# Parallel Computing

Daniel Merkle

# Course Introduction

- Communication media:
  - http://www.imada.shu.dk/~daniel/parallel
  - Personal Mail: daniel@imada.sdu.dk

- Schedule:
  - Tuesday 8.00 ct, Thursday 12.00 ct (if necessary)
  - 2 quarters

- Evaluation:
  - Project assignments (min. 3 per quarter)
    Theoretical + programming exercises

    Oral Exam

  - ...course may change to a reading course

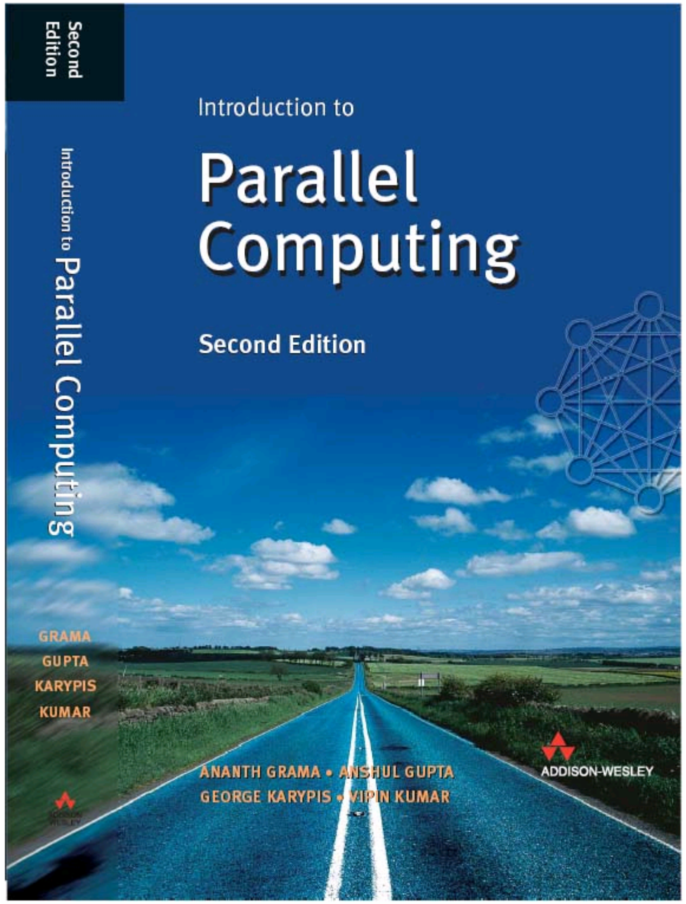
# Course Introduction

- Literature:
  - main course book:

    Grama, Gupta, Karypis, and Kumar : Introduction to Parallel Computing (Second Edition, 2003)

    other sources will be announced

  - Weekly notes



# Parallel Computing – Course Overview

- PART I: BASIC CONCEPTS
- PART II: PARALLEL PROGRAMMING
- PART III: PARALLEL ALGORITHMS AND APPLICATIONS

# Outline

PART I: BASIC CONCEPTS

- Introduction
- Parallel Programming Platforms
- Principles of Parallel Algorithm Design
- Basic Communication Operations
- Analytical Modeling of Parallel Programs

PART II: PARALLEL PROGRAMMING

- Programming Shared Address Space Platforms
- Programming Message Passing Platforms

# Outline

PART III: PARALLEL ALGORITHMS AND APPLICATIONS

- Dense Matrix Algorithms
- Sorting
- Graph Algorithms
- Discrete Optimization Problems
- Dynamic Programming
- Fast Fourier Transform
- maybe also: Algorithms from Bioinformatics

# Example: Discrete Optimization Problems

- The 8-puzzle problem

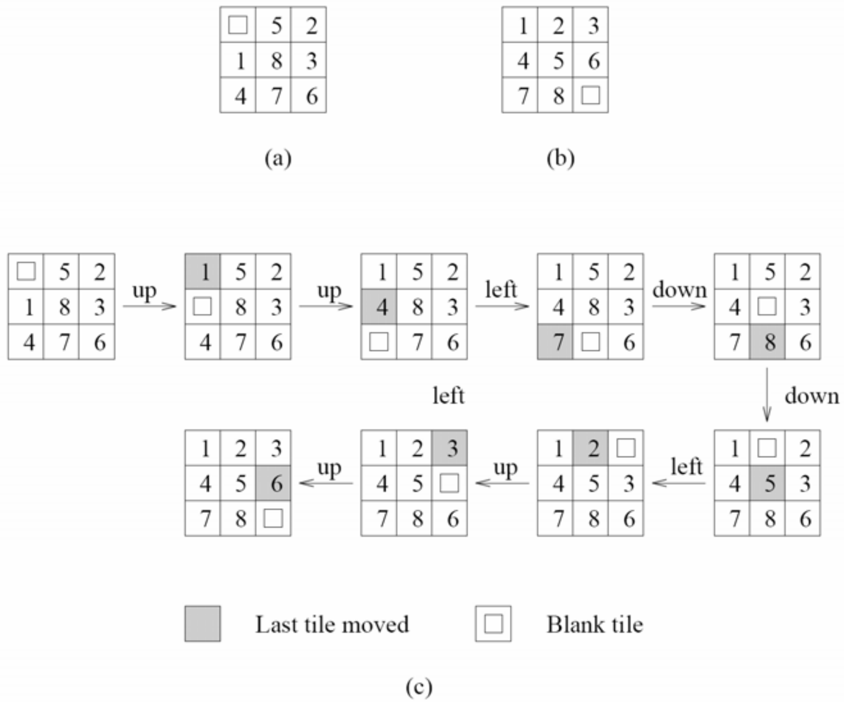



**Figure 11.1** An 8-puzzle problem instance: (a) initial configuration; (b) final configuration; and (c) a sequence of moves leading from the initial to the final configuration.

