Parallel Processing & Distributed Systems

Thoai Nam

Faculty of Computer Science and Engineering

HCMC University of Technology

Chapter 1: Introduction

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

Supercomputers: TOP500

Titan – 17,59 petaflops (560.640 cores)

K computer – 8 petaflops (548.352 cores)

Tianhe-2 (MilkyWay-2) – 33,8 petaflops (3.120.000 cores)

Sequoia – 17,17 petaflops (1.572.864 cores)

SuperMUC – 2,897 petaflops (147.456 cores)

Supercomputers: TOP500

Thiên hà 1A – 2,57 petaflops

Jaguar XT5 – 1,76 petaflops

Nebulae – 1,27 petaflops

Tsubame 2.0 – 1,2 petaflops

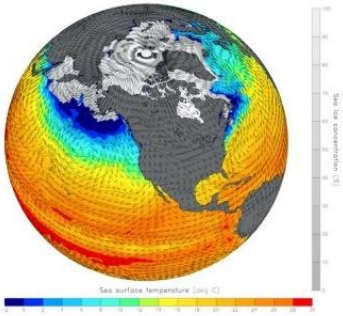
Hopper – 1,054 petaflops

Tera-100 – 1,05 petaflops

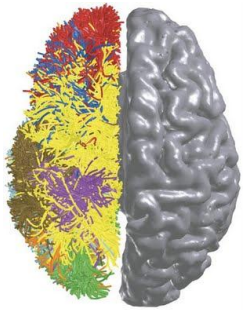
Supercomputing applications

Khí động học trong tàu vũ trụ

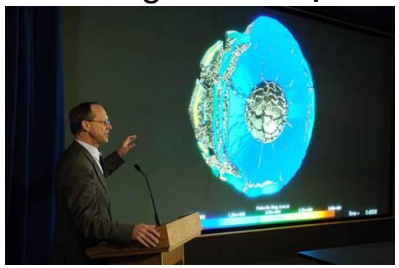
Tràn dầu của BP



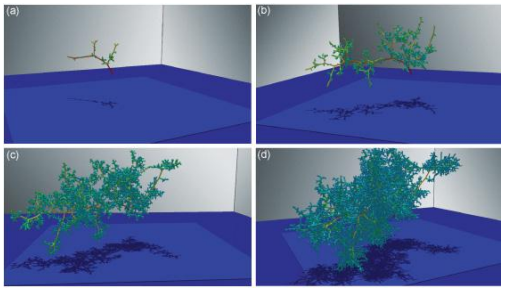
Mô hình thời tiết PCM



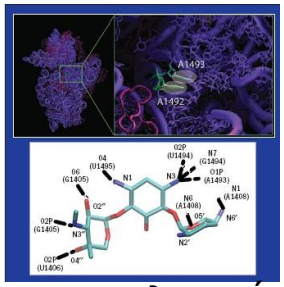
Mô phỏng não



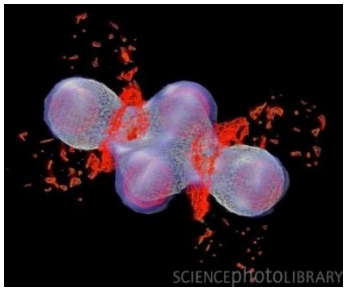
Mô phỏng tiểu hành tinh



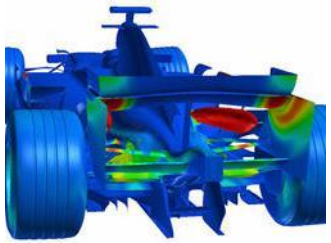
Mô phỏng Uranium-235 hình thành từ phân rã Phutonium-239



