Increasing efficiency of DaCS programming model for heterogeneous systems

9th INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING
AND APPLIED MATHEMATICS
September 11-14, 2011
Toruń, Poland

Maciej Cytowski, Marek Niezgódka

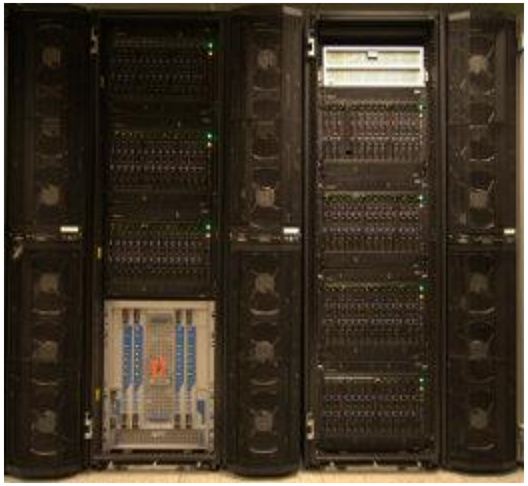
Interdisciplinary Centre for Mathematical and
Computational Modeling
University of Warsaw

Email: m.cytowski@icm.edu.pl

Topics

- Introduction
- Increasing efficiency of DaCS Programming Model
- Use case scenarios

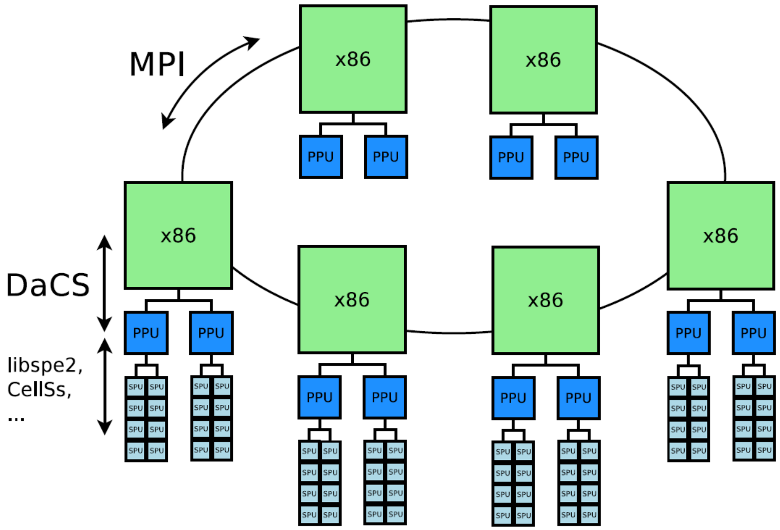
PowerXCell8i Hybrid Environment



- IBM PowerXCell8i – the enhanced Cell processor
- Nautilus Hybrid System
  - 75 IBM QS22, 2xPowerXCell8i, 8GB RAM
  - 18 IBM LS21, Quad-Core AMD Opteron, 32GB RAM
- No PowerXCell8i successors planned
- Still many advantages: single and double precision performance, energy efficiency
- Nautilus and Green500 List
  - 1st Place - November 2008 and June 2009
  - 16th Place – Little Green500, November 2010

