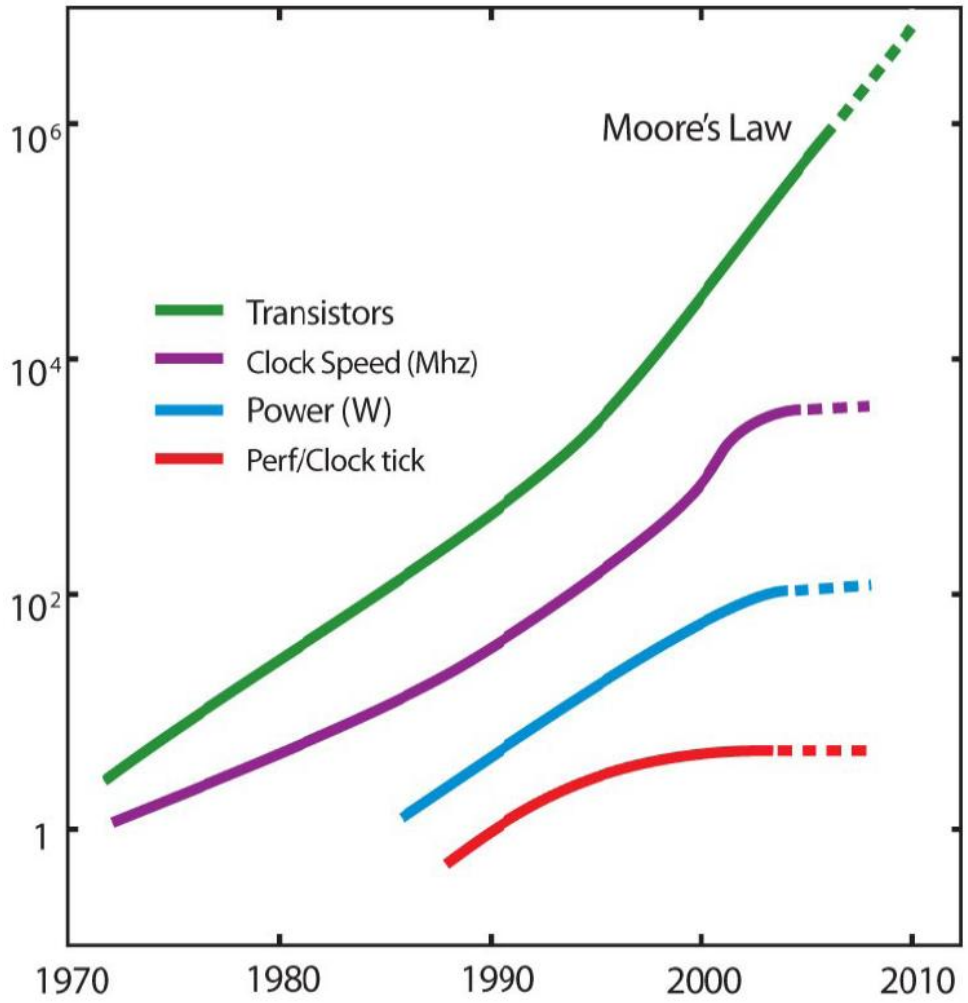
The Beach Law [Gottbrath et al. 1999]

One way of doubling the performance of your computer program is to go to the beach for 2 years and then buy a new computer.

Processor development 1970-2010



"The number of transistors on an integrated circuit for minimum component cost doubles every 24 months"