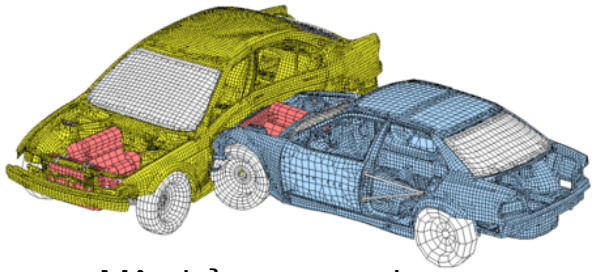
Tác dụng của thuốc ở mức phân tử



Mô phỏng nguyên tử Lithium



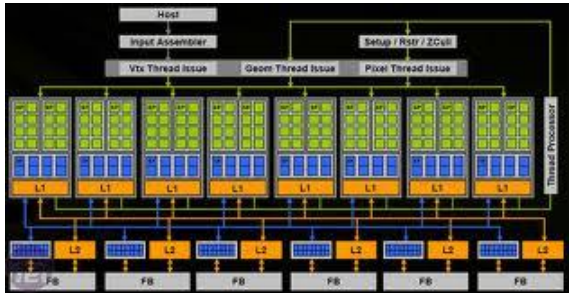
Mô phỏng Renault F1

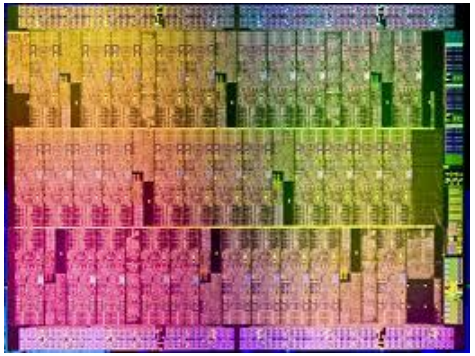


Mô phỏng xe va chạm

Parallel architecture

- Multi-core
- Many core

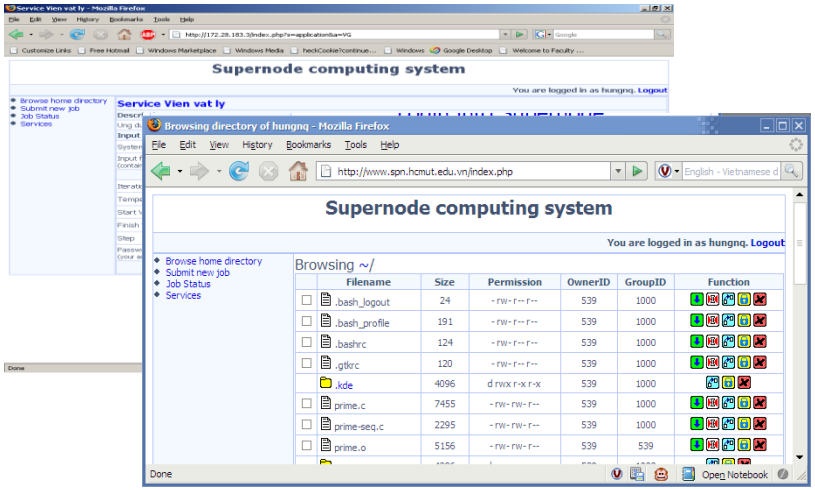




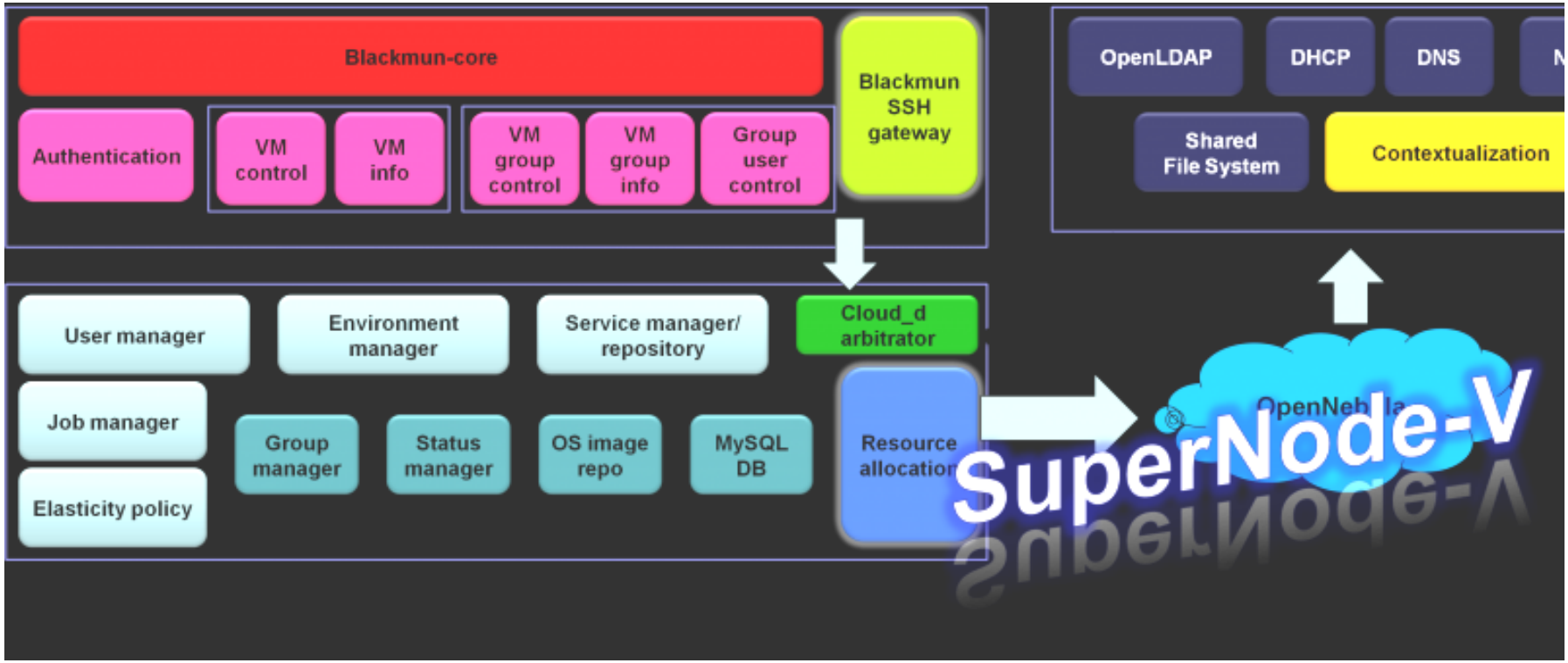
SuperNode I & II

SuperNode I in 1998-2000

SuperNode II in 2003-2005

