SPEEDUP – Optimization and Porting of Path Integral MC Code to New Computing Architectures

V. Slavnić, A. Balaž, D. Stojiljković,
A. Belić, A. Bogojević
Scientific Computing Laboratory
Institute of Physics Belgrade, Serbia
http://www.scl.rs/

Scientific Computing Laboratory

Sep 08, 2009

Overview

- Introduction
- SPEEDUP code
- Tested hardware architectures
- Results
  - Setup
  - Serial SPEEDUP code
  - MPI SPEEDUP code
  - Modified SPEEDUP code
  - Cell SPEEDUP code
- Comparison of hardware performance results
- Conclusions

Scientific Computing Laboratory

Introduction

- SPEEDUP code is used for numerical studies of Quantum Mechanical systems, properties of BECs and ultra-cold atomic gases
- Porting of the code enables its use on a broader set of computing resources
- Code optimization allows us to
  - Fully utilize computing resources
  - Eliminate bottlenecks in the code
  - Use different architectures in a proper way
  - But, it must be done carefully (verification!)
- Possibility of benchmarking of different hardware platforms
- Use results for planning of hardware upgrades

Scientific Computing Laboratory

SPEEDUP CODE (1/2)

- Monte Carlo simulations are natural choice for numerical studies of relevant physical systems in the functional formalism – Path Integral Monte Carlo
- Speedup code calculates transition amplitudes using the effective action approach

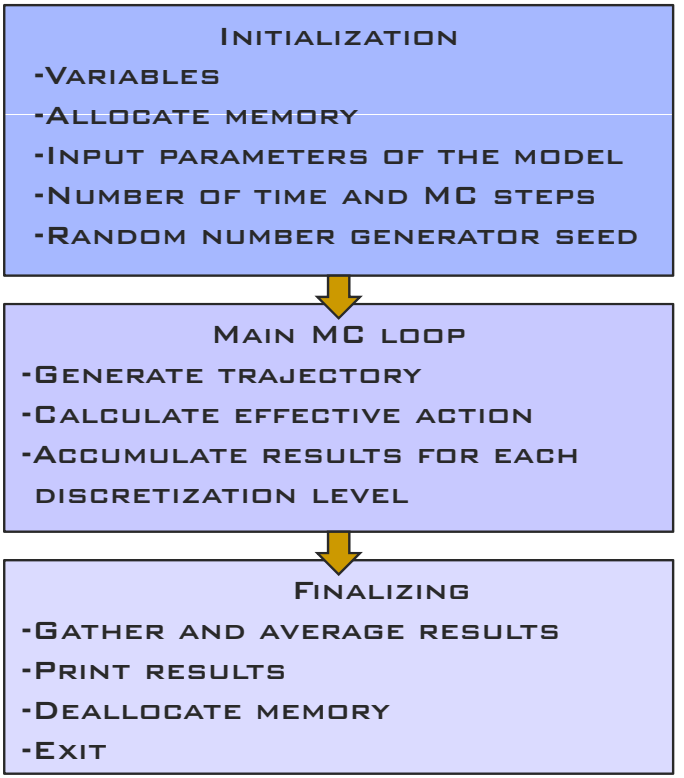
$$A_N(i;f;T)=\left(\frac{1}{2\pi\varepsilon_N}\right)^{N/2}\int dq_1...dq_{N-1}e^{-S_N}$$

- It is able to calculate partition functions and expectation values
- It can be also used to extract information about the low-lying energy spectra of quantum systems

Scientific Computing Laboratory

SPEEDUP CODE (2/2)

- ALGORITHM



- GOOD RNG IS ESSENTIAL – WE USE SPRNG

SCIENTIFIC COMPUTING LABORATORY

Tested architectures

- IBM BladeCenter with 3 kinds of server systems in the HPC H-type chassis:
  - HX21XM blade Server
    - Intel Xeon based
    - 2 quadcore 5405 processors (SSE4.1)
    - ICC and GCC compilers used
  - JS22 blade server
    - POWER6 based
    - 2 dualcore processors supporting multithreading and AltiVec extensions
    - IBM XLC/C++ and GCC compilers used
  - QS22 blade server
    - Cell B/E architecture – 2 PowerXCells 8i on board
    - 1 PowerPC Processor Element (PPE)
    - 8 Synergetic Processing Elements (SPEs)
    - IBM XL C/C++ Compiler for Multicore Acceleration and GCC compilers used

Performed tests

- Nmc=5120000 MC samples
- boundary conditions for the transition amplitude
  - q(t=0)=0
  - q(t=T=1)=1
  - zero anharmonicity
  - level of effective action p=9 for the quartic anharmonic oscillator
- Same seed for SPRNG generator used for easy verification of the obtained results

