Parallel Processing & Distributed Systems

Lectured by: Phạm Trần Vũ

Prepared by: Thoại Nam

Course Detail

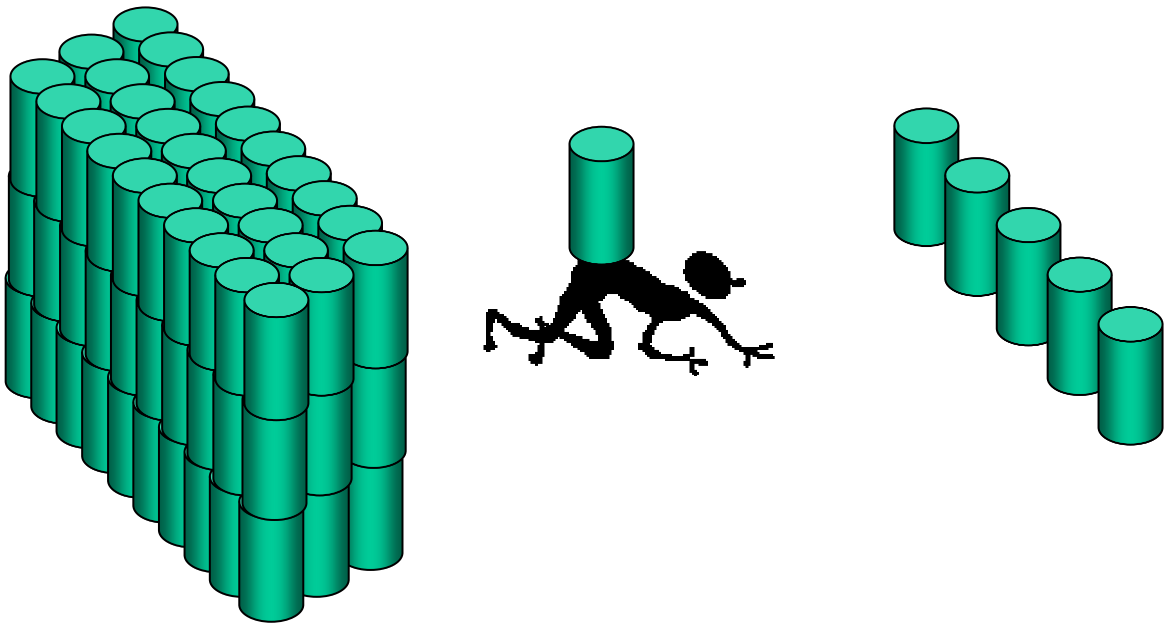
- Two lectures per week (90 minutes each)
  - Tuesday: 10:00 – 11:35
  - Thursday: 8:15 – 9:50
- References
  - *Scalable Parallel Computing: Technology, Architecture, Programming*, Kai Hwang & Zhiwei Xu, McGRAW-HILL, 1997.(*)
  - *Parallel Computing – theory and practice*, Michael J. Quinn, McGRAW-HILL, 1994.(*)
  - *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*, Barry Wilkinson and MiChael Allen, Second Edition, Prentice Hall, 2005.
  - *Distributed Systems: Concepts and Design*, George Coulouris, Jean Dillimore, Tim Kindberg, Addison-Wesley, 2005.(*)
  - *Distributed Algorithms*, Nancy Lynch, Morgan Kaufmann, 1997.
  - *Distributed Operating Systems*, Andrew S. Tanenbaum, Prentice Hall, 1990.
  - *MPI*: http://www.mpi-forum.org/docs/docs.html
  - *PVM*: http://www.csm.ornl.gov/pvm/pvm_home.html
  - *The GRID2: Blueprint for a New Computing Infrastructure*, Ian Foster and Carl Kesselman, Morgan Kaufmann 2004.
  - *Grid Computing: Making the Global Infrastructure a Readlity*, Fran Berman, Geoffgrey Fox and Tony Hey.