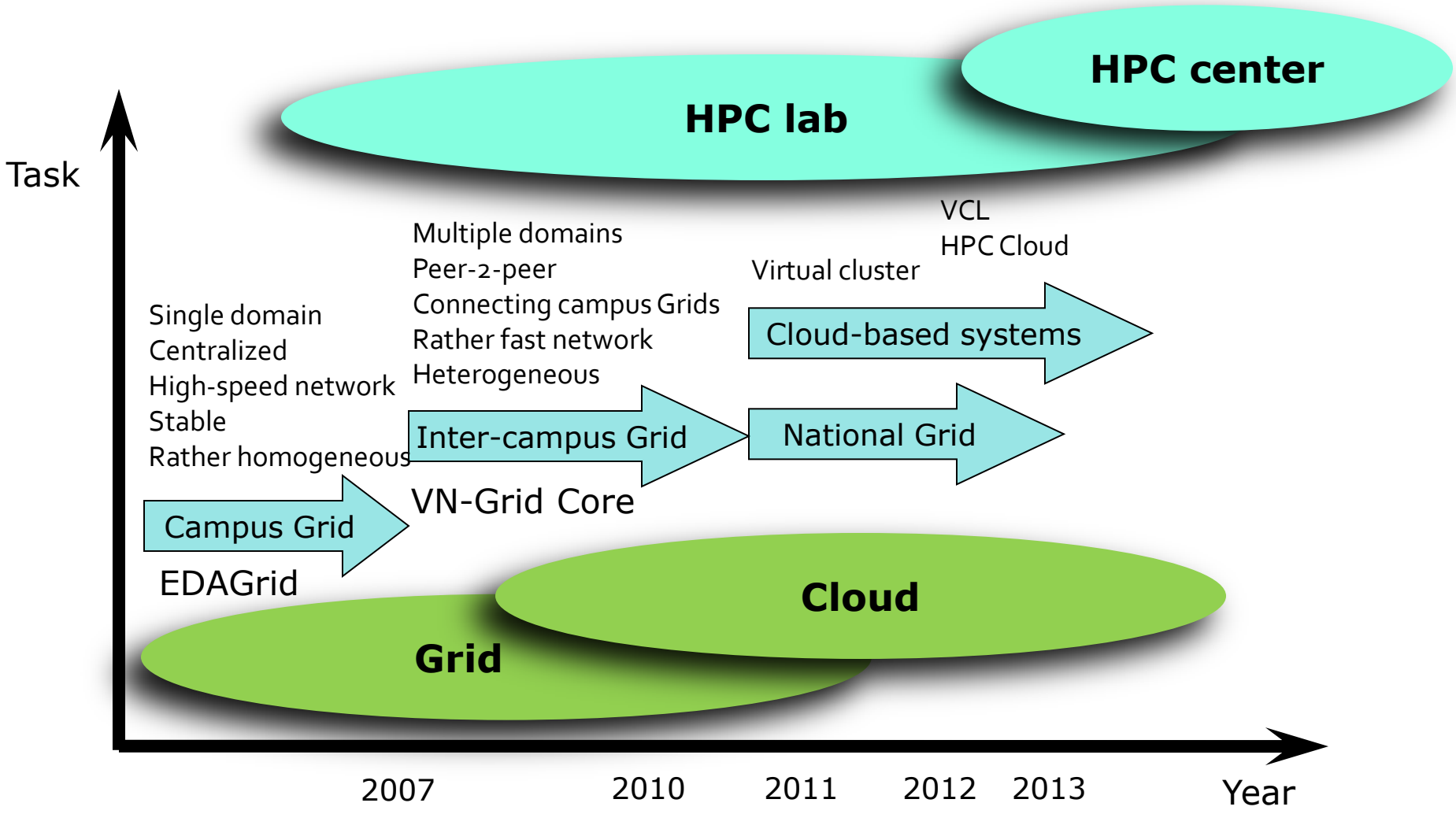


SuperNode V



SuperNode-V project: 2010-2012

SuperNode V


HPC center
HPC lab
Task
VCL
HPC Cloud
Multiple domains
Peer-2-peer
Connecting campus Grids
Rather fast network
Heterogeneous
Virtual cluster
Single domain
Centralized
High-speed network
Stable
Rather homogeneous
Cloud-based systems
Inter-campus Grid
National Grid
VN-Grid Core
Campus Grid
EDAGrid
Cloud
Grid
2007
2010
2011
2012
2013
Year

