Adapting to changes in HPC

Group A

RIKEN AICS HPC Youth Workshop 2016

Adapting to Changes in HPC

- Points of concern (+ve or -ve):
  - Diversity in Architectures (Manycore, reconfigurable, non-Von Neumann, … , etc)
  - Hardware specialization (as HW gets cheaper, will specialized HW be a feasible path?)
  - Diversity in parallel programming models and languages
  - Need to account for optimizing for power
  - Need to account for fault tolerance

Edgar Josafat MARTINEZ NORIEGA
@U. Electro-comm. Tokyo

GPGPU virtualization frameworks for real time simulations and visualization

Talk: High performance MD simulation and visualization for low powered mobile devices using streaming compressed graphics

MD Simulation and Visualization on Low-powered Devices

- Challenge utilizing diverse architectures:
  - Porting the MD code optimized for GPU to other accelerators.
  - Possible bottleneck sharing memory resources between different accelerators for rendering purposes.
- Hardware specialization:
  - Usage of MD-GRAPE like boards for MD computation again?

MD Simulation and Visualization on Low-powered Devices

- Data access:
  - Memory bandwidth between client/server.
- Fault tolerant mechanism:
  - For MD simulation is totally necessary.
  - This is one of the major concerns in our project.

Opportunity for New Applications and Visualization

- As the exa-scale moves on, new applications for HPC will arise.
- New ideas for interacting with data and representation will be proposed.
- VR, immersive projections and hand-held devices are good candidates to interact with these new HPC apps.

Jan Winkelmann
@Jülich Supercomputing Centre

High Performance Interior Eigensolvers

Talk: Towards Automated Load Balancing via Spectrum Slicing for FEAST-like solvers

My Work

- I work on solvers for diagonalization
- Kernel-based programming:
  - Largely calls to MPI, BLAS+LAPACK
- ZGEMM and Linear System Solves
- Extending the work should be easy …

Challenges: New architectures

- Example: Polynomial filter, loop over ZGEMM
- Jureca GPUs (2xK80):
  - 4 visible GPUs is already surprisingly difficult
  - Vendors fail to supply reasonable kernels
- MKL (Intel Math Kernel Library):
  - Tall and skinny matrices benefit from OpenMP
- Abstractions will increasingly leak

Other Challenges

- Algorithm-Based Fault Tolerance
  - Already under investigation for Eigensolvers [Sakurai]
  - Kernels: Leaky abstractions
- "Just re-implement it”
  - Iterative solvers are numerically delicate
  - Different paradigm == different algorithm
  - Validation is problematic

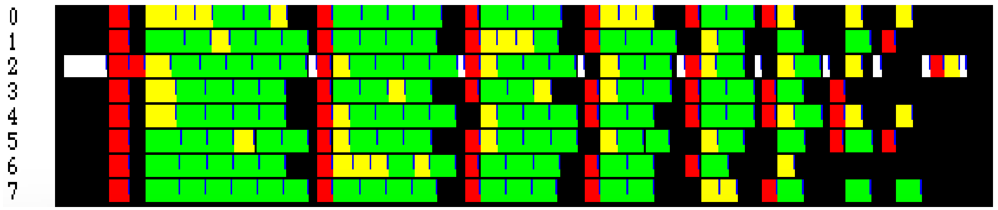
Keisuke Tsugane
@U. Tsukuba

Parallel Programming Models

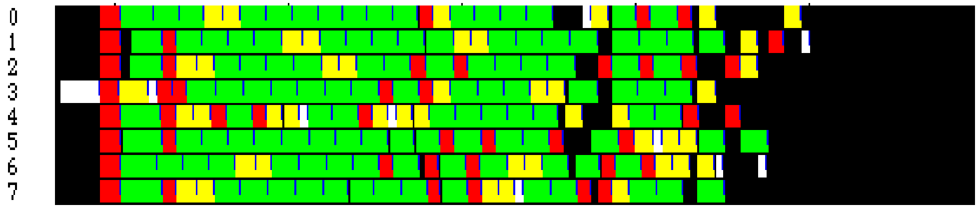
Talk: Proposal for Dynamic Task Parallelism in PGAS Language XcalableMP

Extension of XcalableMP for Multitasking

- Task parallelism has become popular for shared memory programming
  - OpenMP supports task directive
  - User can describe data dependencies for task-to-task synchronization
- Proposal for the tasklet directive for dynamic task parallelism in PGAS language XcalableMP
  - To reduce synchronization costs and to write easily dynamic task parallelism on distributed memory systems



Barrier synchronization



Task–to–task synchronization

Results of Block Cholesky Factorization on 8 threads execution

White : potrf
Red : trsm
green : gemm
yellow : syrk
Black : thread is waiting for synchronization

BLAS or LAPACK functions

Adapting to changes in HPC to my work

- Diversity in Architectures : Manycore
  - Barrier synchronization costs become more and more large for many core
    - Each core needs to execute the calculation and communication
  - Task-to-task synchronization by XMP tasklet directive can reduce the execution time on distributed memory systems
    - It reduces waiting threads for synchronization and overlaps the computation and communication executed by each core

- Diversity in parallel programming models and languages
  - Directive-based programming models for accelerator are proposed
    - OpenACC, OpenMP 4.0 : target directive
  - Combination of XMP tasklet directive and other directive-based model
    - Task-to-task synchronization by XMP tasklet directive can schedule these executions on each accelerator and accelerator cluster
    - It may reduce a synchronization cost between tasks on accelerator

Ivo Kabadshow
@Jülich Supercomputing Centre

Fast Multipole Methods for MD

Talk: In C++ we trust -- performance portability out of the box?

