CERN openlab II

Multi-threading and multi-core optimizations

November 15th 2007

Andrzej Nowak

andrzej.nowak at cern.ch

This research project has been supported by a Marie Curie Early Stage Research Training Fellowship of the European Community's Sixth Framework Programme under contract number (MEST-CT-2004-504054)

Modern supercomputing limitations

- The constant need for faster and more capable systems
- Today's options:
  - Frequency scaling techniques of CPUs are nearly exhausted
    - Increasing core frequency does not yield linear performance improvements
    - High power consumption
    - High heat dissipation
  - Parallel architectures introduced; additional "cores" available at a low cost

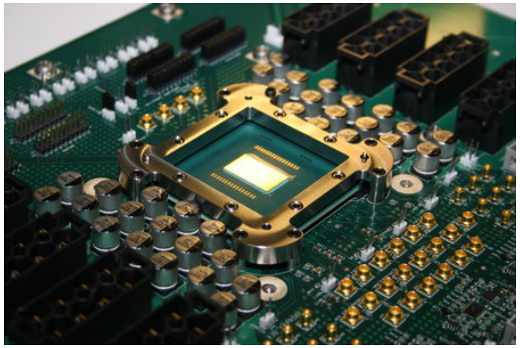
Hardware

> Currently:

- Intel's 45nm Quad-core designs (Penryn)
  - Harpertown (Xeon)
  - Yorkfield (desktop)
- At CERN
  - 2 double-processor boards in a 1U enclosure
  - 2x2x4 = 16 cores per 1U enclosure
  - 32 GB of RAM
  - Only 6 machines per rack, 700W each

> Future:

- Nehalem (from 2008 on)
  - 4-8 cores, 2 threads per core
- Other (i.e. hybrid designs by Intel)
  - Polaris (pure research): 80 cores



The imminent move to multi-core

- The move to heavily multi-core architectures is imminent
  - Advantages:
    - Less power used
    - Less heat dissipated
    - More processing power in a single package
    - Fast communication between cores: nanoseconds with multi-core, 100’s of nanoseconds with SMP
- Timeline:
  - “Today”: 2, 4, 8 cores
  - Heavily multi-core designs are already used in graphics and network processing
  - 16 or 32 cores in general purpose CPUs in the near future
  - Dozens of general-purpose cores might be available in the further future
- How do we prepare for this revolution?

GPUs

- NVIDIA G80
  - First major breakthrough
  - 128 stream processors
  - Chip consumed ~150W
  - CUDA development kit issued
  - 330 GFLOPS (single precision, non-IEEE format)

- NVIDIA G92
  - Currently in development
  - Close to 1 TFLOP
  - 64-bit floating point support
- AMD/ATI NV670
  - 0.5 TFLOP, double precision (64-bit)
- CERN’s Top500 run: 8.3 TFLOPS @ 1360 cores