EDA-Grid & VN-Grid

SuperNode II

Applications
Chip design
Data mining
Airfoid optimization

Security

Monitoring

User Management

Scheduling

Campus/VN-Grid (GT)

Resource Management

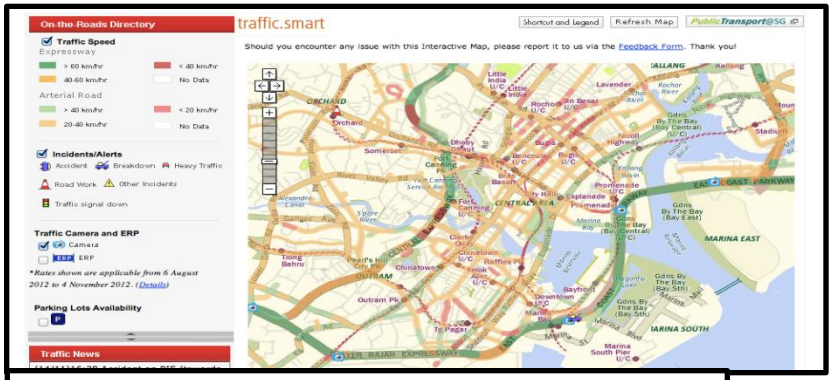
Information Service

Data Service

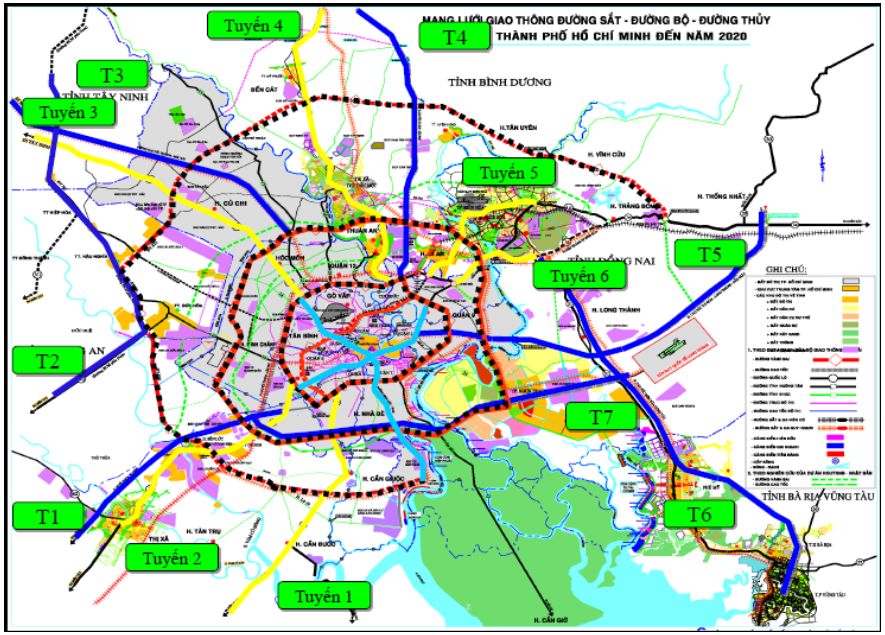
POP-C++

HPC group at HCMUT

- 5 Dr. + 6 Postdoc
- Research projects: Clusters, Grid and Cloud Computing
- Region activities: PRAGMA
- HPC Center
- Solving big problems



Singapore (http://interactivemap.onemotoring.com.sg/mapapp/index.html)