Chapter 1: Introduction

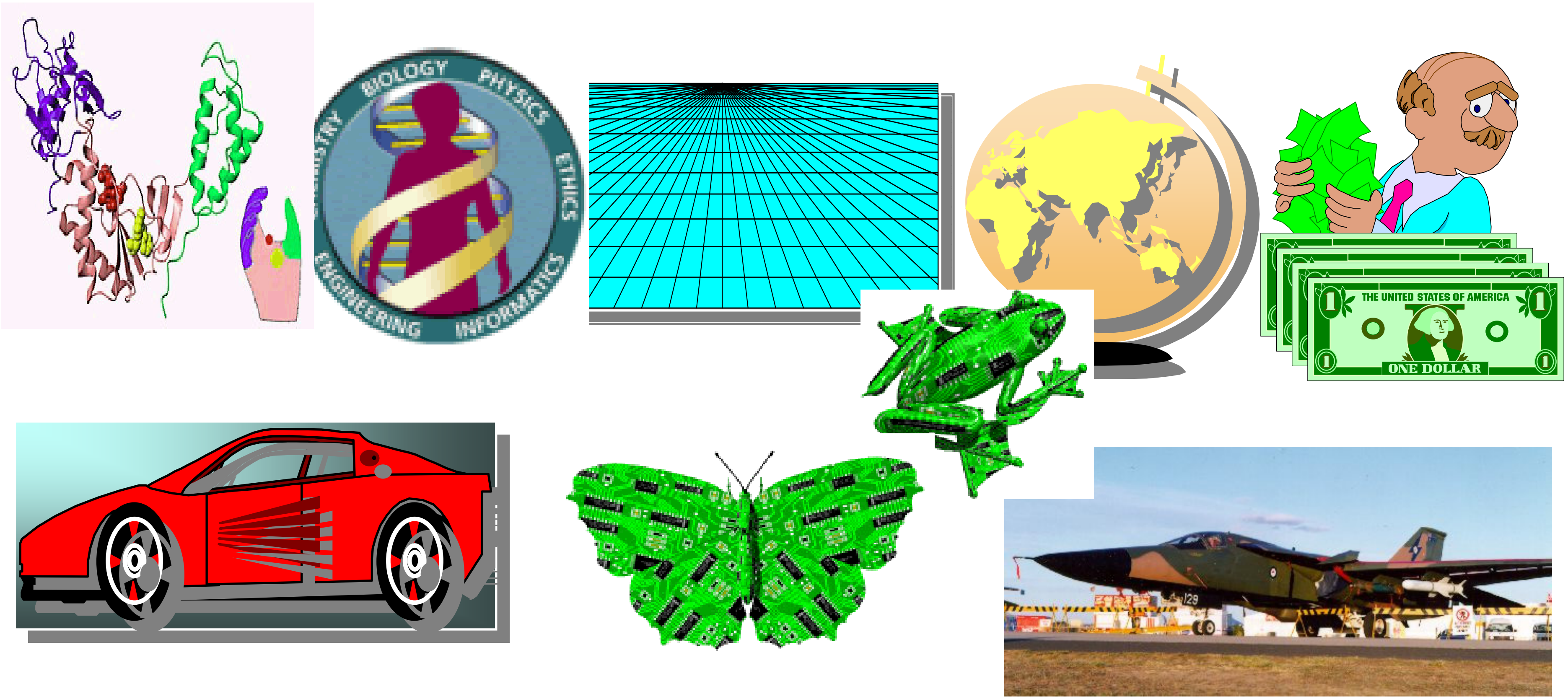
- Introduction
  - What is parallel processing?
  - Why do we use parallel processing?
- Applications
- Parallelism

Sequential Processing

- 1 CPU
- Simple
- Big problems???