# Discrete Optimization – sequential

- Depth-First-Search, 3 steps:

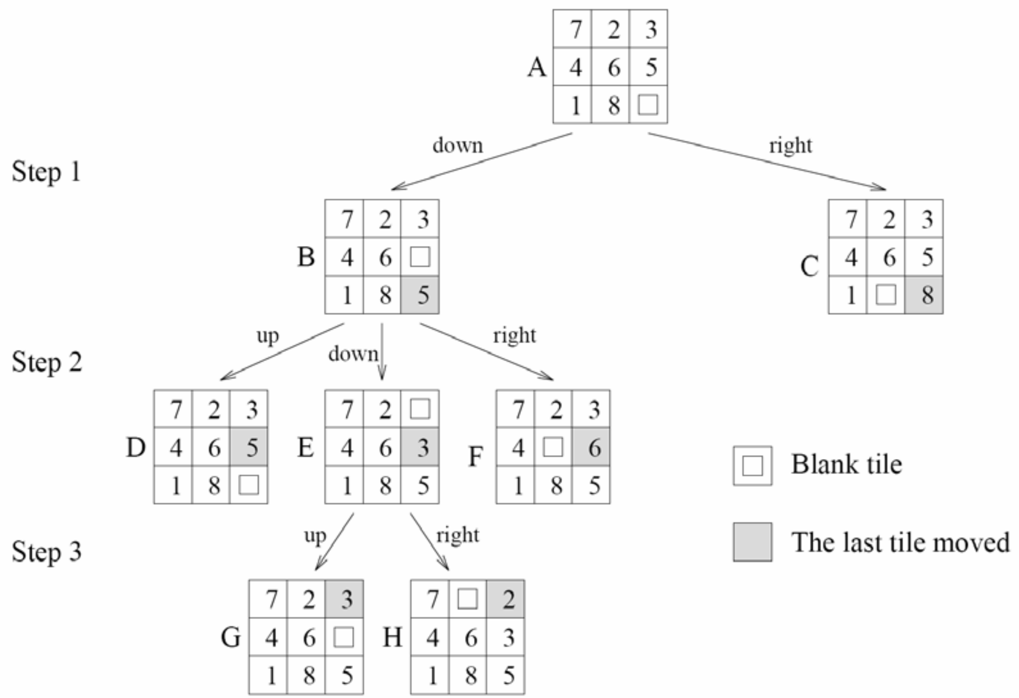
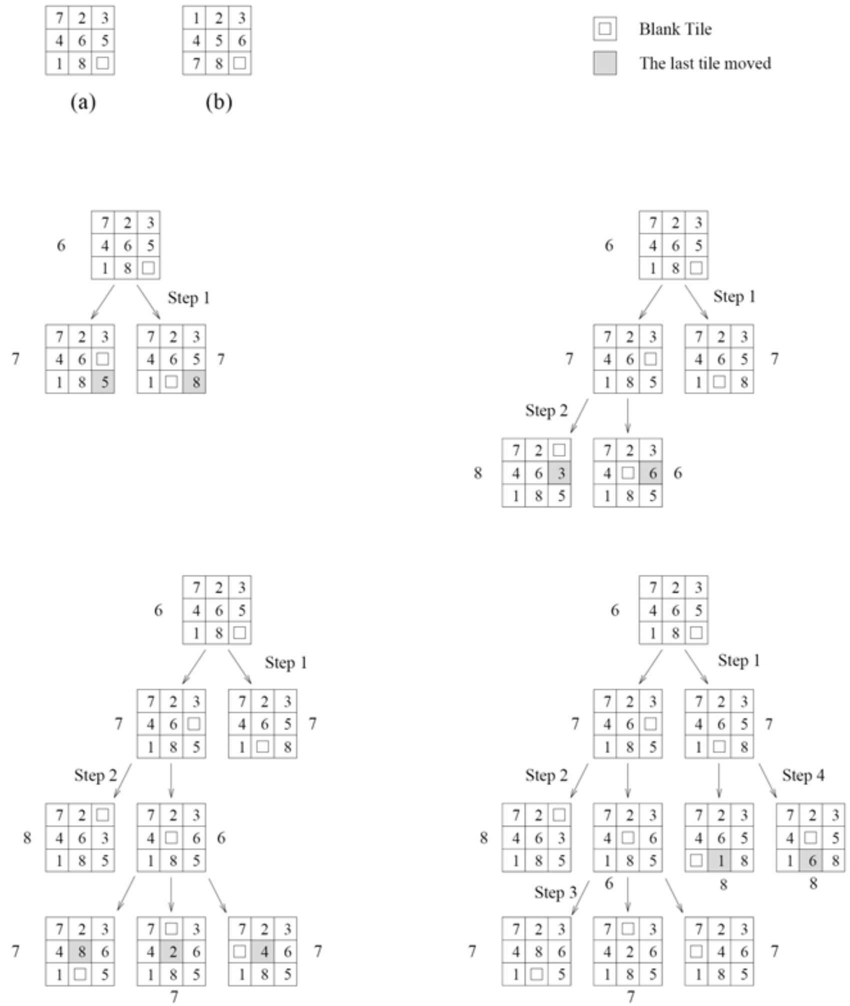



**Figure 11.4** States resulting from the first three steps of depth-first search applied to an instance of the 8-puzzle.