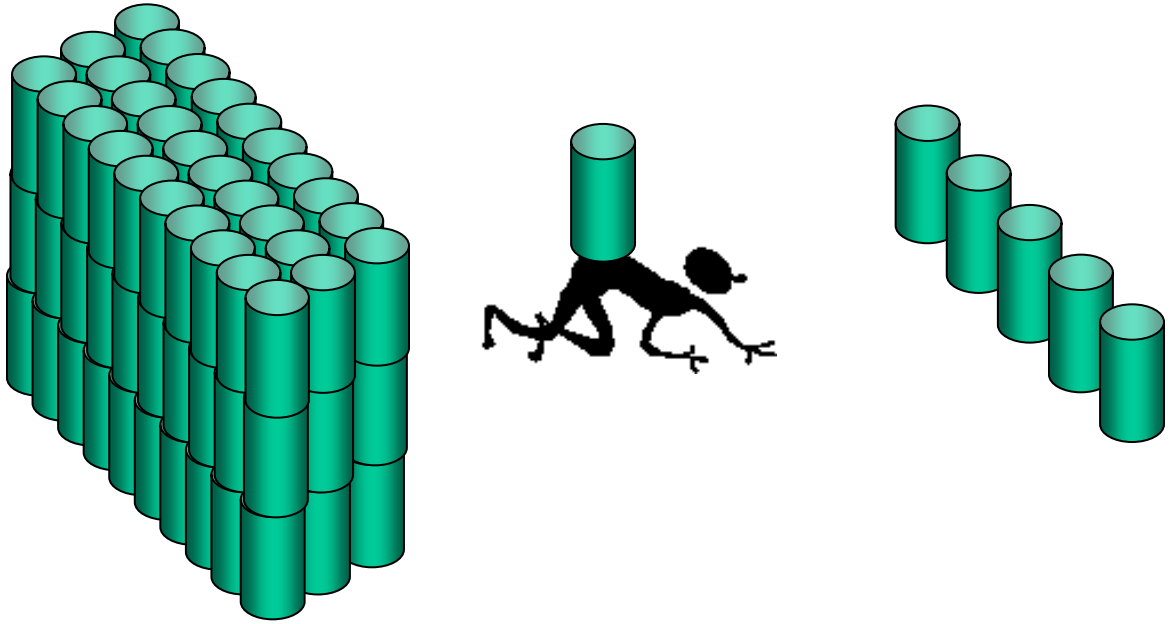


How to do

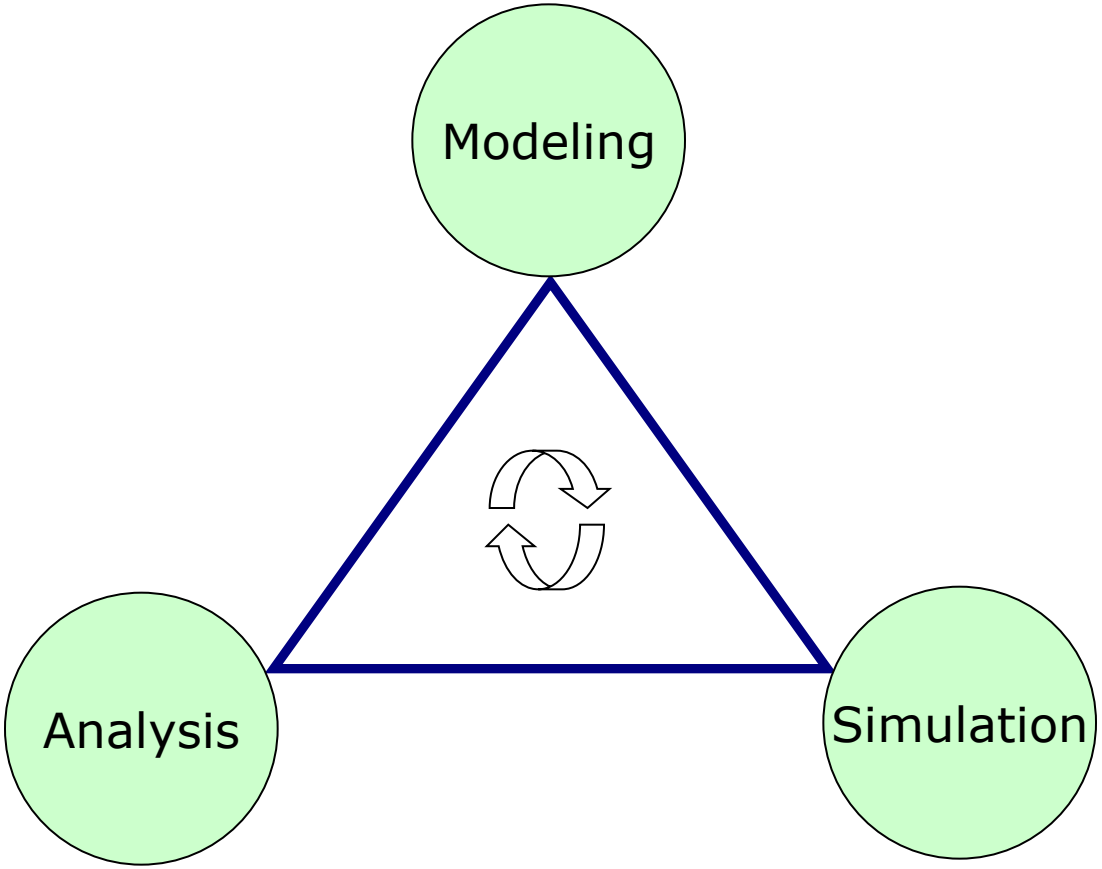
Parallel processing & Distributed systems

Sequential Processing

- 1 CPU
- Simple
- Big problems???