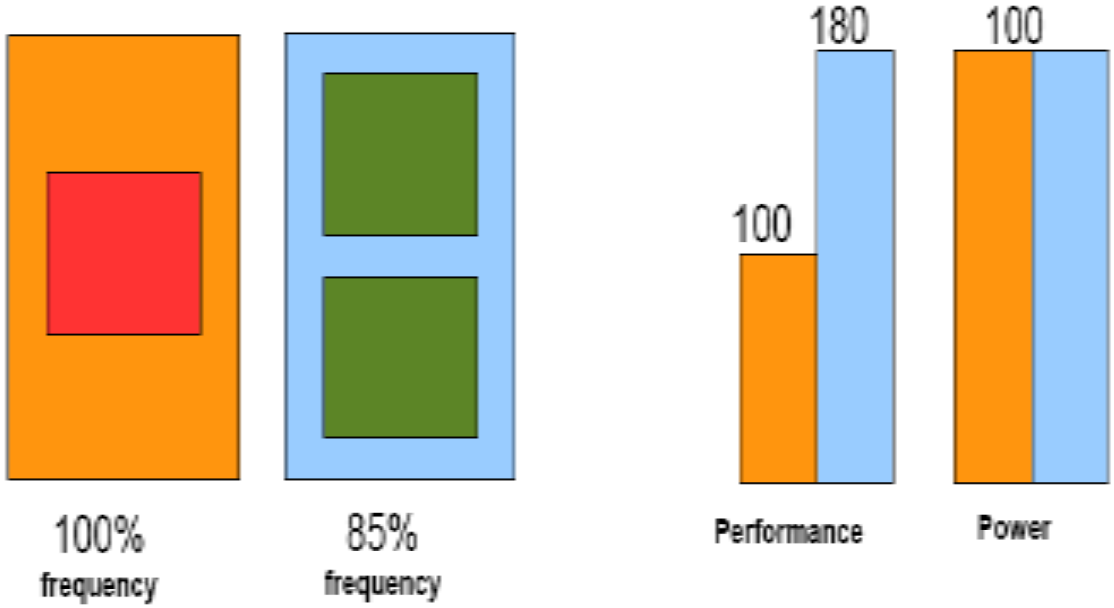
– Gordon Moore, 1965.

What happened?

- Moore's law at work, expected to hold until 2030 ...
- The Beach Law was valid until about 2005 ...
- Heat dissipation etc. stopped it
- PC computing power still benefits from Moore's law
- Multi-core processors for task parallelization (multi-threading, shared memory)
- Accelerators for data parallelization (stream processing)

- Drastic change in the development of processors

Multi-core processors

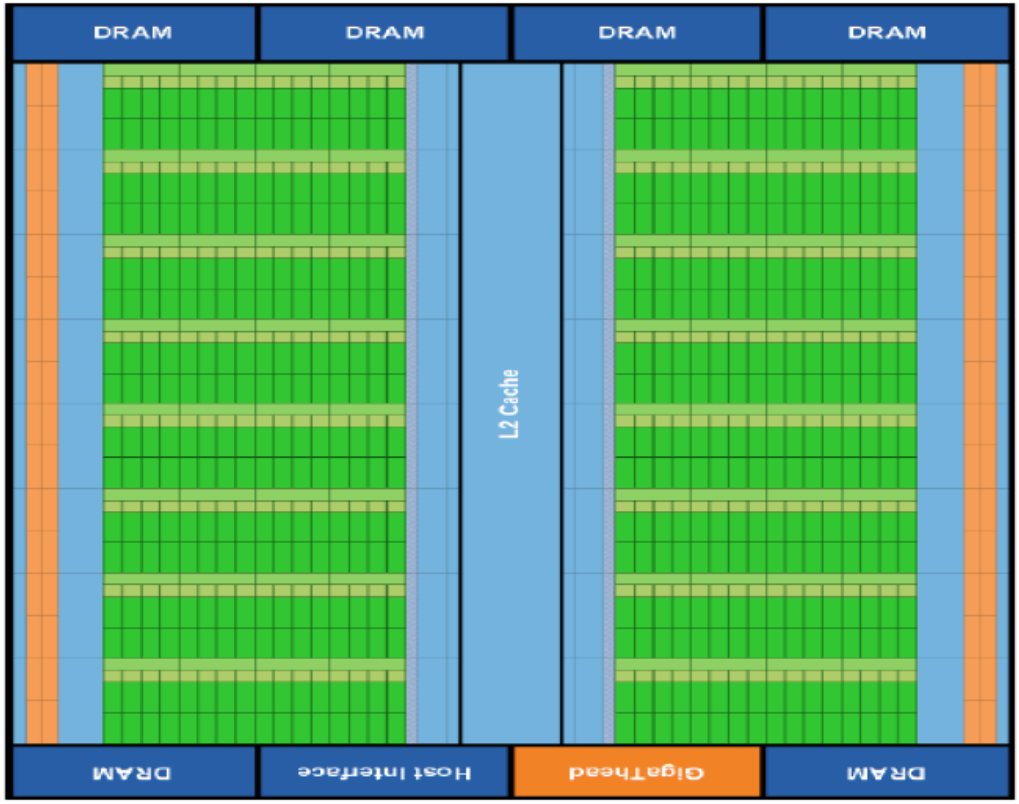


- Heat dissipation varies with clock frequency cubed
- 2 cores, reduced frequency, same heat dissipation
- 70% higher computing performance **if you can exploit it**
- **Sequential programs will run slower**