# Discrete Optimization – sequential

- Best-First-Search:



# Discrete Optimization - parallel

- Depth First Search - parallel:

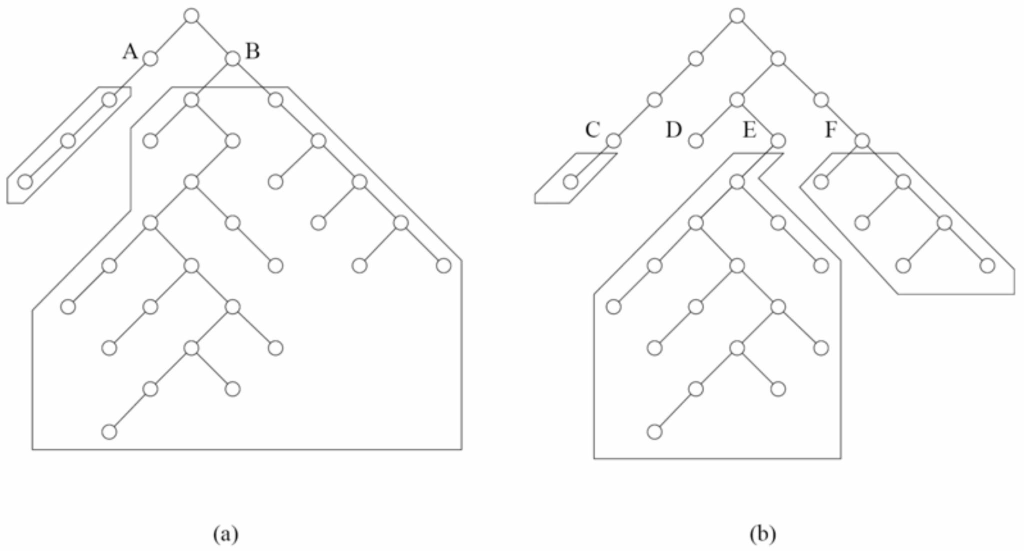
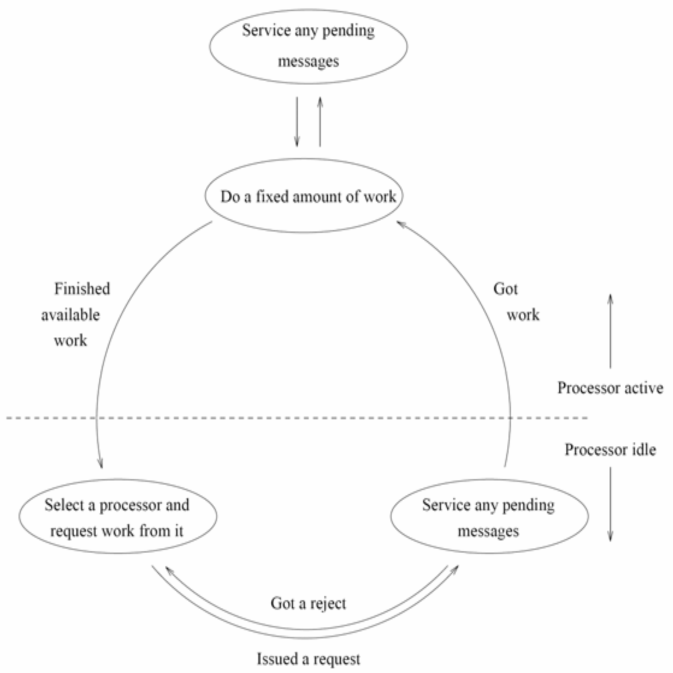



**Figure 11.7** The unstructured nature of tree search and the imbalance resulting from static partitioning.

⇨load balancing

# Discrete Optimization - parallel Dynamic Load Balancing

- Generic Scheme:

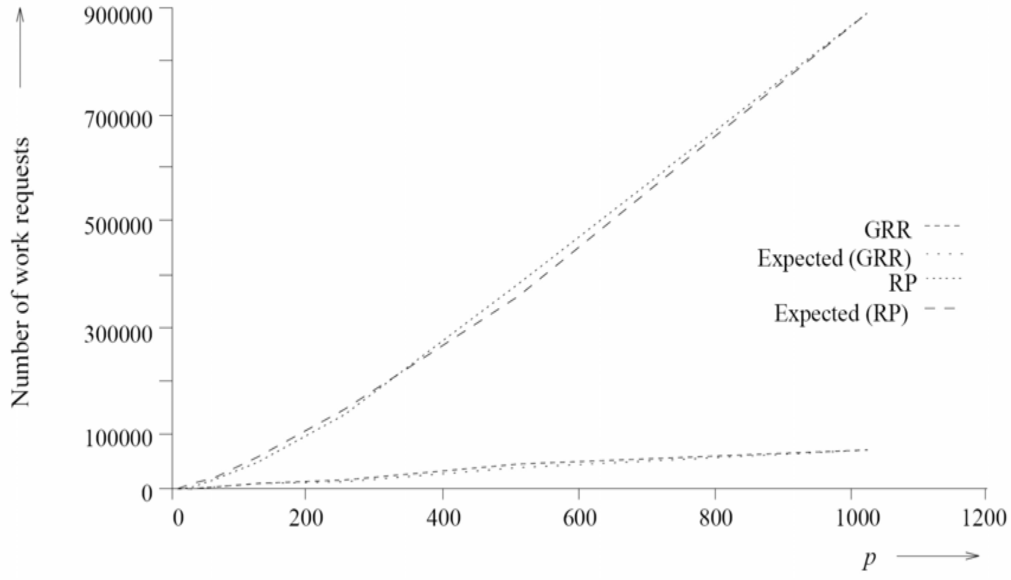



- Load Balancing Schemes: e.g. Round-Robin, Random Polling
  - Scalability analysis
  - Experimental results
  - Speedup anomalies