Scientific Computing Laboratory

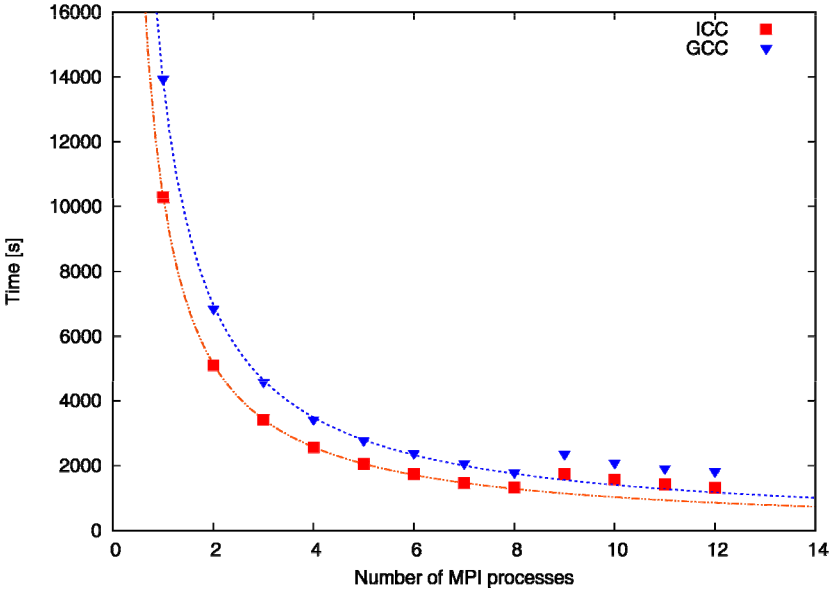
Serial SPEEDUP results

Compiler / Platform	GCC	ICC	XLC
Intel	(13760±50) s	(10160±30) s	
POWER6	(17000±10) s		(1900±10) s
Cell	(49410±50) s		(14020±20) s

- Significant Increase in the speed when platform-specific compiler is used
- POWER performance dominates in this benchmark
- Cell is no match when only PPE is used (without the use of SPEs)

Scientific Computing Laboratory

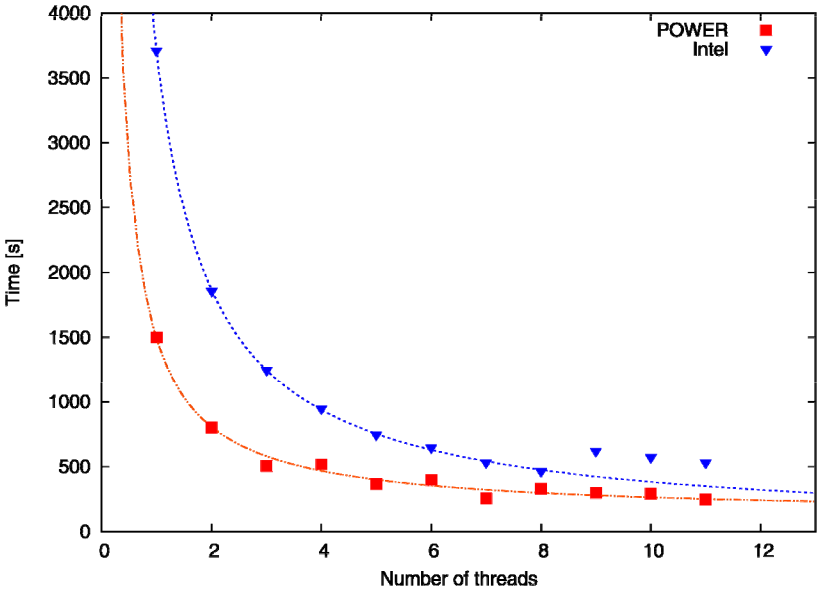
MPI SPEEDUP RESULTS



- EXCELLENT SCALABILITY WITH THE NUMBER OF MPI PROCESSES
- INTERESTING BEHAVIOR WHEN THE NUMBER OF MPI PROCESSES ≥ 9
- MINIMAL EXECUTION TIME OF 1320S