New Approach



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

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

Solutions

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

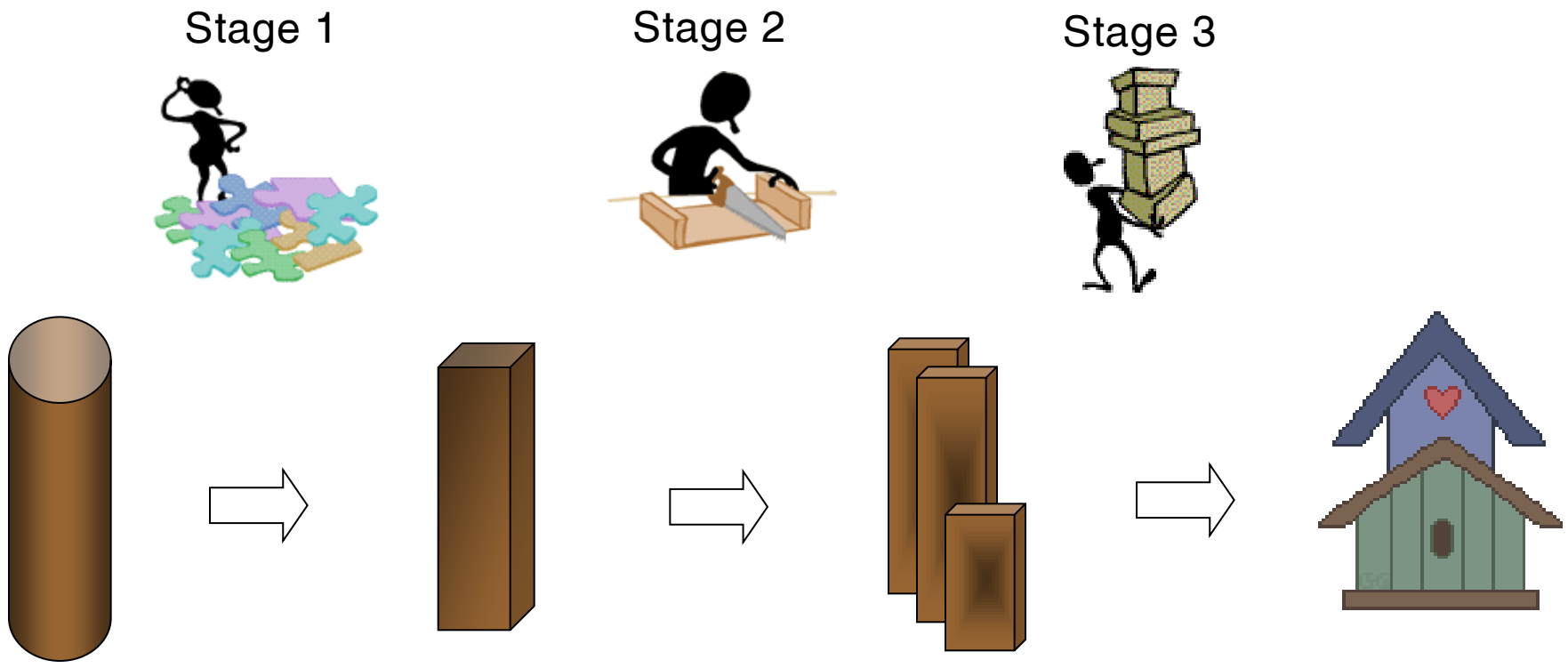
Parallel Processing Terminology

- 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

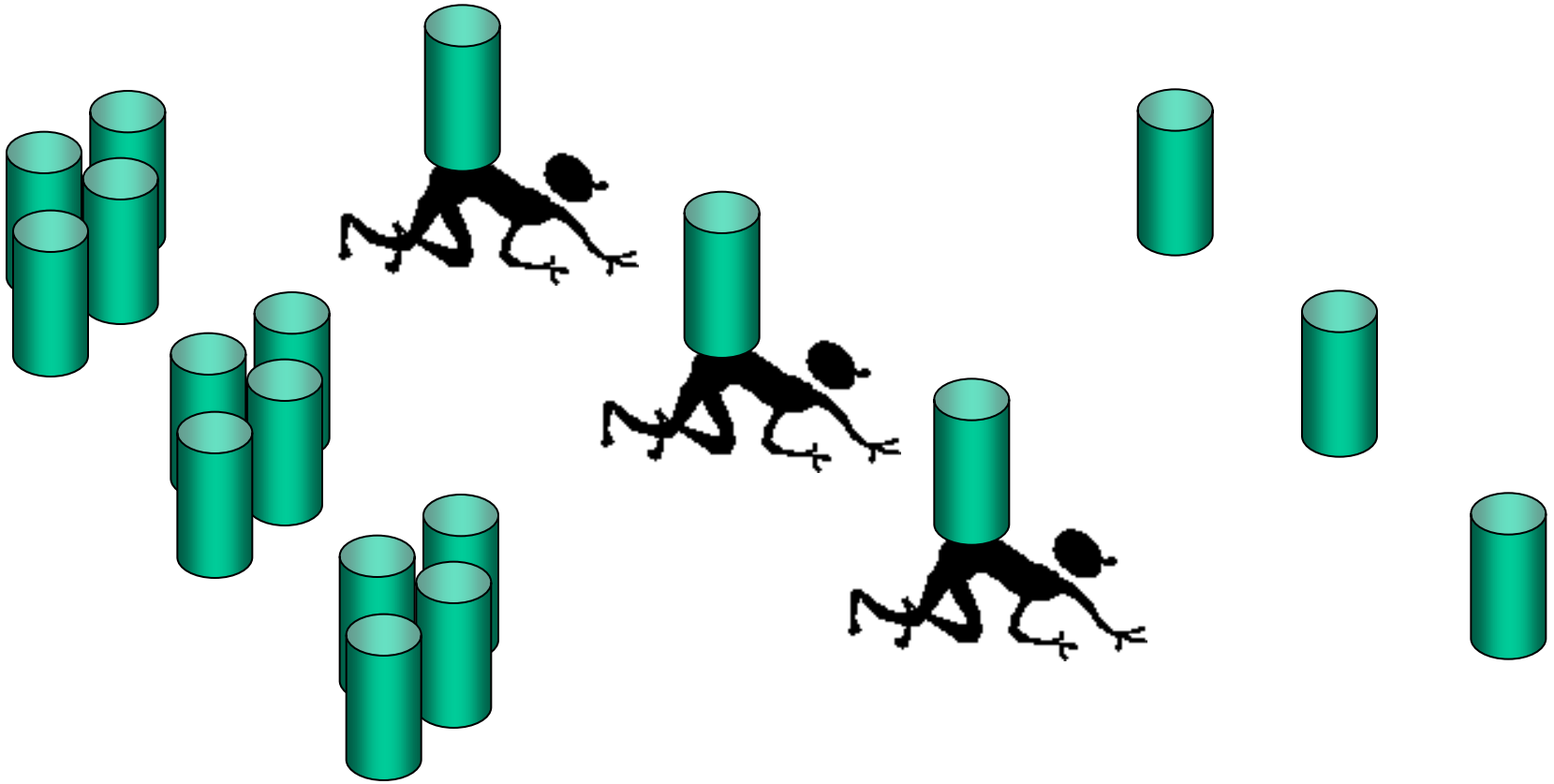
Pipeline