# Discrete Optimization
# Analytical vs. Experimental Results

- Number of work requests
  (analytically derived expected values and experimental results):

| Scheme | Number of processors 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 |
|---|---|---|---|---|---|---|---|---|
| GRR | 260 | 661 | 1572 | 3445 | 8557 | 17088 | 41382 | 72874 |
| RP | 562 | 2013 | 5106 | 15060 | 46056 | 136457 | 382695 | 885872 |



# Introduction

Information Technology

## Gartner's Seven IT Grand Challenges

**What are the most important IT challenges for the next 25 years? At the recent Gartner Emerging Trends Symposium/ITxpo, Gartner analysts identified seven IT grand challenges that, if met, will have profound economic, scientific and societal impacts. They are:**

- Eliminate the need to manually recharge wireless devices
- Parallel programming applications that fully exploit multicore processors
- Non-tactile, natural computing interfaces
- Automated computer-to-human speech translation
- Reliable, long-term digital storage
- Increase programmer productivity by 100 percent
- Identify the financial consequences of IT investments

"IT leaders should always be looking ahead for the emerging technologies that will have a dramatic impact on their business, and information on many of these future innovations are already in some public domain," says Gartner VP Ken McGee. To find such information, Gartner suggests examining relevant research papers, patents, and production prototypes.

10 COMMUNICATIONS OF THE ACM JULY 2008 | VOL. 51 | NO. 7

# Introduction

- Motivating Parallelism
  - Multiprocessor / Multicore architectures get more and more usual
  - Data intensive applications: web server / databases / data mining
  - Computing intensive applications: for example realistic rendering (computer graphics), simulations in life sciences: protein folding, molecular docking, quantum chemical methods, ...
  - Systems with high availability requirements: Parallel Computing for redundancy

# Cisco unveils 40-core networking processor

## Cisco packs 800 million transistors on its new Quantum Flow Processor

By Sharon Gaudin    Comments 1    Recommended 122    Share

February 26, 2008 (Computerworld) Cisco Systems Inc. unveiled its QuantumFlow Processor, a networking semiconductor that has 40 cores on a single chip.

The processor, the result of a five-year Cisco development effort, has a fully integrated and programmable networking chip set that controls different functions of a system, such as data transfer. According to Cisco, the processor can perform up to 160 simultaneous processes.

# Tilera announces 64-core processor

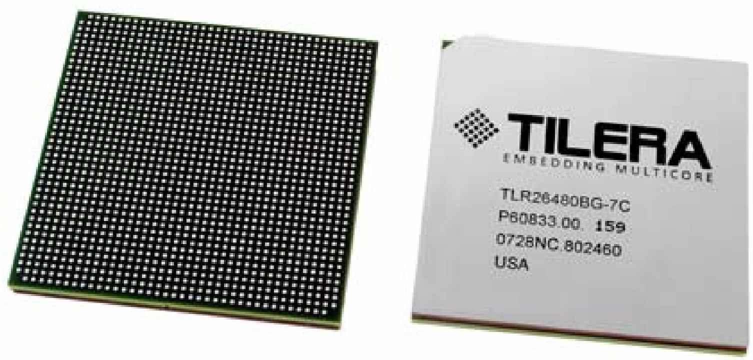
Hardware

By Wolfgang Gruener

Monday, August 20, 2007 00:00

3 diggs  vote now  digg it  buzz up

**Palo Alto (CA) - Silicon Valley startup Tilera today announced the Tile64, a processor with 64 programmable cores that, according to the company, houses ten times the performance and 30 times the power efficiency of Intel's dual-core Xeon processors.**



# General-purpose computing on graphics processing units

From http://www.acmqueue.org 04/08

## Scalable Parallel Programming with CUDA

*by John Nickolls, Ian Buck, and Michael Garland, Nvidia, Kevin Skadron, University of Virginia*