Stream processing accelerators

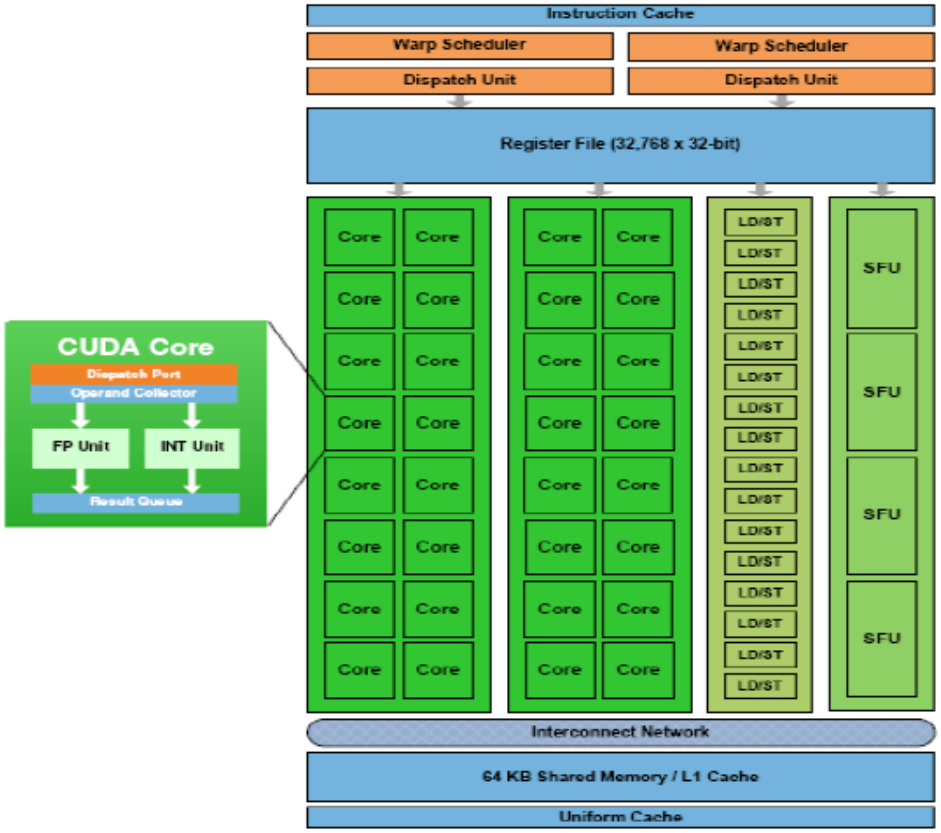
- The graphics card was the origin
- Developent driven by gaming industry
- Computing power increases rapidly
- Programmability improves rapidly
- Libraries, debugging and profiling tools
- Single Program Multiple Data
- Massively parallel, thousands of threads
- You need to
  - understand the architecture
  - worry about code diversion
  - worry about memory latency