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



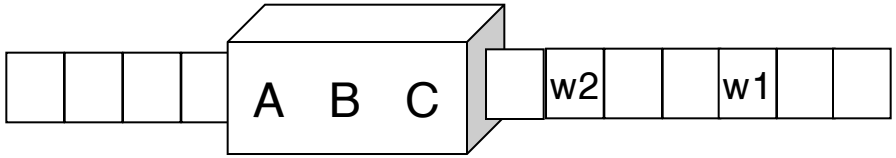
Data Parallelism

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

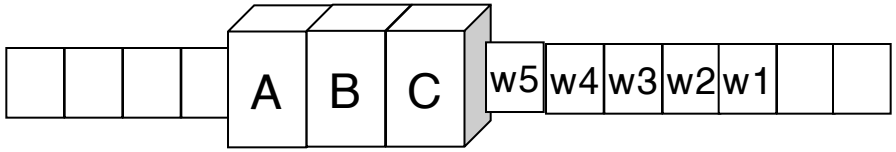


Pipeline & Data Parallelism

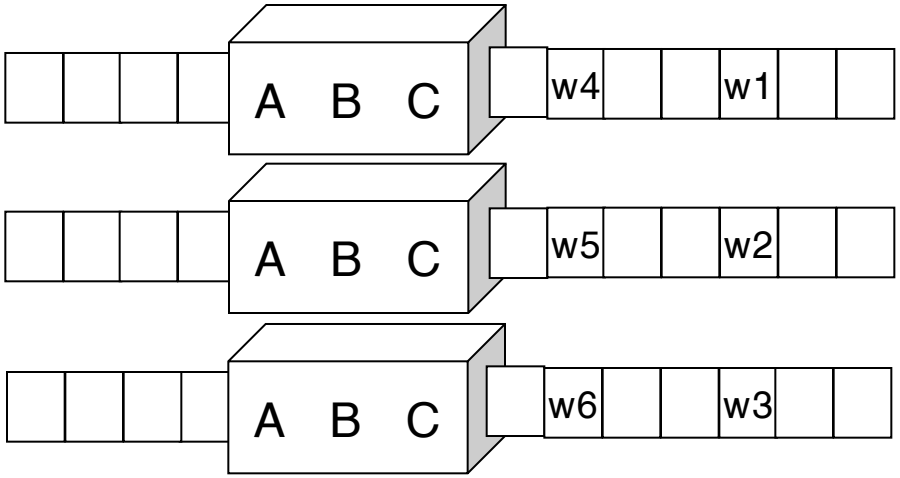
1. Sequential execution



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)