IBM DaCS Programming Model

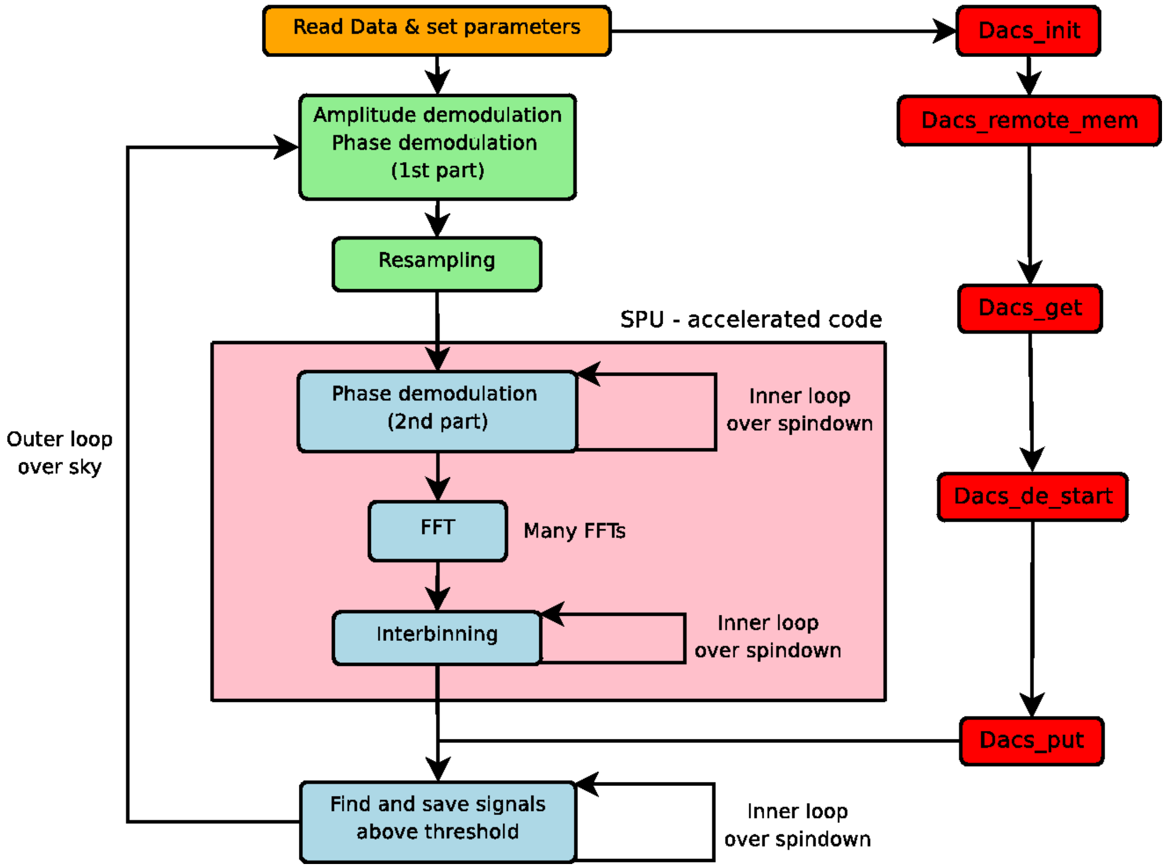
- IBM DaCS – Data Communication and Synchronization library and runtime
- Supports development of applications for heterogeneous systems based on PowerXCell8i and x86 architectures
  - Resource and process manager
  - Data transfers
  - Synchronization
  - Error handling



- Multi-level Parallelism:
  - MPI accross hybrid nodes
  - DaCS on hybrid nodes
  - Libspe2, CellSs, OpenMP, OpenCL on accelerator
- Developed for hybrid environments like Roadrunner (LANL) and Nautilus (ICM)

Example: IBM DaCS Programming Model

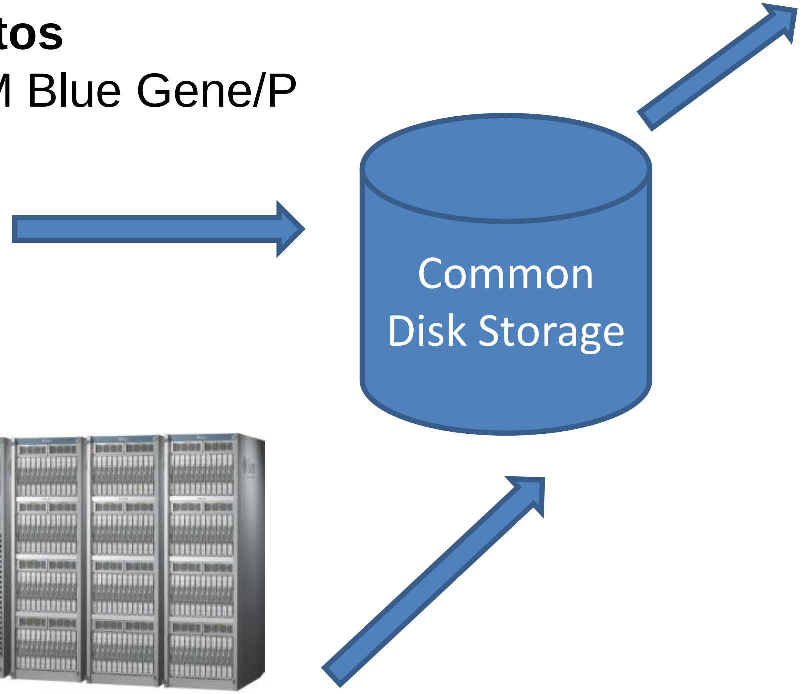
- Run the application on x86 core and offload some of its parts on PowerXCell8i.