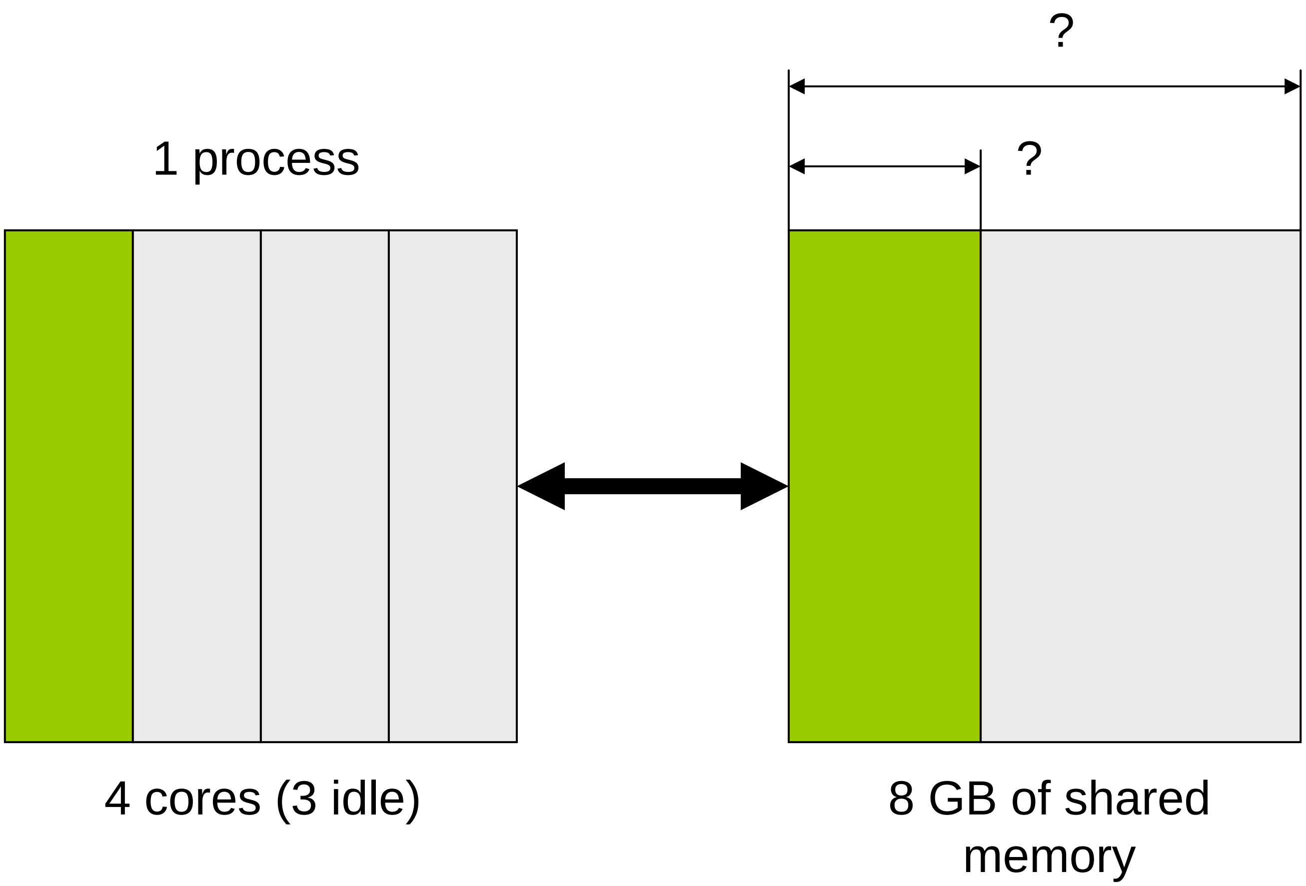
Multi-threading issues

- Exploiting multi-core architectures is a necessity. What are the issues?
  - Can the problem be solved via parallel computing? What is the best approach?
  - The implications of running multiple demanding threads in a single system: some resources might become choking points
    - Memory bandwidth/size
    - System bus
    - Inter-CPU communication
    - Network
    - Hard drive performance
    - Hard drive space