The GPU – NVIDIA Fermi Architecture



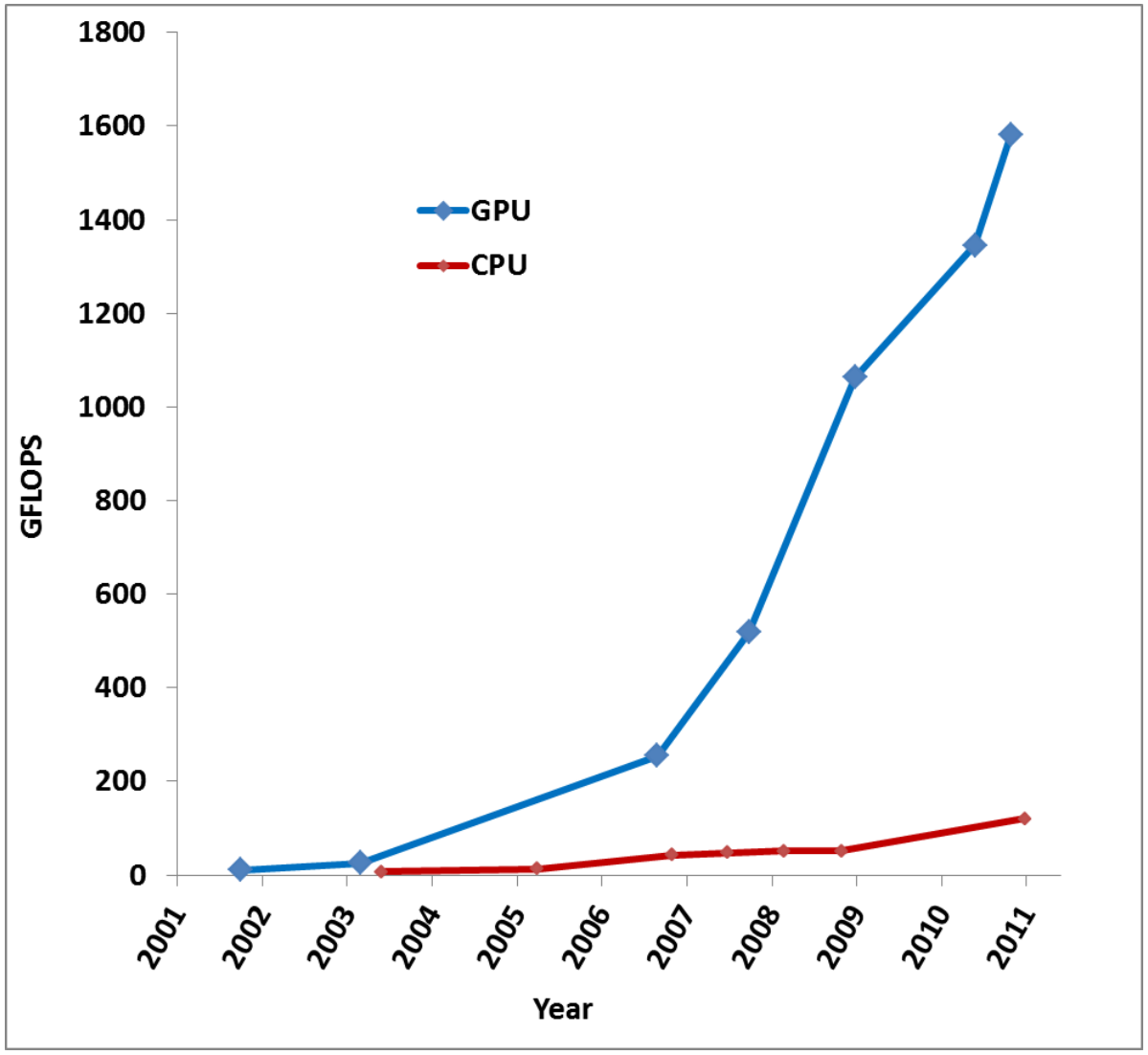
16 streaming multiprocessors are positioned around a common L2 cache

The GPU – NVIDIA Fermi Architecture



Each of the16 Streaming Multiprocessors (SMs) has 32 cores, 512 cores in total. Each core run the same program («kernel»), with individual data and individual code flow (SPMD). Divergence means serialization. Need more threads than cores to hide latency, typically >512 threads for each SM, say 10.000. One may run multiple kernels concurrently.