ICM’s HPC Environment

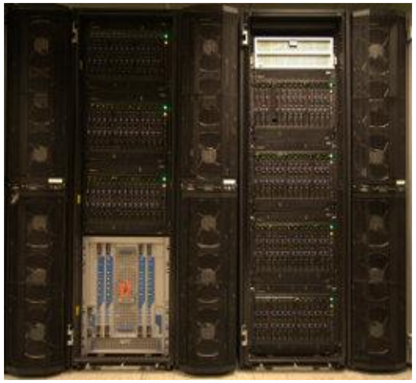
Computational systems

Notos
IBM Blue Gene/P



Halo²
Sun Constellation System

Post-processing and visualization system



Nautilus
Hybrid x86 & Cell

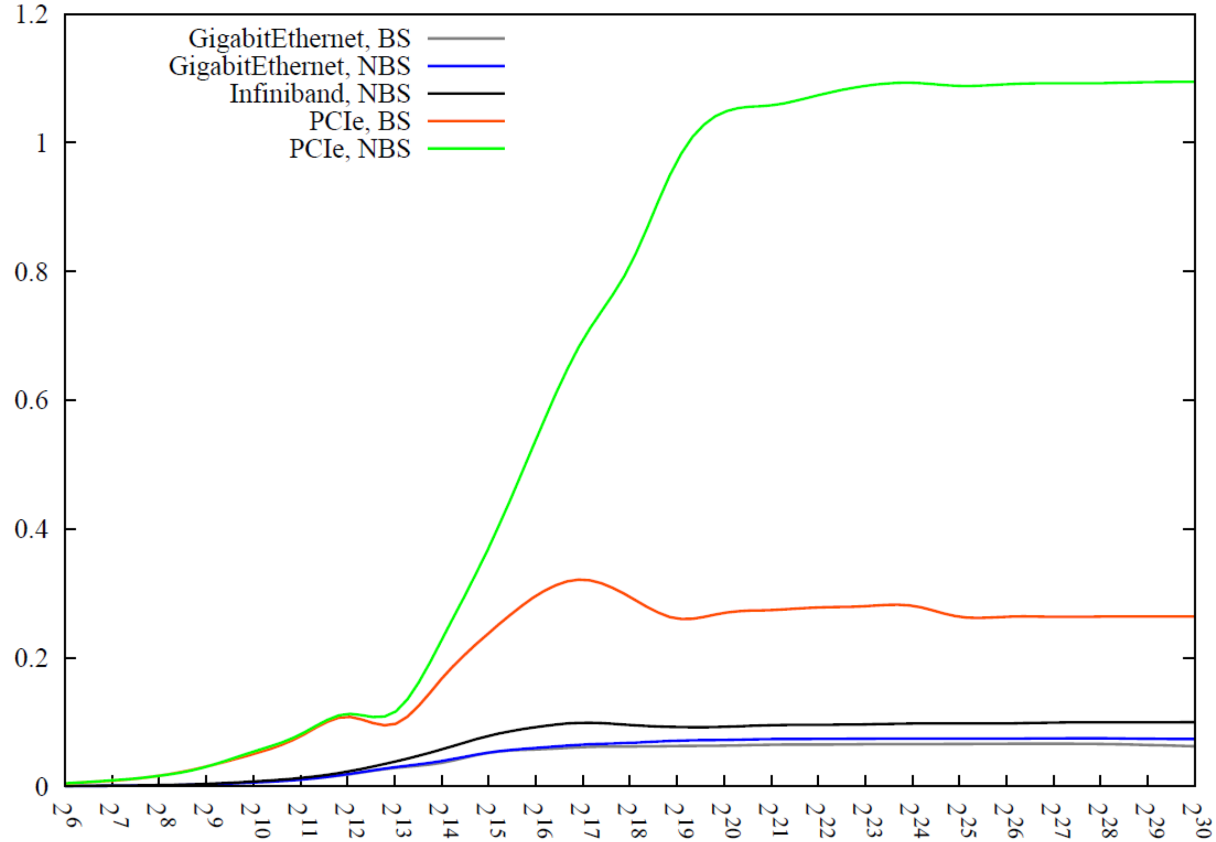
Performance Benchmarking of DaCS

- **A common future of heterogeneous systems**: bottleneck introduced by the data transfers crossing the accelerator boundary
- The computational granularity and performance of compute kernels must be carefully measured and compared with data transfers performance

- **The benchmark program**: PING-PONG between host and accelerator
- **Systems in use**: Roadrunner architecture (Rochester, USA), Nautilus (ICM)
- **Note**: host and accelerator CPUs have different Endianess (additional byte-swap step is needed)
- DaCS library includes its own byte-swapping mechanism
- **Communication flags**: DACS_BYTE_SWAP_DOUBLE_WORD and DACS_BYTE_SWAP_DISABLE