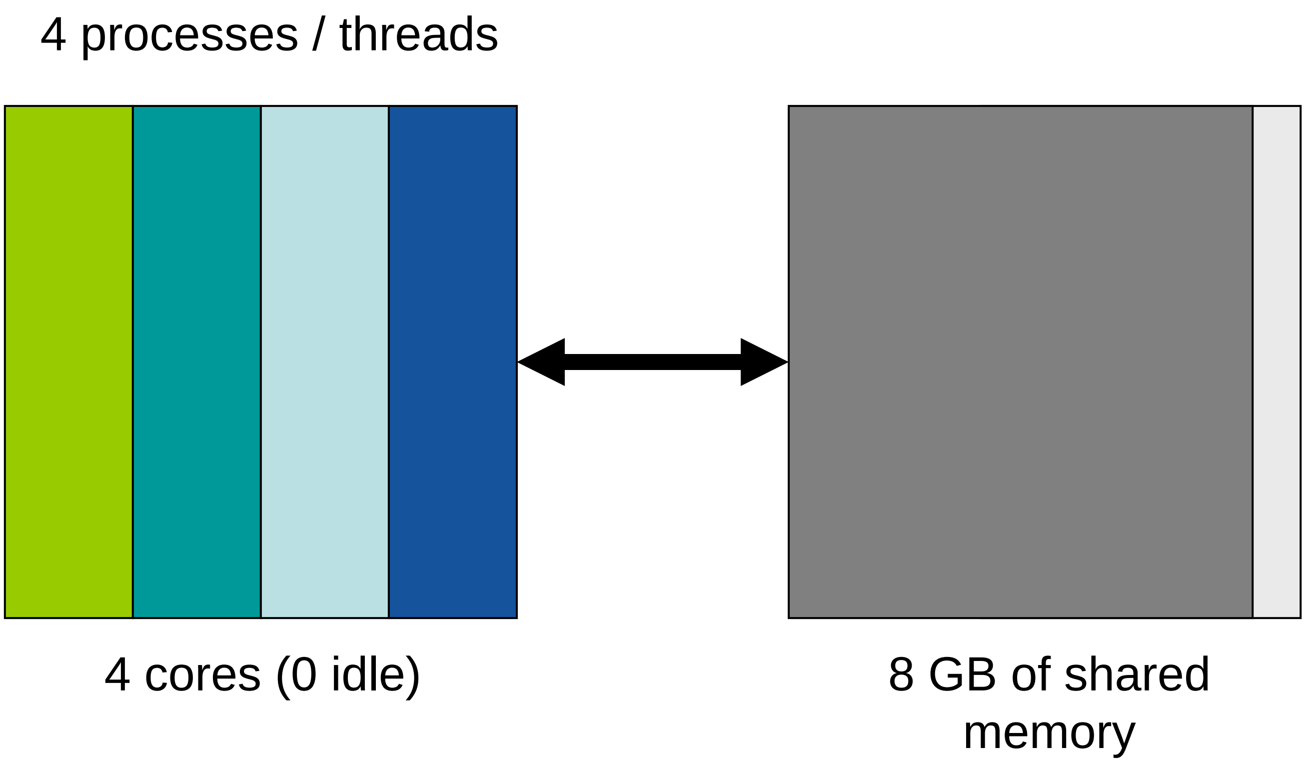
CERN applications

- Many applications at CERN have the following characteristics:
  - CPU-intensive
  - Relatively low amount of RAM transactions
    - but: high RAM usage, up to 1.5GB per instance
  - Embarrassingly parallel (data parallelism)
  - The executable has a small footprint (often fitting into 1MB of cache)
  - Single-threaded
- A lot of “free” processing power is wasted “between the lines”

Example problem illustration – memory limitations



Example problem illustration – memory limitations



Performance monitoring

- System benchmark, based on Geant 4 scientific software
  - Simulates particles passing through matter
  - Real detector geometry from a LHC experiment
  - Real physics processes
  - Loads similar to those expected during LHC operation
- Monitoring using own tool + pfmon

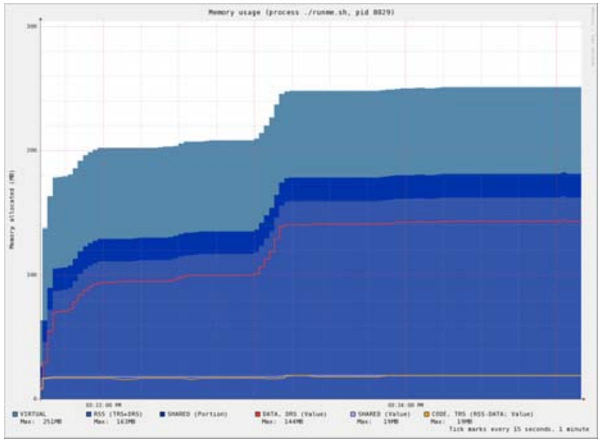
Processing time for 100 events (real time)

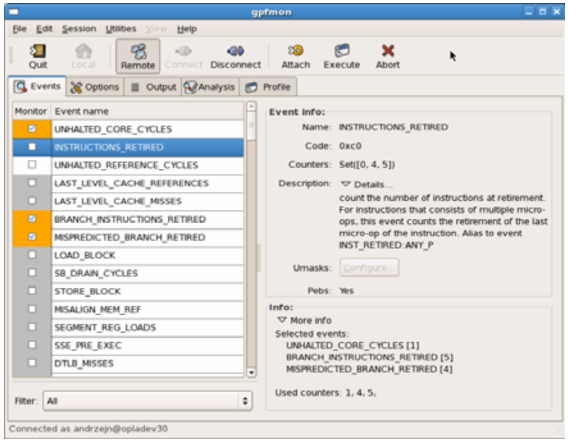
1 process	118s	-	-	-
4 processes	120s	121s	121s	121s

No bottlenecks!

Internal activities in multi-threading

- Benchmarking, performance monitoring
- Tools development
  - pfmon, gpfmon, other
- Technological discussions and analyses
- Training
- Lobbying for multi-threading advancement





Close cooperation with Intel

- Hardware tests and benchmarks
- Hardware pilot programs
- Software benchmarks on new, upcoming or experimental hardware
- Numerous architectural discussions
- Cyclic workshops on multi-threading organized at CERN with the help of Intel
  - Intel speakers

Conclusions

- No easy way to “parallelize” existing software, although efforts are being made
- Numerous tools for programmers simplify common parallelism concepts
  - Intel Threading Building Blocks
  - OpenMP
  - MPI/PVM
  - Emerging technologies: CT (Intel), RapidMind
- The solution for now: multiple independent processes and threads per physical processor
- The solution for tomorrow: threaded software
- Programmer awareness and education is key to achieving good results with multi-core systems
  - Workshops organized with Intel at CERN

CERN openlab

Q&A