Application Demands



Grand Challenge Problems

- A grand challenge problem is one that cannot be solved in a reasonable amount of time with today's computers
- Ex:
  - Modeling large DNA structures
  - Global weather forecasting
  - Modeling motion of astronomical bodies

Solutions

- Power processor
  - 50 Hz -> 100 Hz -> 1 GHz -> 4 Ghz -> ... -> Upper bound?
- Smart worker
  - Better algorithms
- Parallel processing

N-body

- The N^2 algorithm:
  - N bodies
  - N-1 forces to calculate for each bodies
  - N^2 calculations in total
  - After the new positions of the bodies are determined, the calculations must be repeated
- A galaxy:
  - **10^7** stars and so **10^{14}** calculations have to be repeated
  - Each calculation could be done in 1µs (10^{-6}s)
  - It would take **10 years** for one iteration
  - But it only takes **1 day** for one iteration with **3650** processors

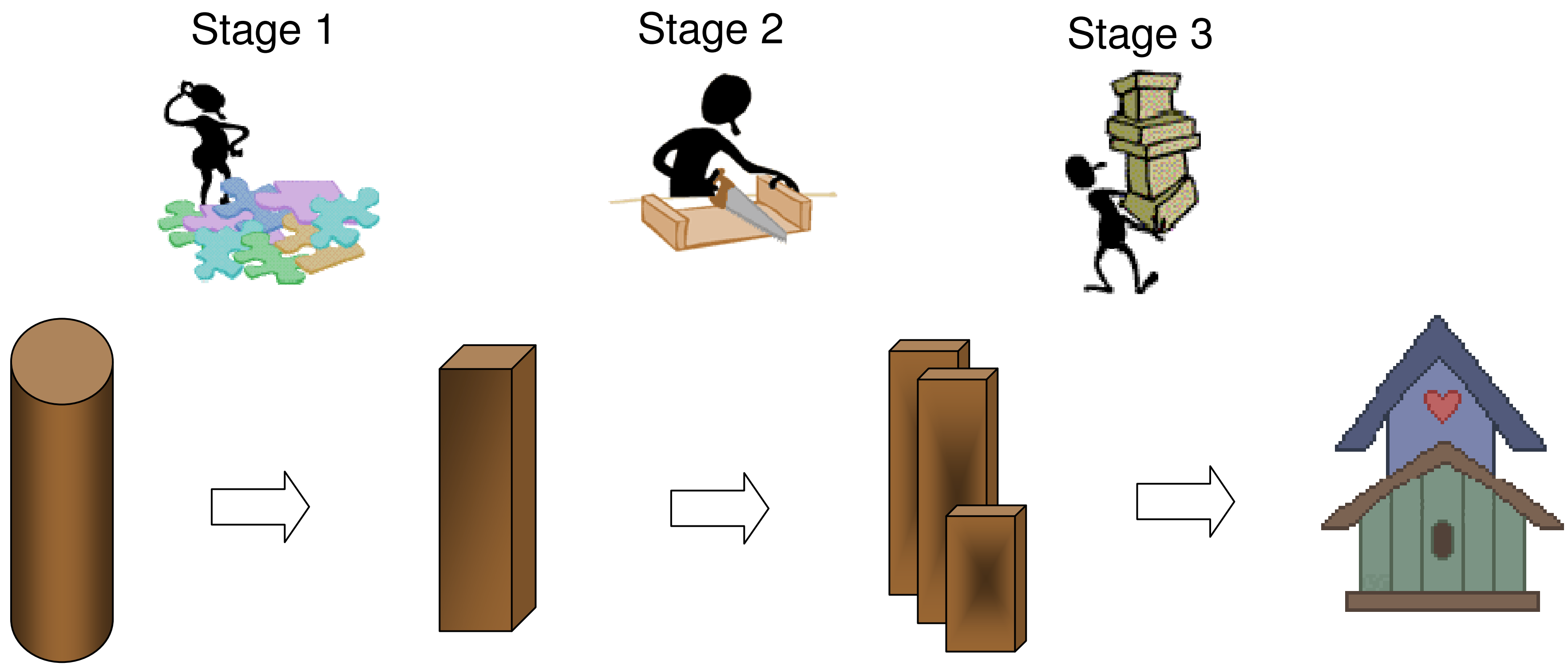
Parallel Processing Terminology

- Parallel processing
- Parallel computer
  - Multi-processor computer capable of parallel processing
- Throughput:
  - The throughput of a device is the number of results it produces per unit time.
- Speedup