Performance Benchmarking of DaCS

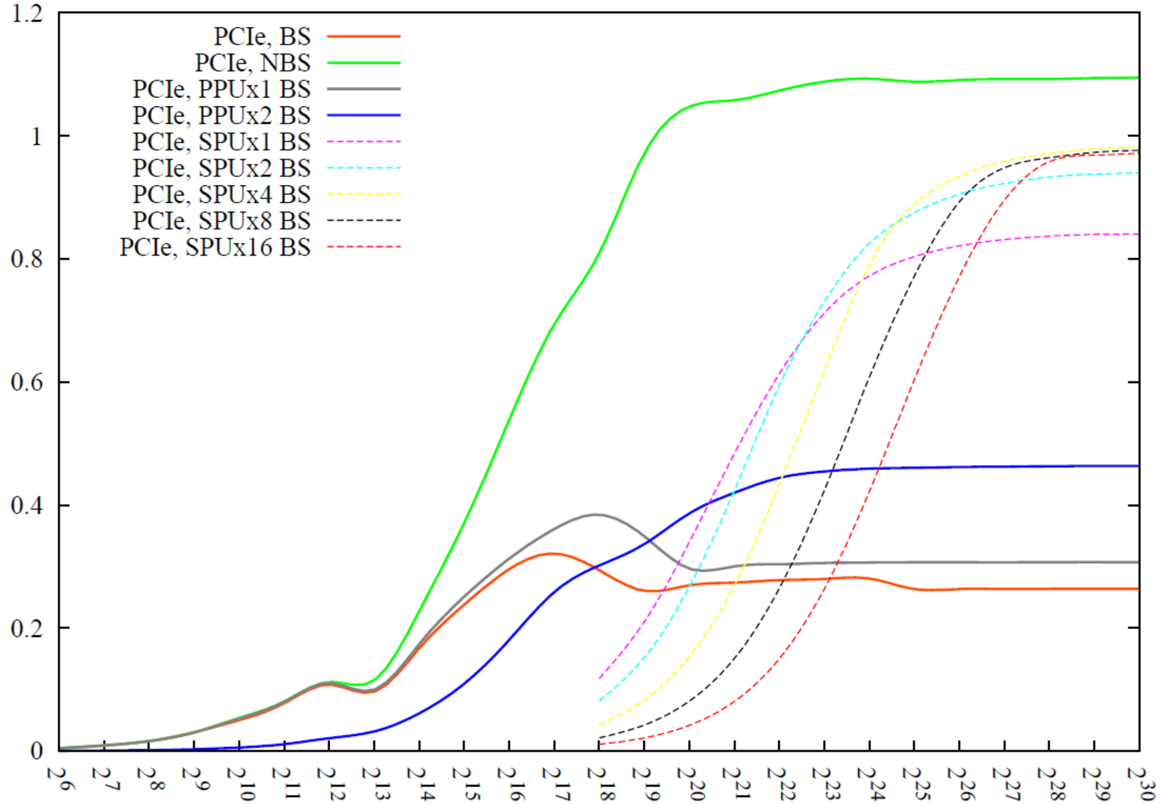
- PING-PONG Performance Tests



Optimized Byte-Swapping

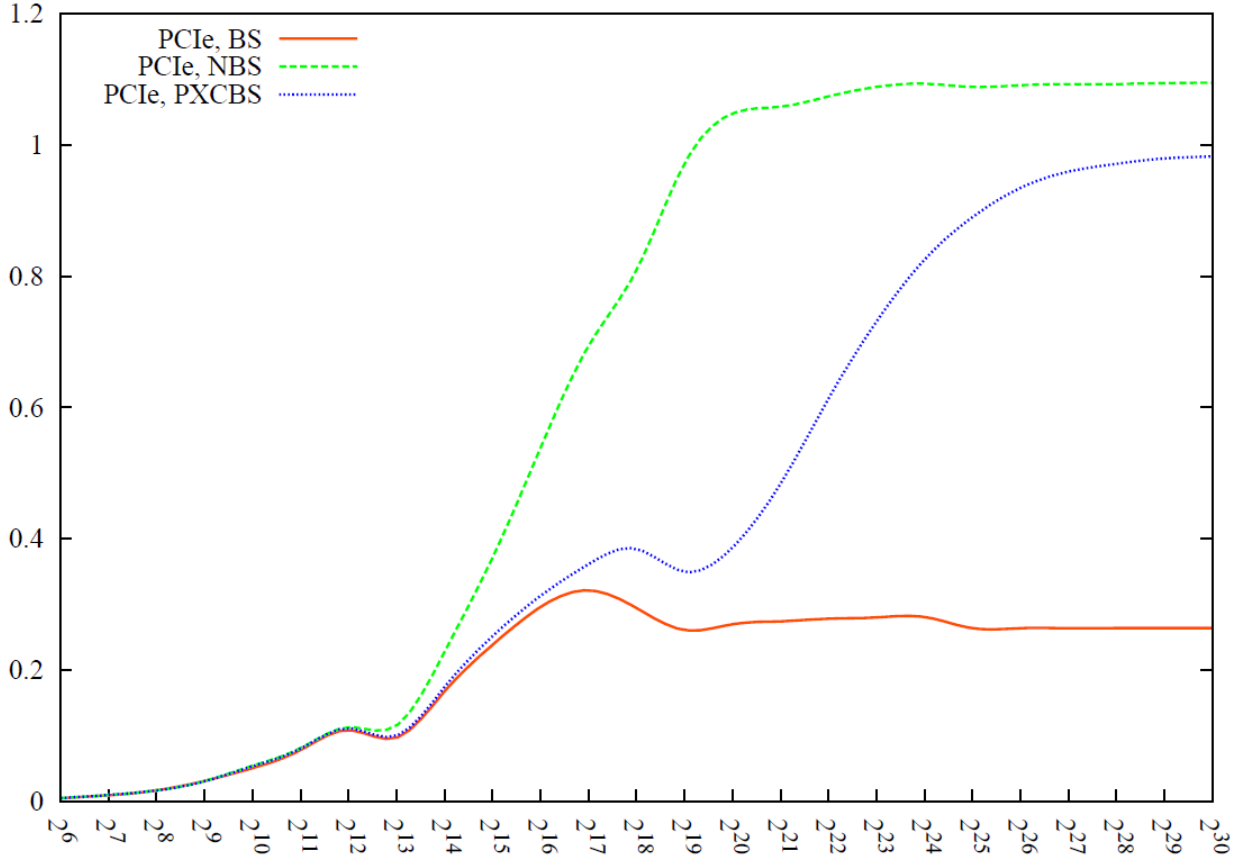
- **Simple idea**: For large data transfers byte swapping could be optimized via vectorization or parallelization on SPUs.
- Development steps:
  - 1,2,4,16 SPUs SIMD versions
  - PPU SIMD and dual-threaded PPU SIMD versions