SCIENTIFIC COMPUTING LABORATORY

Modified SPEEDUP results



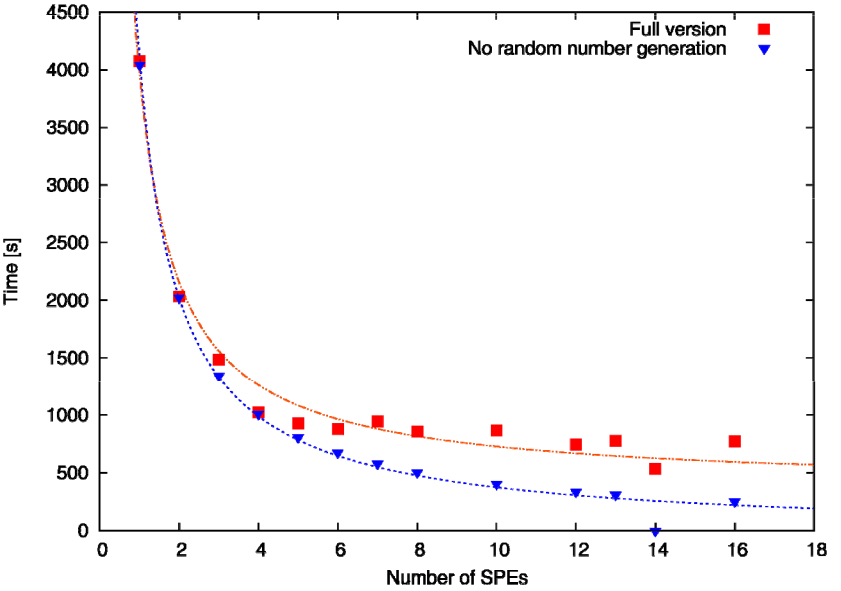
- Implemented as a threaded version using POSIX threads (pthreads)
- Each thread calculates Nmc/Num_threads
- Intel has better relative increase in the speed (2.8x comparing to POWER6's 1.3x)
- Best results: POWER6 250s, Intel 460s

Cell SPEEDUP results (1/3)

- Heterogeneity of the architecture required the slight rearrangement of the code
- Same code is executed on all SPEs
- Each SPE performs Nmc/Number_of_SPEs MC steps
- No SPRNG library for SPEs!!!
- Pthreads on PPE for control of SPEs and RNG generation
- DMA transfers of generated random trajectories from PPEs to SPEs
- Synchronization with mailbox technique

Scientific Computing Laboratory

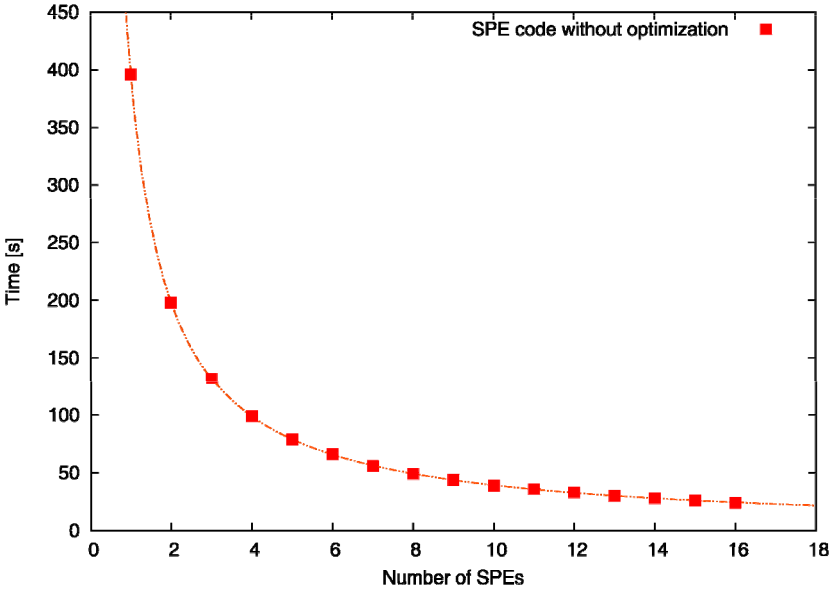
Cell SPEEDUP results (2/3)



- Saturation of the performance around 4 SPEs caused by RNG
- Communication does not have significant impact on the execution time
  - Tested with RNG only, for verification
- Test result: 750s; ideal time: 250s

Scientific Computing Laboratory

Cell SPEEDUP results (3/3)



- To fully utilize all SPE capabilities, one has to extend SPE calculation time
  - Increase in the effective action level p
  - We demonstrate this by compiling the code without optimization
- perfect scaling when PPEs have enough time for RNG

Comparison of results

Intel	POWER6	Cell	Cell ideal
460s	250s	750s	250s

- Results for Intel and POWER6 are obtained using modified SPEEDUP code
- Cell ideal time corresponds to the full utilization of SPEs (estimated)

Scientific Computing Laboratory

Conclusions

- POWER6 and Intel optimization is done using threaded version of the code
- Cell platform requires more complex changes of the code
- platform-specific compilers always give much better performance
- SPEEDUP easily optimized on the POWER6 platform, with superior performance
- Good performance and scalability for Intel platform
- Same level of performance as POWER6 with higher calculation times for Cell
- Future work: porting of SPRNG library to SPEs and implementation of platform-specific instructions (vectorization) for each tested platform

Scientific Computing Laboratory