  S = Time(the most efficient sequential algorithm) / Time(parallel algorithm)
- Parallelism:
  - Pipeline
  - Data parallelism
  - Control parallelism

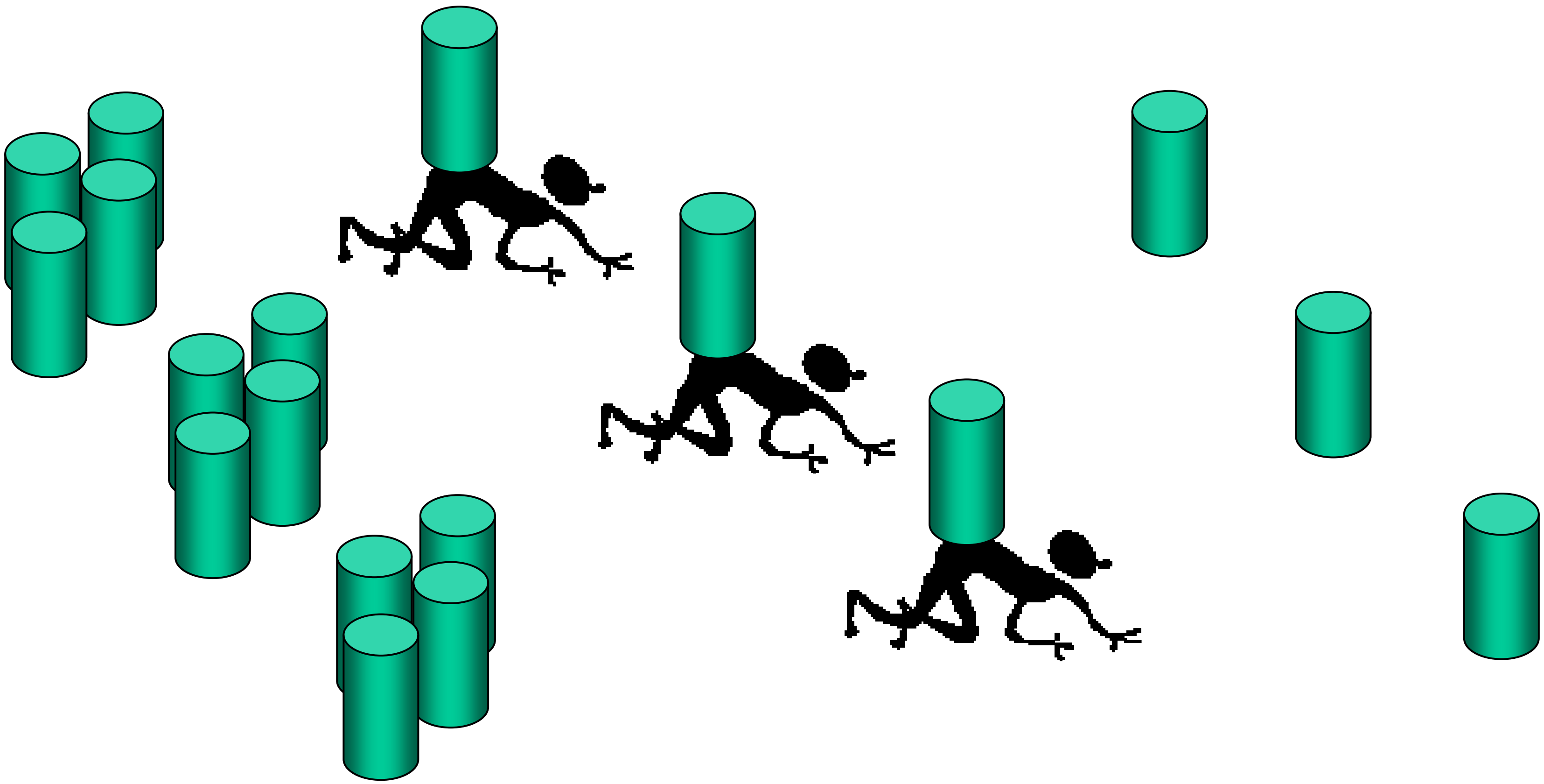
Pipeline

- A number of steps called **segments** or **stages**
- The output of one segment is the input of other segment