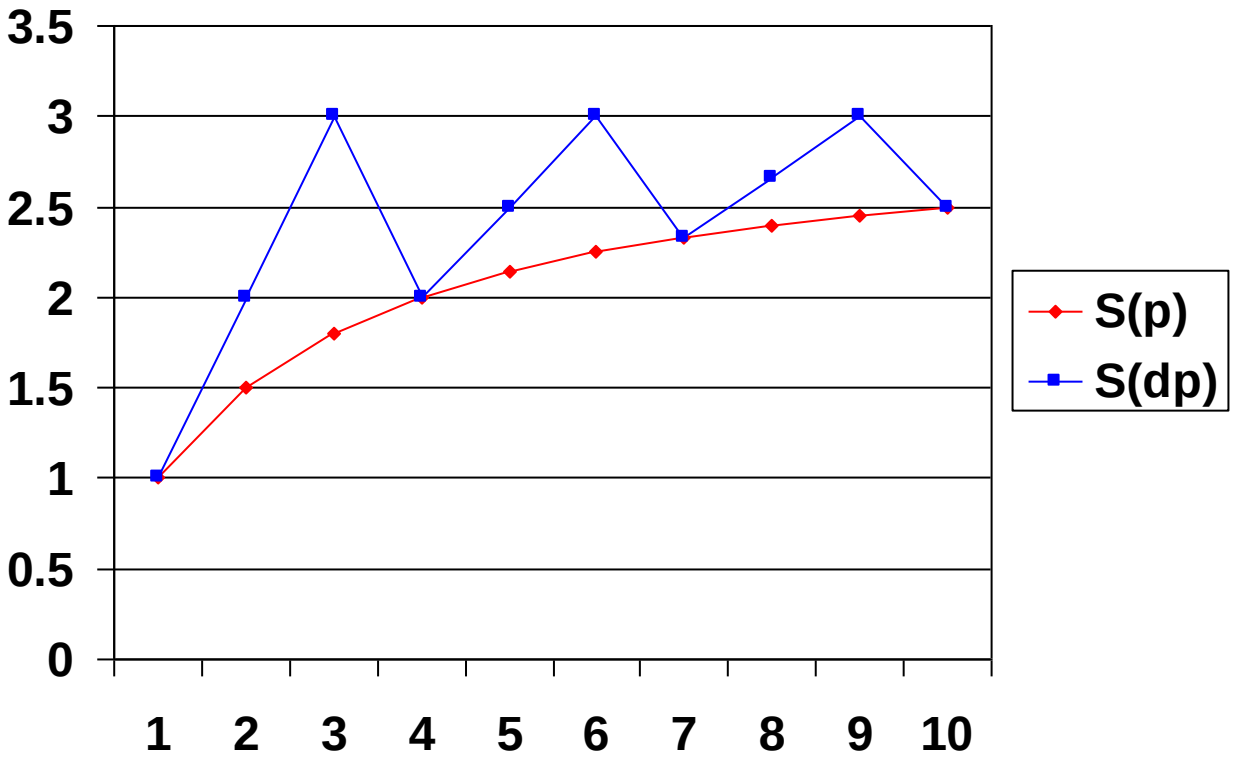
  T(dp): Data-parallelism execution time (with 3 processors)

  S(p): Speedup of pipeline

  S(dp): Speedup of data parallelism

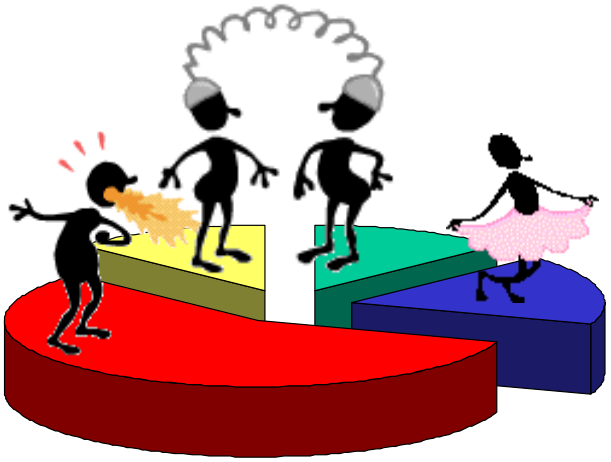
widget	1	2	3	4	5	6	7	8	9	10
T(s)	3	6	9	12	15	18	21	24	27	30
T(p)	3	4	5	6	7	8	9	10	11	12
T(dp)	3	3	3	6	6	6	9	9	9	12
S(p)	1	1+1/2	1+4/5	2	2+1/7	2+1/4	2+1/3	2+2/5	2+5/11	2+1/2
S(dp)	1	2	3	2	2+1/2	3	2+1/3	2+2/3	3	2+1/2

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