Awareness of MPI Virtual Process Topologies on the Single-Chip Cloud Computer

Steffen Christgau, Bettina Schnor

Potsdam University
Institute of Computer Science
Operating Systems and Distributed Systems

HIPS@IPDPS, 21. May 2012
Outline

- The Single-Chip Cloud Computer (SCC)
  - RCKMPI – MPI on the SCC
  - Topology-Awareness for RCKMPI – The concept
  - Topology-Awareness for RCKMPI – Evaluation
  - Summary and Future Work
Outline

■ The Single-Chip Cloud Computer (SCC)
■ RCKMPI – MPI on the SCC
  ■ Topology-Awareness for RCKMPI – The concept
  ■ Topology-Awareness for RCKMPI – Evaluation
  ■ Summary and Future Work
Outline

- The Single-Chip Cloud Computer (SCC)
- RCKMPI – MPI on the SCC
- Topology-Awareness for RCKMPI – The concept
  - Topology-Awareness for RCKMPI – Evaluation
  - Summary and Future Work
The Single-Chip Cloud Computer (SCC)
RCKMPI – MPI on the SCC
Topology-Awareness for RCKMPI – The concept
Topology-Awareness for RCKMPI – Evaluation
Summary and Future Work
Outline

- The Single-Chip Cloud Computer (SCC)
- RCKMPI – MPI on the SCC
- Topology-Awareness for RCKMPI – The concept
- Topology-Awareness for RCKMPI – Evaluation
- Summary and Future Work
1. The Single-Chip Cloud Computer (SCC)

Many-Core Architecture Research Community (MARC)

- established by Intel
- research of universities together with Intel
- world-wide community (dominated by European institutions)
- Website, regular symposia (every 6 months, 5 up to now)

Our group is MARC member

- Focus: application scalability
- Experiences with parallel ASP, climate simulation
24 tiles, 48 P54C cores, connected via Network-on-Chip, no Cache-Coherence

fast 16 KB tile SRAM on each tile Message Passing Buffer (MPB)
2. RCKMPI: MPI on the SCC

- fork of MPICH2

![Diagram showing MPI components and implementations]
2. RCKMPI: MPI on the SCC

- fork of MPICH2

Diagram:

- Application
- Message Passing Interface
  - MPICH2
  - ROMIO (MPI IO Implementation)
  - Abstract Device Interface (ADI3)
  - ADIO
  - Channel-3 Device
    - BG
    - Cray

- Process Management Interface
  - Sock
  - Nemesis
  - SCCMPB
  - SCCSHM
  - SCCMulti

Bettina Schnor (Potsdam University)
SCCMPB uses the fast Message Passing Buffer of each tile as shared memory and divides it into $n$ equal-size **Exclusive Write Sections (EWS)** ⇒ remote write, local read
SCCMPB uses the fast Message Passing Buffer of each tile as shared memory and divides it into $n$ equal-size **Exclusive Write Sections (EWS)** $\Rightarrow$ remote write, local read
Comparison of different CH3-devices at maximum Manhattan distance 8
Bandwidths for Manhattan distance 0, 5 and 8 (two processes started).
Bandwidths for maximum Manhattan distance 8, and varied number of MPI processes
Remember:

- SCCMPB uses the fast Message Passing Buffer of each tile as shared memory (total 384 KB)

The MPB is equally divided in \( n \) sections \( \rightarrow \) depending on the number of started MPI processes.
3. Topology Awareness for RCKMPI: The Concept

The bandwidth between 2 RCKMPI processes depends on:

- the number of started MPI processes since a fully-connected network between all MPI processes is managed.
Goal: The bandwidth between communicating processes, so-called neighbors in the Task Interaction Graph should be increased.
Goal: The bandwidth between communicating processes, so-called neighbors in the Task Interaction Graph should be increased.
Requirements:

1. An improved MPB layout must consider both: communication neighbors and group communication (barriers, broadcasts, gather/scatter, ...)
2. Each MPI process has to know its new offset within all remote MPBs.
Putting things together ...

Internal barrier for recalculation phase of new MPB addresses.
MPI offers API to specify virtual process topology

```c
#define NUM_DIMS 2

int grid_dims[NUM_DIMS];
int grid_periods[NUM_DIMS];
MPI_Comm comm_topo;

/* for a grid, set all items of grid_periods to 0 */
for (int i = 0; i < NUM_DIMS; i++)
  grid_periods[i] = 0;

MPI_Dims_create(numProcs, NUM_DIMS, grid_dims);
MPI_Cart_create(MPI_COMM_WORLD, NUM_DIMS, grid_dims,
                grid_periods, true, &comm_topo);
```
MPI offers API to specify *virtual process topology*

```
#define NUM_DIMS 2

int grid_dims[NUM_DIMS];
int grid_periods[NUM_DIMS];
MPI_Comm comm_topo;

/* for a grid, set all items of grid_periods to 0 */
for (int i = 0; i < NUM_DIMS; i++)
  grid_periods[i] = 0;

MPI_Dims_create(numProcs, NUM_DIMS, grid_dims);
MPI_Cart_create(MPI_COMM_WORLD, NUM_DIMS, grid_dims,
                grid_periods, true, &comm_topo);
```
4. Topology Awareness for RCKMPI: Evaluation

- enhanced RCKMPI with 1D topology (48 procs, 2 Cache lines)
- enhanced RCKMPI with 1D topology (48 procs, 3 Cache lines)
- enhanced RCKMPI without topology (48 procs)
Results for 2D CFD application with **ring topology**:
Summary and Future work

- SCC is equipped with a fast NoC.
- Message Passing Buffer on tile is beneficial for fast communication.
- Bandwidth performance gain for MPI applications by using MPI’s virtual process topologies and rearranging the MPB layout.

Current/Future Work:

- Comparison with I. C. Ureña, and M. Gerndt: Improved RCKMPI's SCCMPB Channel: Scaling and Dynamic Processes Support, ARCS 2012
- Fixed the One-Sided Communication in RCKMPI support of applications based on Global Arrays
Summary and Future work

- SCC is equipped with a fast NoC.
- Message Passing Buffer on tile is beneficial for fast communication.
- Bandwidth performance gain for MPI applications by using MPI’s *virtual process topologies* and rearranging the MPB layout.

Current/Future Work:

- Comparison with  
- Fixed the One-Sided Communication in RCKMPI support of applications based on Global Arrays
Summary and Future work

- SCC is equipped with a fast NoC.
- Message Passing Buffer on tile is beneficial for fast communication.
- Bandwidth performance gain for MPI applications by using MPI’s *virtual process topologies* and rearranging the MPB layout.

Current/Future Work:

- Fixed the One-Sided Communication in RCKMPI $\implies$ support of applications based on Global Arrays