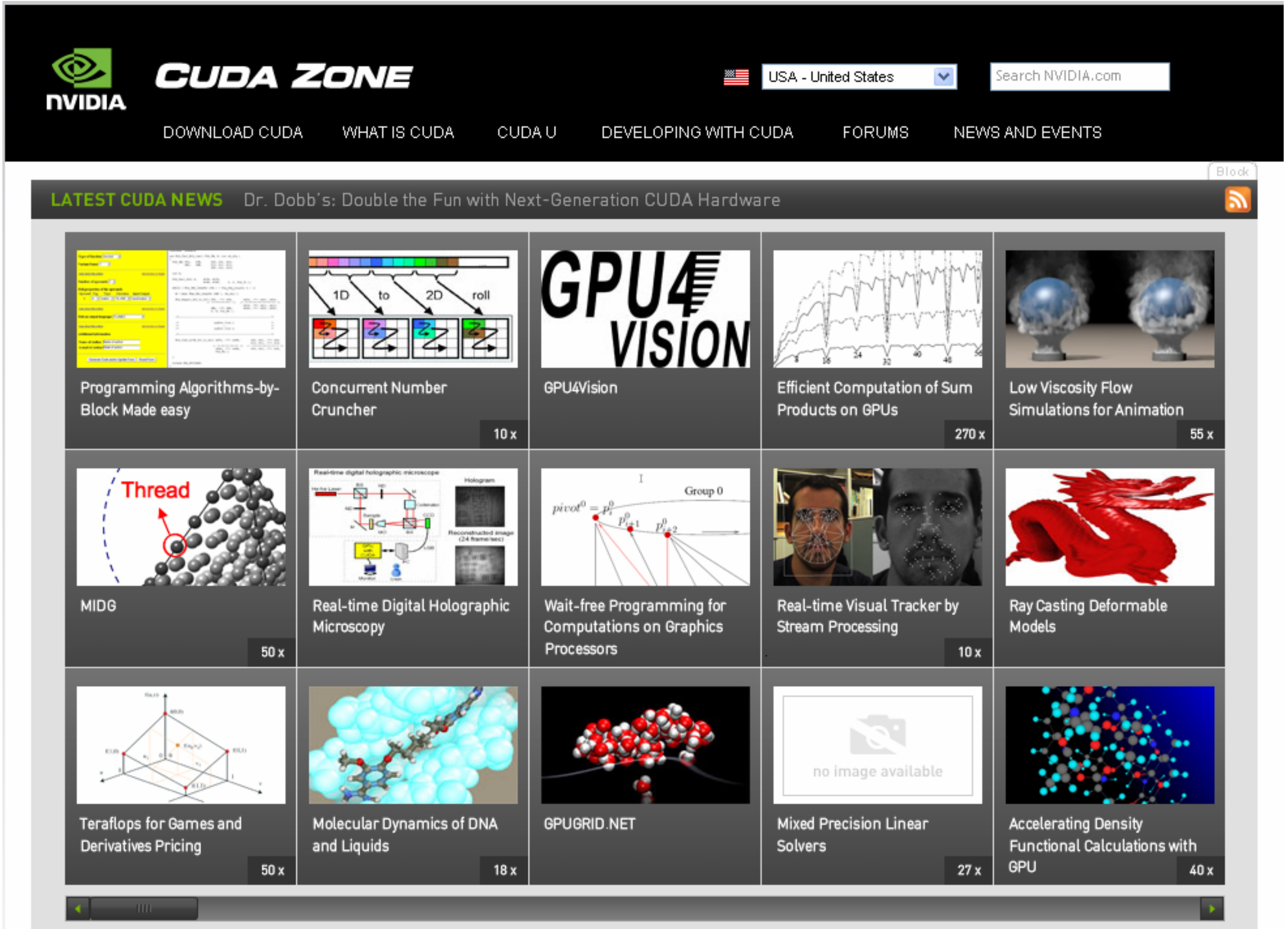
printer-friendly format

recommend to a colleague

**Is CUDA the parallel programming model that application developers have been waiting for?**

### Paradigm

The advent of multicore CPUs and manycore GPUs means that mainstream processor chips are now parallel systems. Furthermore, their parallelism continues to scale with Moore's law. The challenge is to develop mainstream application software that transparently scales its parallelism to leverage the increasing number of processor cores, much as 3D graphics applications transparently scale their parallelism to manycore GPUs with widely varying numbers of cores.



# Motivating Parallelism

- Why Parallel Computing with the rate of development of microprocessors in mind?
  - Trend: Uniprocessor architectures are not able to sustain the rate of realizable performance. Reasons are the for example lack of implicit parallelism or the bottleneck to the memory.
  - Standardized hardware interfaces have reduced time to build a parallel machine based on a microprocessor.
  - Standardized programming environments for parallel computing (for example MPI/OpenMP or CUDA)

# Computational Power Argument – Many transistors = many useful OPS ?

- *„The complexity for minimum component costs has increased at a rate of roughly a factor of two a year. Certainly over short term this rate can be expected to continue, if not increase. Over the long term, the rate of increase is a bit more uncertain, although there is no reason to believe it will remain not constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65000.*" (Moore, 1965)

- 1975: 16K CCD memory with approx. 65000 transistors
- Moore's Law (1975): The complexity for minimum component costs doubles every 18 months
- Does this reflect a similar increase in practical computing power? No! Due to missing implicit parallelism and the unparallelised nature of most applications.

⇨ Parallel Computing

| Prozessor | Transistoren | SPEC-Werte | |
|---|---|---|---|
| | | Ganzzahl | Fließkomma |
| Pentium-III 500 MHz (externer L2-Cache) | 9,5 Mio. | 20,6 | 14,7 |
| Pentium-III 1000 MHz (interner L2-Cache) | 28,5 Mio. | 46,8 | 32,2 |

# Memory Speed Argument

- Clock rates: approx. 40% increase per year
  DRAM access times: approx. 10% increase per year
  Furthermore, #instructions executed per clock cycle increases
  ⇨performance bottleneck
  reduction of the bottleneck: hierarchical memory organization, aiming at many “fast” memory requests satisfied by caches (high cache hit rate)

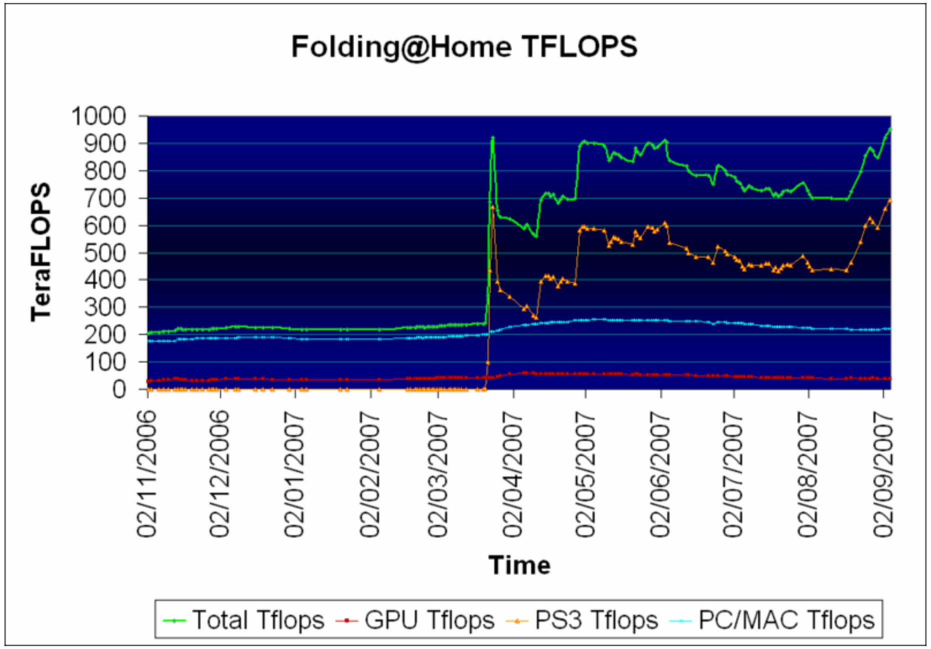
⇨ Parallel Platforms:

- Larger aggregate caches
- Higher aggregate bandwidth to the memory
- Parallel algorithms are cache friendly due to data locality

# Data Communication Argument

- Wide area distributed platforms:
  e.g. Seti@Home, factorization of large integers, Folding@Home, ...

- Constraints on the location of data (e.g. mining of large commercial datasets distributed over a relatively low bandwidth network)



# IBM Roadrunner

Currently (Aug. 2008) the world's fastest computer

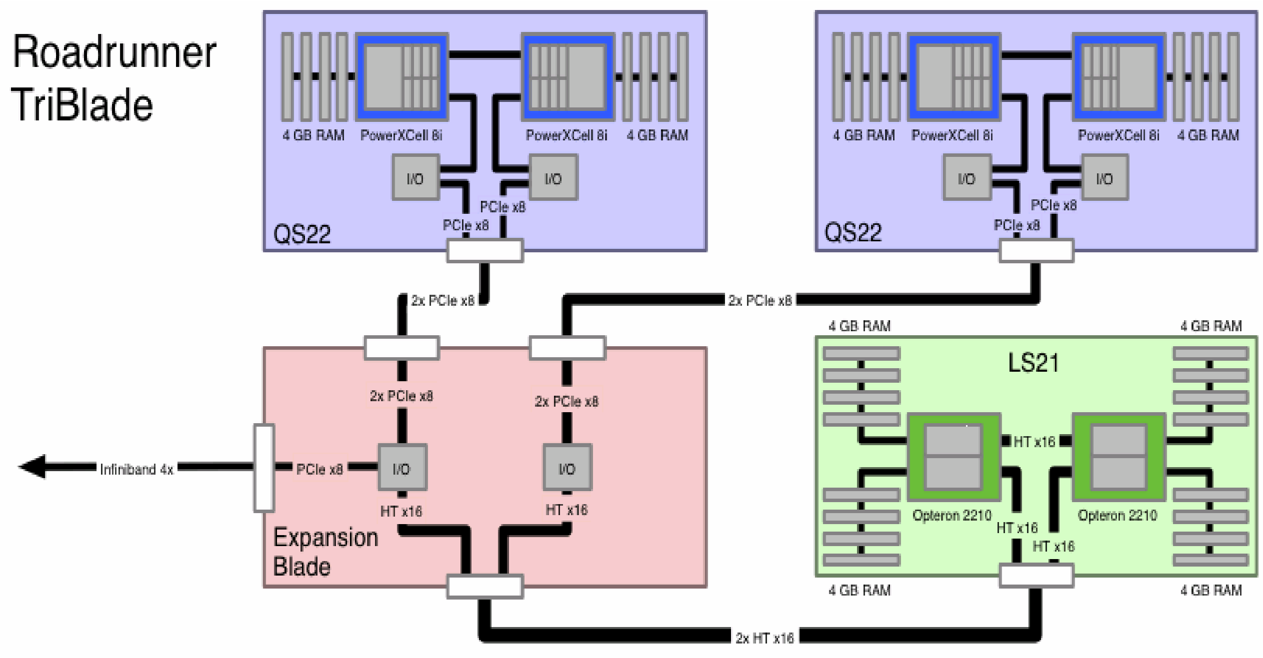
First machine with >1.0 Petaflop performance

No. 1 on the TOP500 since 06/2008

# IBM Roadrunner

Technical Specification:

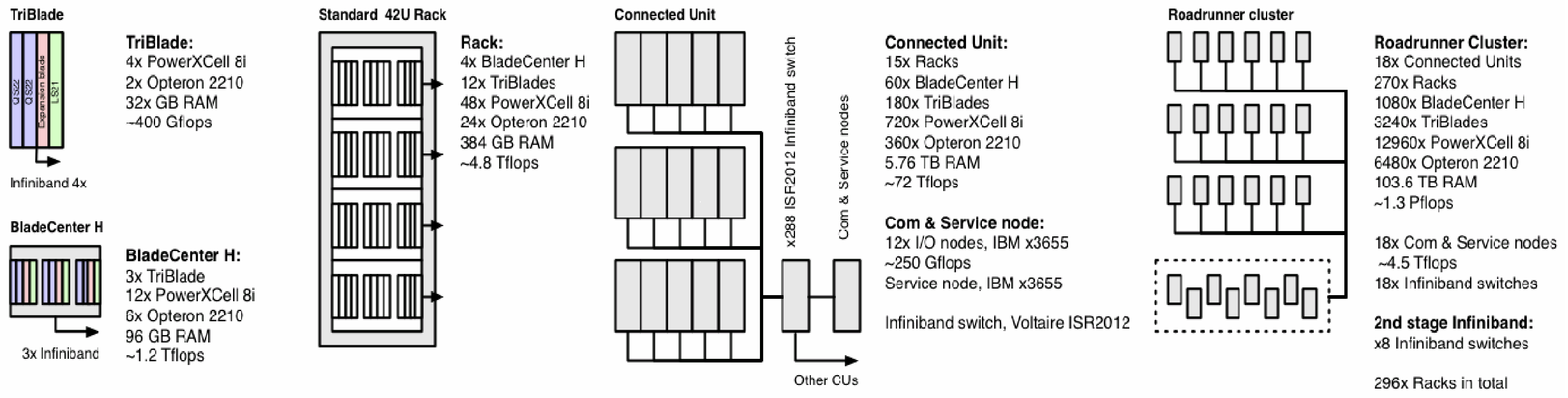
Roadrunner uses a hybrid design with 12,960 IBM PowerXCell 8i CPUs and 6,480 AMD Opteron dual-core processors in specially designed server blades connected by Infiniband



