $n >>$ maximum parallelism in the program, $m$.

$A$, Average Parallelism = \[
\frac{1}{t_2 - t_1} \int_{t_1}^{t_2} DOP(t) dt
\]

\[
A = \left( \sum_{i=1}^{m} i t_i \right) / \sum_{i=1}^{m} t_i
\]

$t_i = \text{Total time}$

$DOP = i$

\[
\sum_{i=1}^{m} t_i = t_2 - t_1
\]
Amdahl’s Law

Let \( \alpha = \) fraction of time for serial execution of program
\( n = \) number of processors

Speed up \( S = \) \( \frac{n}{1 + (n-1)\alpha} \)

As \( n \to \infty \) \( S = \alpha \)

The best Speed up is upper bounded by \( \frac{1}{\alpha} \)
System Efficiency:

\[ E(n) = \frac{S(n)}{n} \]

where \( S \) is the speedup factor and \( n \) is the number of processors.

\[ 1 \leq S(n) \leq n \]; Hence \[ \frac{1}{n} \leq E(n) \leq 1 \]

Redundancy:

\[ R(n) = \frac{O(n)}{O(1)} \]

Where \( O(n) \) = number of unit operations by 1 of the \( n \) processors
\( O(1) \) = number of unit operations in an uniprocessor system
Utilization:
\[ U(n) = R(n) E(n) = \frac{O(n)}{nT(n)} \]

\( U \) = percentage of Processor use

Quality of Parallelism:
\[ Q(n) = \frac{S(n)E(n)}{R(n)} \]

\( Q \) is upper bounded by \( S(n) \)
Standard Performance Measure

MIPS, MFLOPS, KLIPS, Dhrystone, Whetstone LINPACK, TPS
(Transactions per Second)
KLIPS (Kilo Logic Inferences Per Second)

3T performance Goal: 1 Tera flops Computing Power
   1 Tera byte main memory
   1 Terabyte/s I/O bandwidth

To maintain high efficiency for the n-processor system, increase the problem size.
Scalable computer for solving scalable problems.
Cost-effectiveness

Computer Cost $C$ and programming overhead, $p$ speedup and efficiency
Parallel Algorithm & efficient data structures are needed.

Scalability is limited by
1. Communication Latency
2. Bounded Memory Capacity
3. Bounded I/O Bandwidth
4. Limited CPU Speed
Amdahl’s law for fixed workload (1967)

$W_1 = \text{Sequential portion of the work}$

$W_n = \text{Portion of program that can be executed in parallel.}$

$S_n = \frac{W_1 + W_n}{W_1 + W_n/n}$

; if $\alpha = W_1$ ; $W_n = 1 - \alpha$

$= \frac{n}{1 + (n-1)\alpha}$
Gustafson’s Law:
As machine size is increased:
• Increase problem size (more work load)
• Try to get more accuracy by computing with higher MFLOPS power
• Keep the execution time (total) reasonable

• MOORE’S LAW
  The number of transistors on a chip doubles annually.

• ROCK’S LAW
  The cost of semiconductor tools double every four years

• Machrones’s law
  The PC you want to buy is always $5000.
METCALFE’S LAW

A network’s value grows proportionately to the number of its users squared.

• WIRTH’S LAW

Software is slowing faster than hardware is accelerating. (1995)
More Definitions..

• GRAIN SIZE
  Fine, Medium, Coarse.

The basic program segment chosen for parallel processing.
• Latency: Communication Time

Memory Latency:
Synchronization Latency

Balance Granularity and Latency to obtain better performance.
Parallelism Levels

- **Instruction Level** === less than 20 instructions; FINE GRAIN
- **Loop Level** === less than 500 instructions
- **Procedure Level** === Subroutine- 2000 instructions; MEDIUM GRAIN
- **Subprogram level** === Message Passing multiprogramming
- **Programming Level** === COARSE GRAIN
• **Fine Grain** provides a higher degree of parallelism, heavy communication overhead, and scheduling overhead. Assisted by parallelism compiler.

• **Coarse Grain** relies heavily on an effective OS and on the efficiency of the parallel algorithm.

• **Medium Grain** parallelism uses the programmer and the compiler
• **Fine Grain** provides a higher degree of parallelism, heavy communication overhead, and scheduling overhead. Assisted by parallelism compiler.

• **Coarse Grain** relies heavily on an effective OS and on the efficiency of the parallel algorithm.

• **Medium Grain** parallelism uses the programmer and the compiler.
• Shared variable communication is used to support fine and medium grain computations.

• N tasks communicating with each other requires: \( N(N-1)/2 \) communication links.

• Depending on the application, you need to choose to utilize Fine, Medium or Coarse grain parallelism.