Kernel version	2^{14} bytes	2^{20} bytes	2^{30} bytes
1 threaded PPU	9 usec	1672 usec	1677745 usec
4 SPU threads	11 usec	1520 usec	99159 usec
AMD Opteron	29 usec	1696 usec	1716653 usec

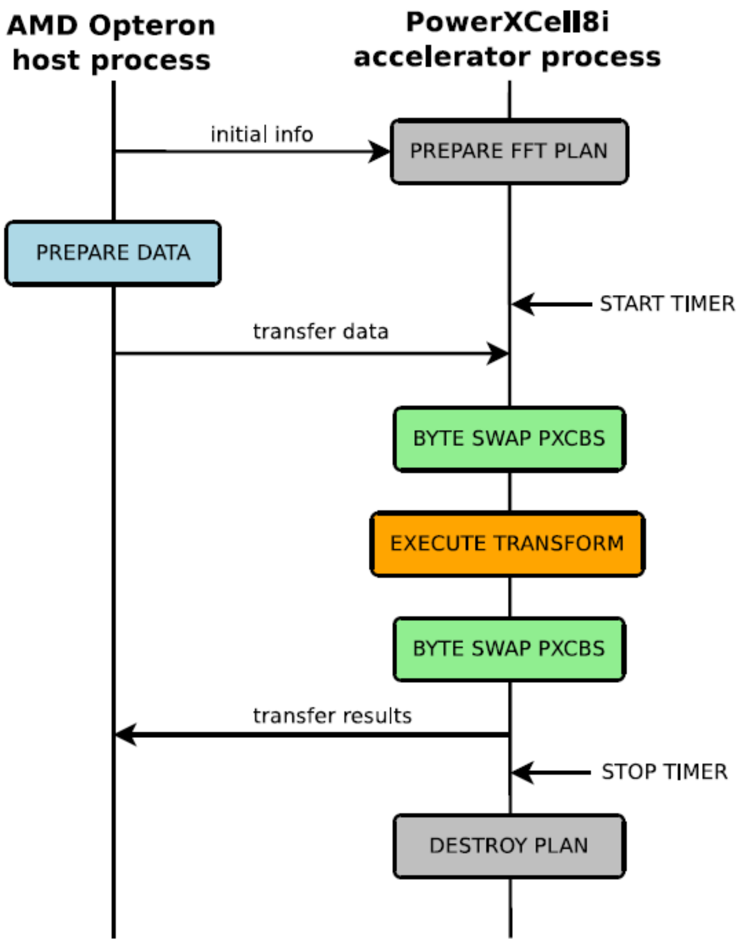


Results: Optimized Byte-Swapping

- Resulting **PXCBS library** is a combination of PPU and SPU implementations used for different transfer sizes