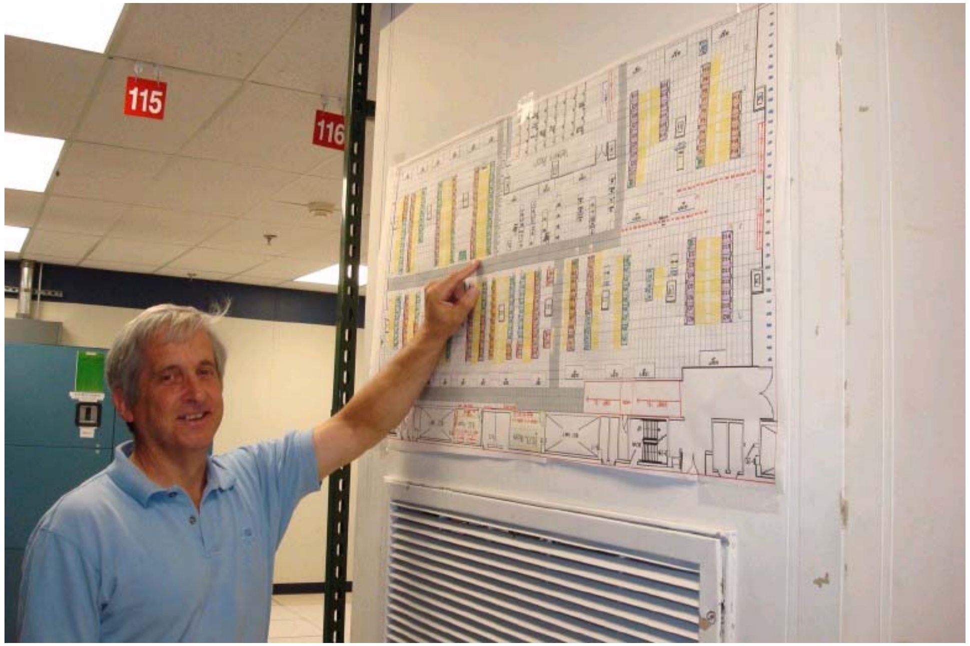
# IBM Roadrunner

## Technical Specification:

- 6,480 Opteron processors with 51.8 TiB RAM (in 3,240 LS21 blades)
- 12,960 Cell processors with 51.8 TiB RAM (in 6,480 QS22 blades)
- 216 System x3755 I/O nodes
- 26 288-port ISR2012 Infiniband 4x DDR switches
- 296 racks
- 2.35 MW power