GPU vs CPU performance



Heterogeneous computing

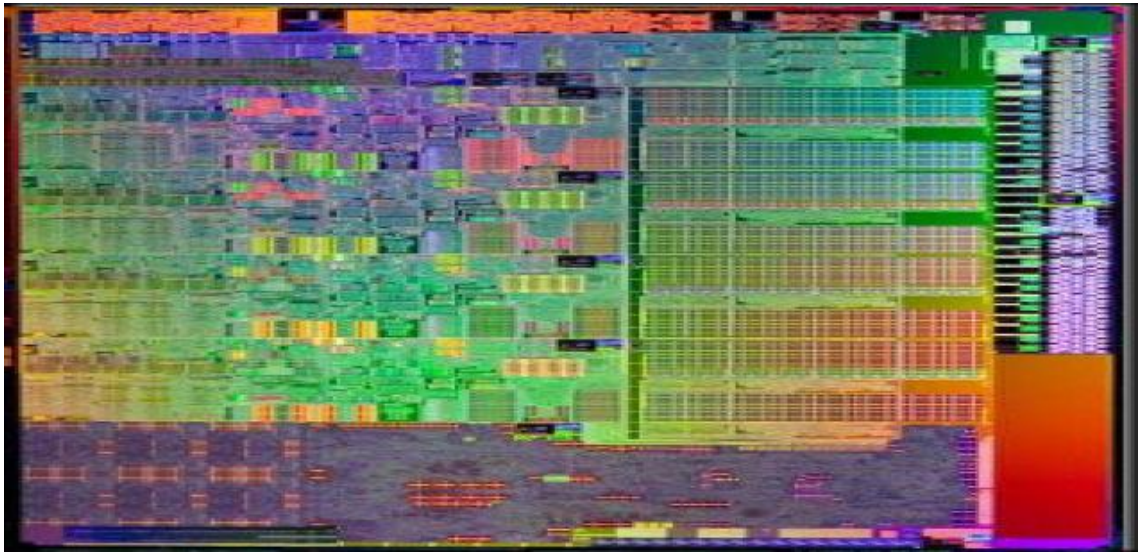
- **Heterogeneous computing systems**:
  electronic systems that use a variety of different types of computational units.
- Current and future PCs are parallel and heterogeneous

"GPUs have evolved to the point where many real-world applications are easily implemented on them and run significantly faster than on multi-core systems. Future computing architectures will be hybrid systems with parallel-core GPUs working in tandem with multi-core CPUs."

Prof. Jack Dongarra
Director of the Innovative Computing Laboratory
The University of Tennessee

Supercomputer on a chip
Single die heterogeneous processors

- AMD Fusion
- Intel Sandy Bridge



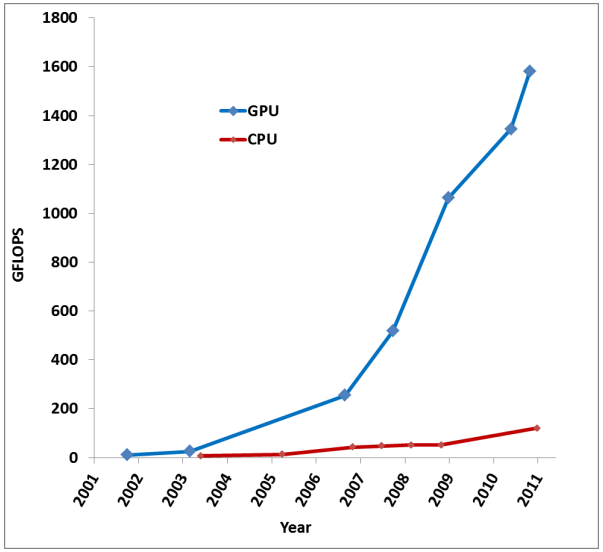
Observations

- «The Beach law» does not hold any more ...
- Fundamental change in the general increase of computing power
- «Moore's law» still at work, until 2030?
- The GPU has become a generally programmable, very powerful device
- Local search up to 1000 times faster on the GPU than on one CPU core
- Every PC will soon have a heterogeneous supercomputer inside
  - multiple cores
  - stream processing accelerator
- Your sequential program can only exploit a small fraction of the power
- Little hope of efficient tools for automatic parallelization
- Providers of basic optimization technology cannot ignore the potential
- Bottlenecks in industry and research
- Opportunities for new ideas in optimization

Why bother?

- Exploit present hardware
- Profit from the future increase of processor power

- Robustness
- Larger-size, more integrated problems
- Real-time applications
- Stochastic models
- Multi-criteria problems
- New ideas in optimization
- Automated parallelization?
- Tool vendors?