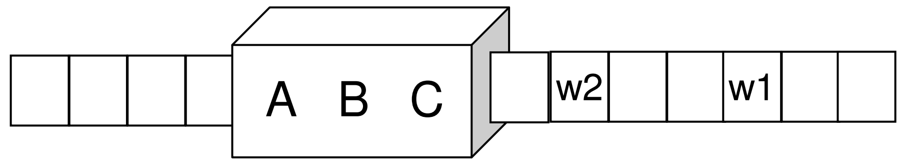
Data Parallelism

- Applying the same operation simultaneously to elements of a data set

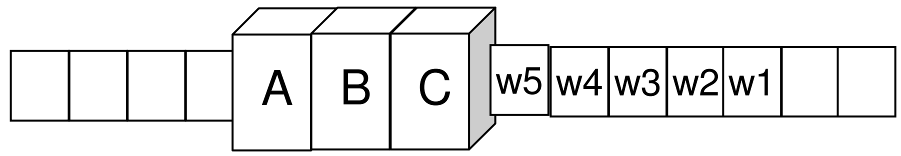


Pipeline & Data Parallelism

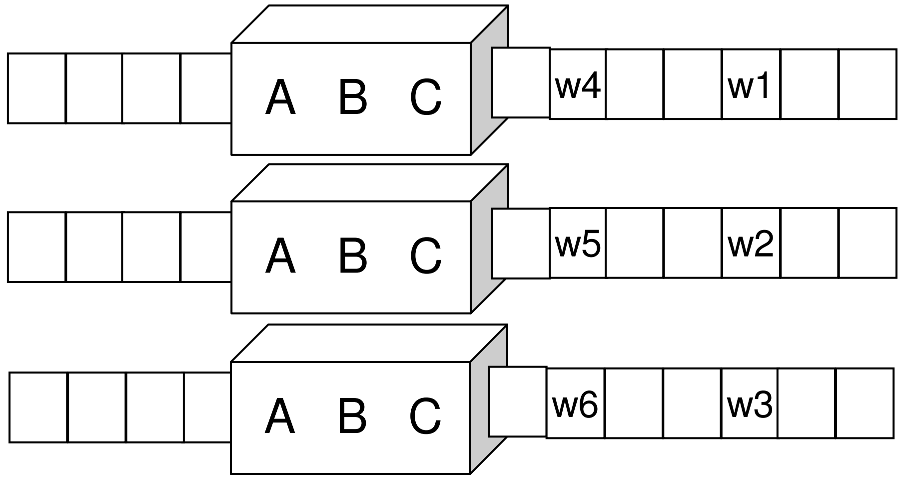
1. Sequential execution



2. Pipeline



3. Data Parallelism

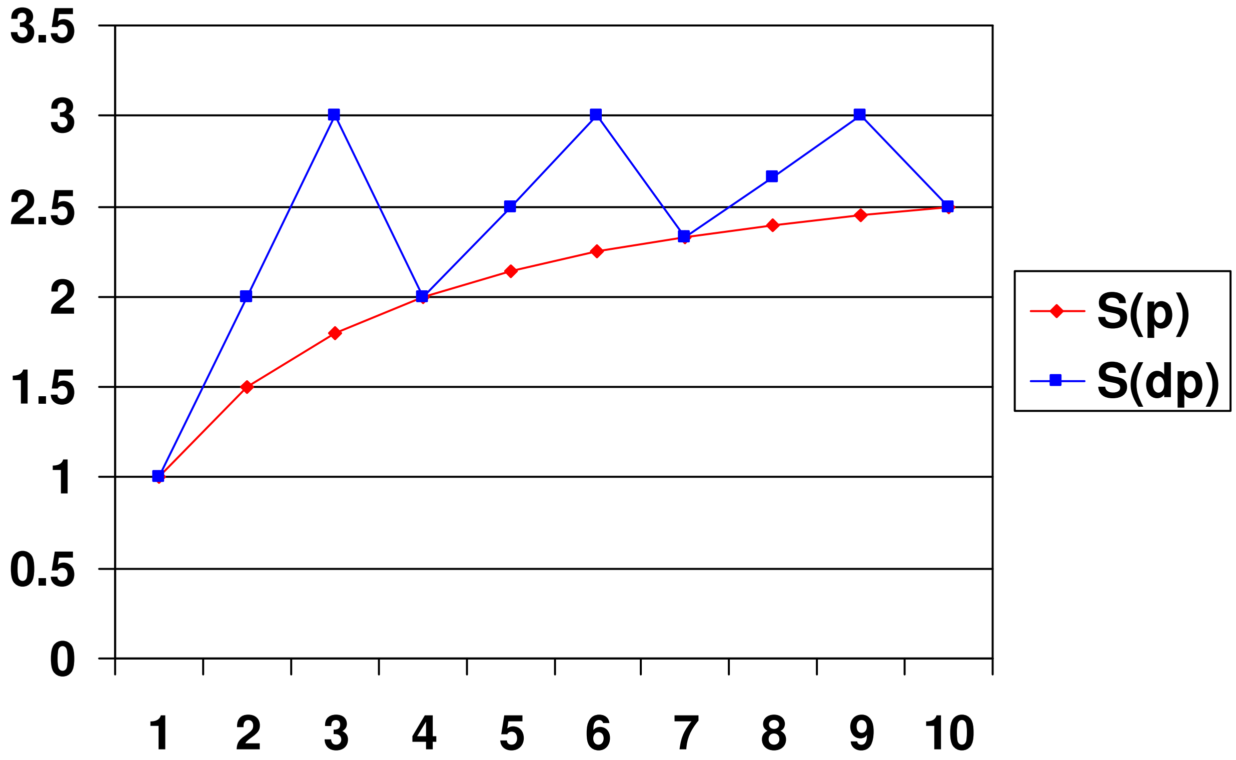


Pipeline & Data Parallelism

- Pipeline is a special case of control parallelism
- T(s): Sequential execution time
  T(p): Pipeline execution time (with 3 stages)
  T(dp): Data-parallelism execution time (with 3 processors)
  S(p): Speedup of pipeline
  S(dp): Speedup of data parallelism

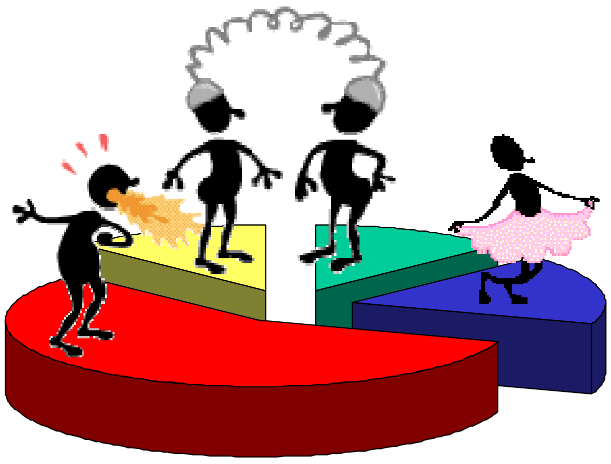
widget	1	2	3	4	5	6	7	8	9	10
T(s)	3	6	9	12	15	18	21	24	27	30
T(p)	3	4	5	6	7	8	9	10	11	12
T(dp)	3	3	3	6	6	6	9	9	9	12
S(p)	1	1+1/2	1+4/5	2	2+1/7	2+1/4	2+1/3	2+2/5	2+5/11	2+1/2
S(dp)	1	2	3	2	2+1/2	3	2+1/3	2+2/3	3	2+1/2

Pipeline & Data Parallelism



Control Parallelism

- Applying different operations to different data elements simultaneously



Scalability

- An algorithm is scalable if the level of parallelism increases at least linearly with the problem size.
- An architecture is scalable if it continues to yield the same performance per processor, albeit used in large problem size, as the number of processors increases.

- Data-parallelism algorithms are more scalable than control-parallelism algorithms