Use Case 1: Hybrid FFTW



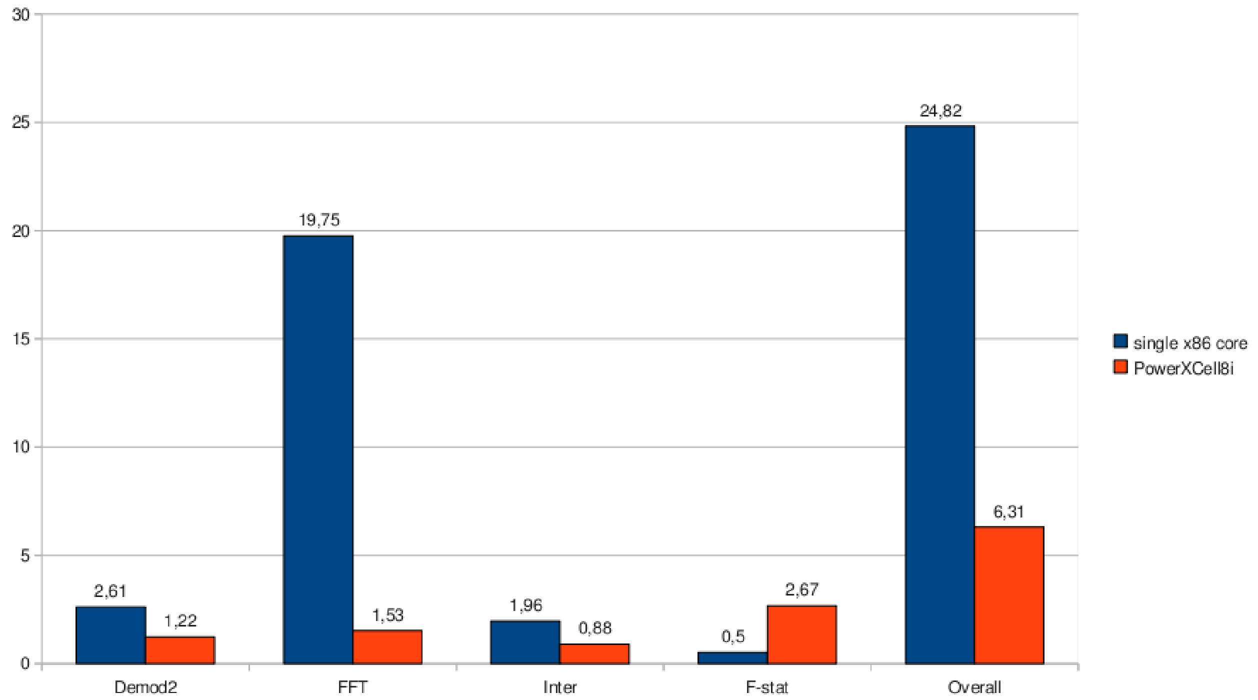
1D FFT size	x86	DaCS PCI		DaCS PCI + PXCBS	
		Comm.	Wall.	Comm.	Wall.
131072	0.025s	0.016s	0.018s	0.009s	0.011s
262144	0.076s	0.032s	0.035s	0.014s	0.018s
524288	0.102s	0.064s	0.067s	0.024s	0.030s
1048576	0.210s	0.127s	0.141s	0.043s	0.057s
2097152	0.446s	0.254s	0.288s	0.079s	0.113s
4194304	0.924s	0.503s	0.704s	0.146s	0.347s
8388608	1.838s	1.007s	1.153s	0.282s	0.425s

2D FFT size	x86	DaCS PCI		DaCS PCI + PXCBS	
		Comm.	Wall.	Comm.	Wall.
256x256	0.009s	0.009s	0.009s	0.006s	0.006s
512x512	0.073s	0.033s	0.047s	0.015s	0.029s
1024x1024	0.345s	0.128s	0.169s	0.044s	0.086s
2048x2048	1.674s	0.512s	0.701s	0.156s	0.346s
4096x4096	7.008s	2.045s	4.118s	0.599s	2.649s

3D FFT size	x86	DaCS PCI		DaCS PCI + PXCBS	
		Comm.	Wall.	Comm.	Wall.
64x64x46	0.021s	0.033s	0.036s	0.016s	0.018s
128x128x28	0.583s	0.261s	0.279s	0.088s	0.105s
256x256x256	6.062s	2.083s	2.246s	0.643s	0.812s

Use Case 2: Gravitational Waves

- Astrophysical application used for performing an all-sky coherent search for periodic signals of gravitational waves in a narrowband data of a detector



- Single PowerXCell8i speedup: 3.24x
- Hybrid DaCS speedup: 3.56x
- Hybrid DaCS and PXCBS speedup: 4.5x

Management of DaCS hybrid jobs

- Integration of the DaCS in the production environment
- Dynamic hybrid node allocation
- Possible core per core ratios (1:8,1:16)
- Hybrid partitions defined within Torque queueing system scripts

```
#!/bin/sh
#PBS -N test_hybrid
#PBS -l nodes=2:ppn=4:opteron+8:ppn=4:cell
#PBS -l walltime=1:00:00

module load openmpi-x86_64
module load dacs
mpiexec ./program_dacs_hybrid
```

Thank you for your attention