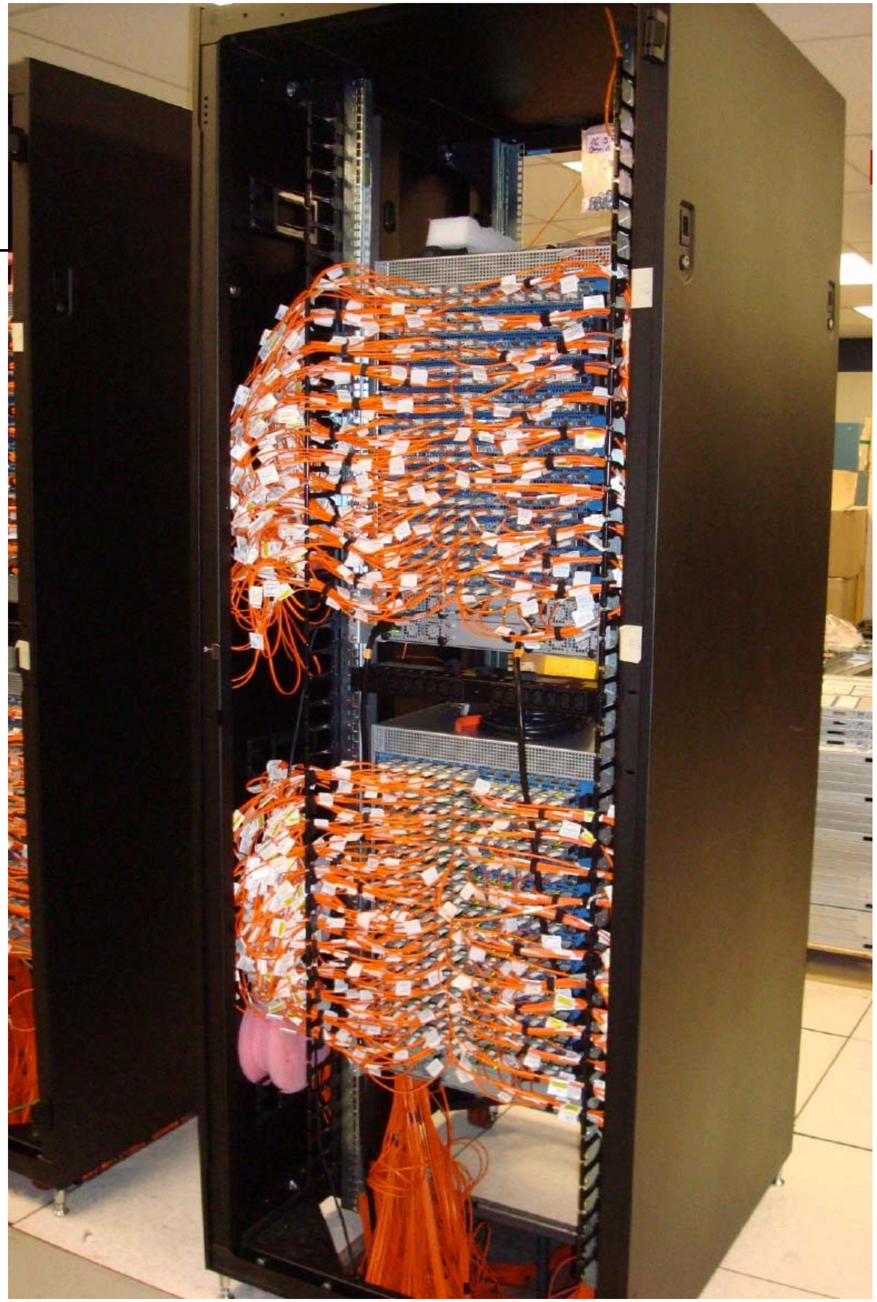


# IBM Roadrunner

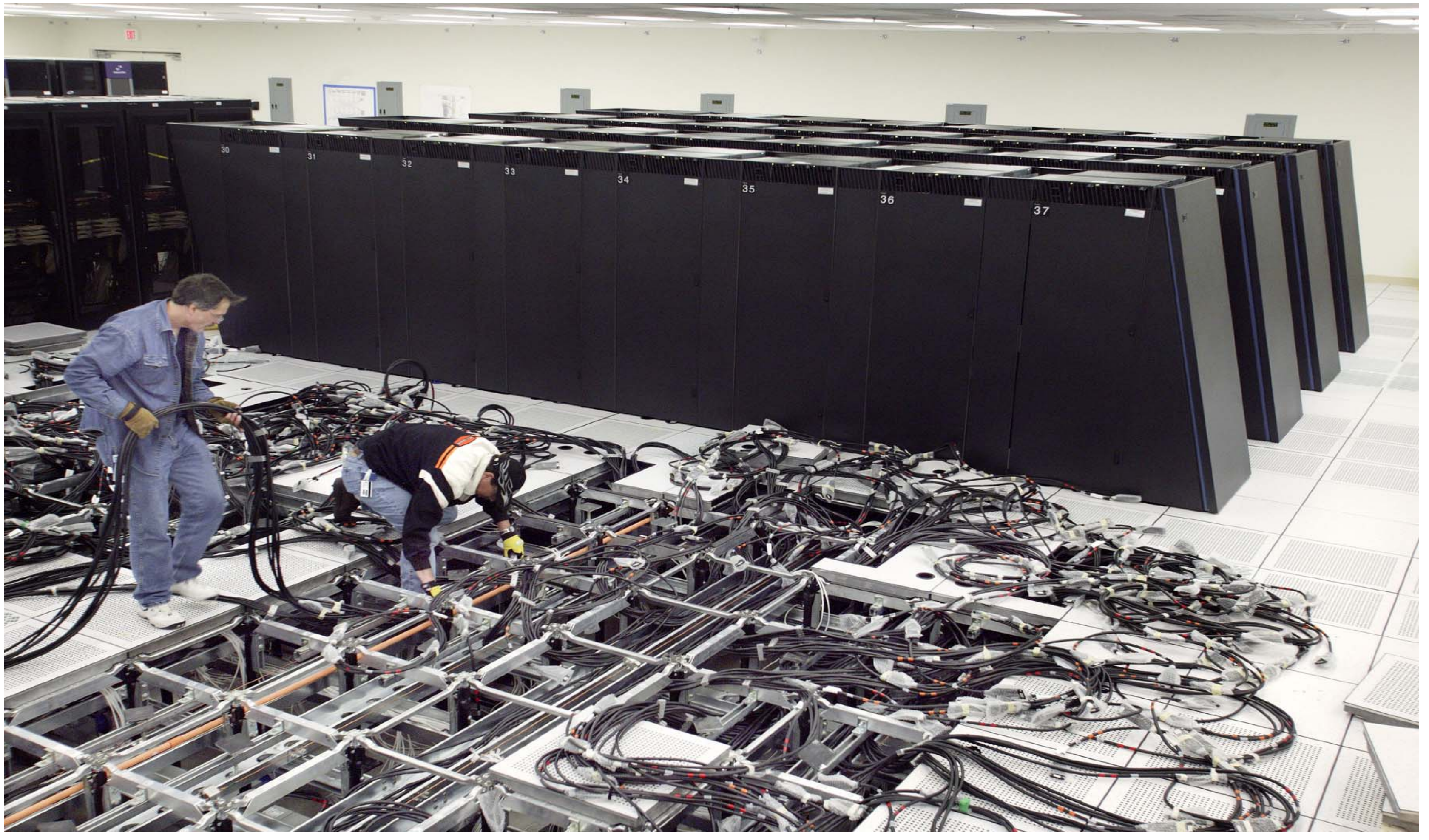



Dr. Don Grice, chief engineer of the Roadrunner project at IBM, shows off the layout for the supercomputer, which has 296 IBM Blade Center H racks and takes up 6,000 square feet.

(source: http://www.computerworld.com)



# 280 TFlops/s : BlueGene/L



# BlueGene/L

# BlueGene/L – System Architecture