Molecular Dynamics Simulations for Exascale

- Status Quo
  - Already communication-bound
  - FLOPs are not a bottleneck
  - Cannot be scaled arbitrarily

MD on New Architectures and New Languages

- Specialized HW might help to overcome current limitations
- Optimize HW towards single application

MD vs Optimization for Power

- Time to solution should be the target
- Not yet seen architecture deviating from that rule

MD vs Fault Tolerance

- Should not be handles by the MD application
- 1k time steps per second -> just redo computation
- How to realize faults?

Hideaki Oba
@U. Kobe

Computational Fundamental Physics

Talk: Tensor Renormalization Group and truncation number of SVD

The limitation of Tensor Network

- HOTRG can apply to high dimensional models
- However, numerical cost of HOTRG is big

  e.g.) 4-dimensional HOTRG

Memory cost	Compute cost
$O(D^8)$	$O(D^{15})$

D: The dimension of tensors

If renormalized tensors can be stored in one node, K computer:

16GB /node → $D < 15$

How to manage the problem?

HPC changes...

- We can treat the tensors in multiple nodes

  → Achieve more accurate result

Other

- Reduce *D* or its power

Seigo Imamura
@U. Kobe

High Performance Computing

Talk: Performance Evaluation of the Parareal Method for Large-scale Diffusion Problem

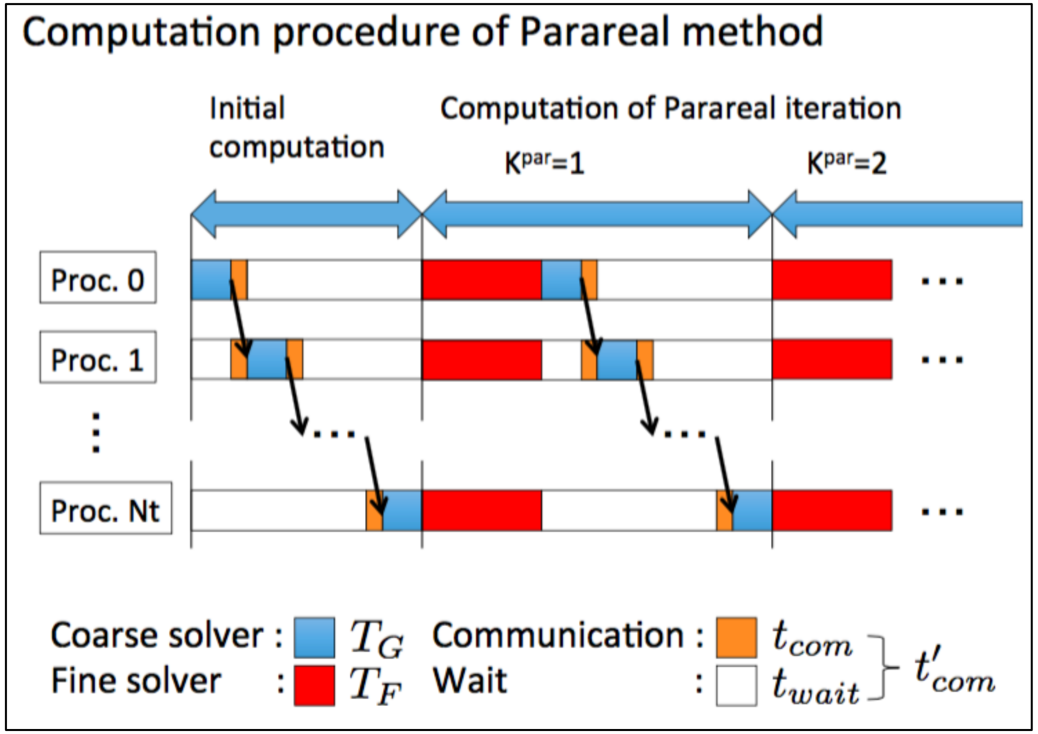
Limitation of Parareal method

Parareal method is parallel time integration method.

There are serial and parallel computation part in the method.

At the large parallelization

- Low parallel efficiency
  - Parallel part is small.
  - Serial part is large.
- Large wait time



Adapting to changes in HPC

- Diversity in Architectures (Many-core, reconfigurable, … , etc.)
- Hardware specialization
  - Adaptable to various architectures
  - Easy communication and one-way
- Developed Parallel numerical library
  - Low cost solver for serial computation
- Need to account for optimizing for power
  - Easy cutting energy cost for simple algorithm