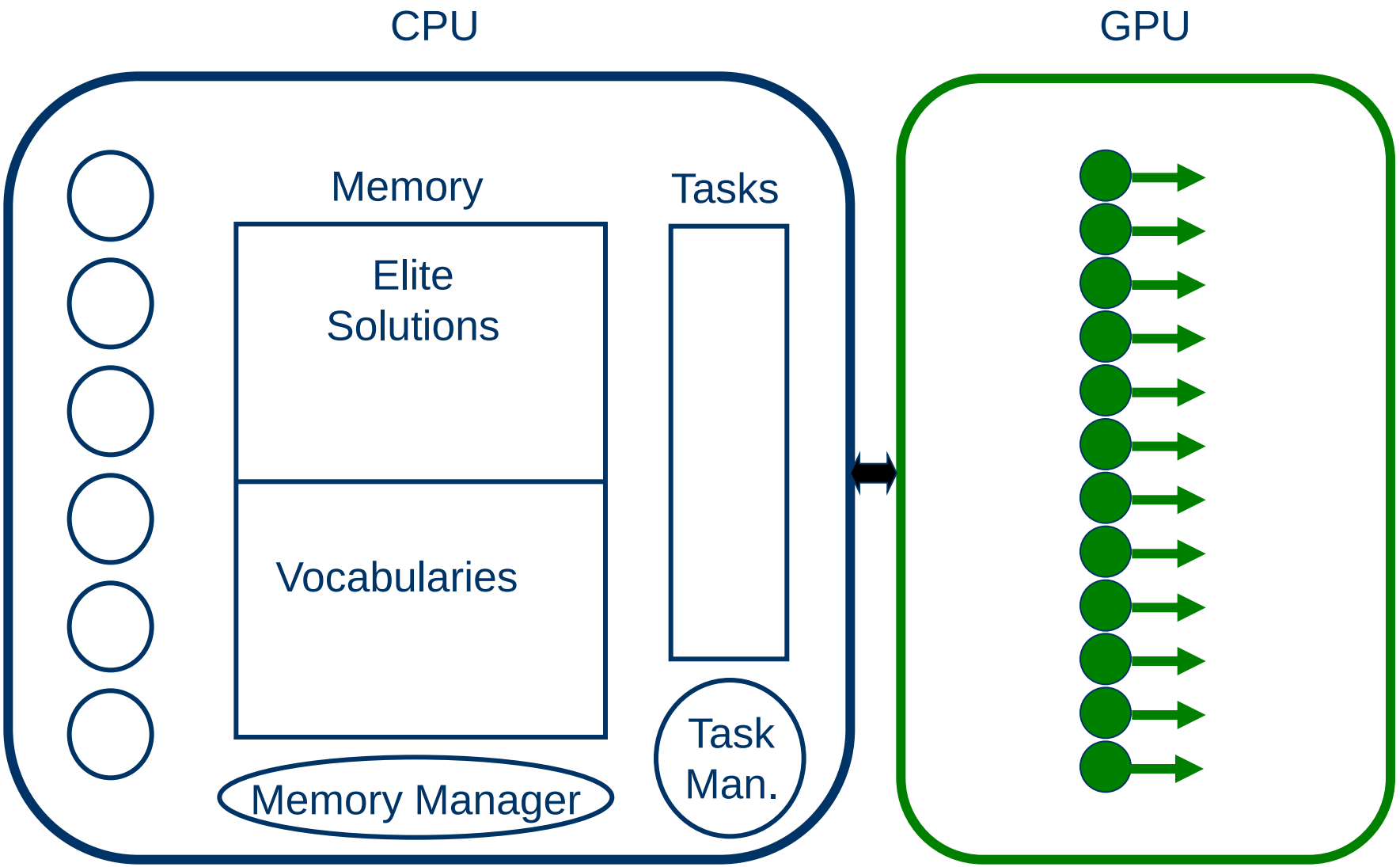
Activities at SINTEF

- PDA-based simulation, geometry, visualization
- Collab project 2009-2012
- Task parallelization of the industrial VRP Solver «Spider»
- Experimental VRP solver: «Camel Spider»
- Project workshops
- META'2010 special session
- JPDC special issue

Ideas – Heterogeneous DOP Computing

- Goal: Balanced use of available computing devices
- Self-adaptation to available hardware
- The GPU is a good intensification machine
  - Local Search
  - Large Neighborhood Search
  - Variable Neighborhood Search
  - ...
- ... but one needs to worry about code diversion and memory management
- CPU used for more «sophisticated» tasks

Sketch of labor division – VRP Solver



Questions

- How to achieve balanced utilization in a self-adaptable way?
- Exact methods?
- How to assess the performance of an optimization algorithm?
- What is «within reasonable time»?