Shared Virtual Memory for the SCC:

bare metal programming for future many-core architectures

Pablo Reble

March 23, 2012

CHAIR FOR OPERATING SYSTEMS

Univ.-Prof. Dr. habil. Thomas Bemmerl

RWTH AACHEN UNIVERSITY

Agenda

- Parallel Programming Concepts
- SCC
- MetalSVM
- SVM subsystem
- Application
- Demo

- Message Passing (MPI)
  - process parallelism
  - explicit communication
- Shared Memory (OpenMP)
  - loop/thread parallelism
  - implicit communication
  - coherent memory required

MARC

The Single-chip Cloud Computer experimental processor is a concept vehicle created by Intel Labs as a platform for many-core software research.

- stands for: Many-Core Application Research Community
- launched in 2010 by Intel
- intention: provide access to future processor architectures to a broader audience
- sponsored Symposium, twice a Year in Europe
- `http://communities.intel.com/community/marc`

P54C

Pentium I family

- presented in 1994
- 32 bit intel architecture
- 75-100 MHz
- 3.3 volt
- on-chip APIC
- multiprocessor capability
- instruction to invalidate cache-located tagged data: `CLIINV`

P54C

Pentium I family

- presented in 1994
- 32 bit intel architecture
- ~~75-100 MHz~~ 100 MHz-1 GHz
- 3.3 volt
- on-chip APIC
- multiprocessor capability
- instruction to invalidate cache-located tagged data: CL1INV

P54C

Pentium I family

- presented in 1994
- 32 bit intel architecture
- ~~75-100 MHz~~ 100 MHz-1 GHz
- ~~3.3 volt~~ 0.66 V - 1.16 V
- on-chip APIC
- multiprocessor capability
- instruction to invalidate cache-located tagged data: CL1INV

P54C

Pentium I family

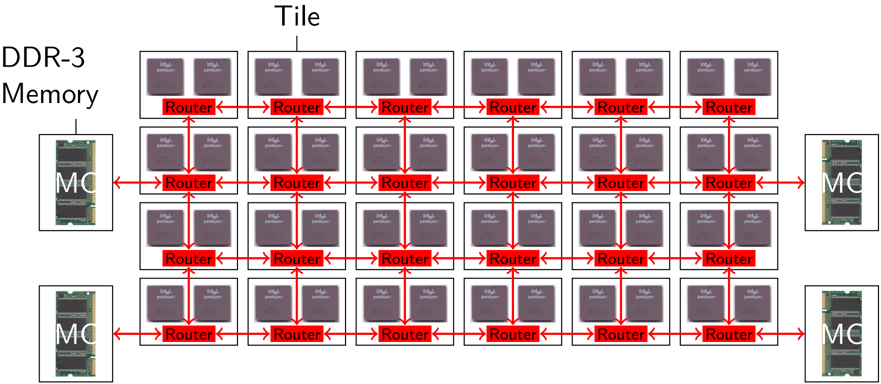
- presented in 1994
- 32 bit intel architecture
- ~~75-100 MHz~~ 100 MHz-1 GHz
- ~~3.3 volt~~ 0.66 V - 1.16 V
- on-chip APIC
- multiprocessor capability
- instruction to invalidate cache-located tagged data: `CLIINV`

SCC Environment

SCC

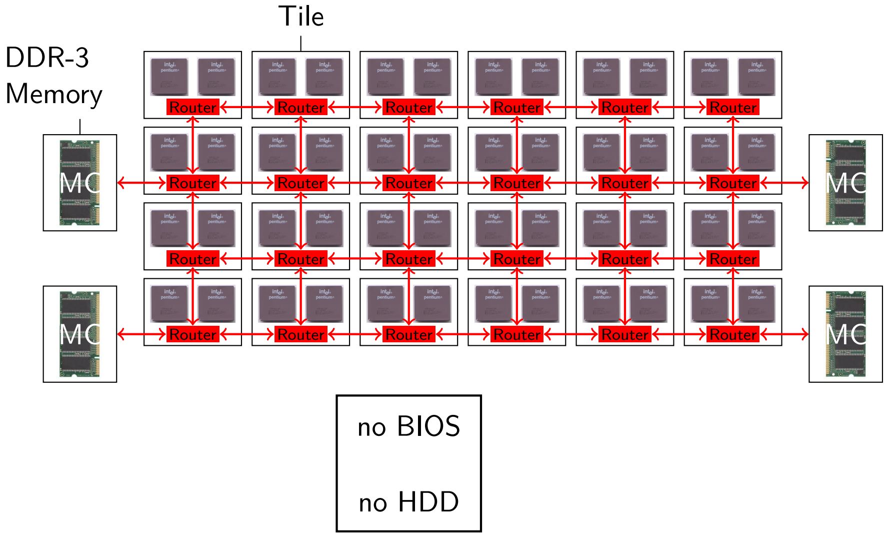
SCC Environment

SCC Environment

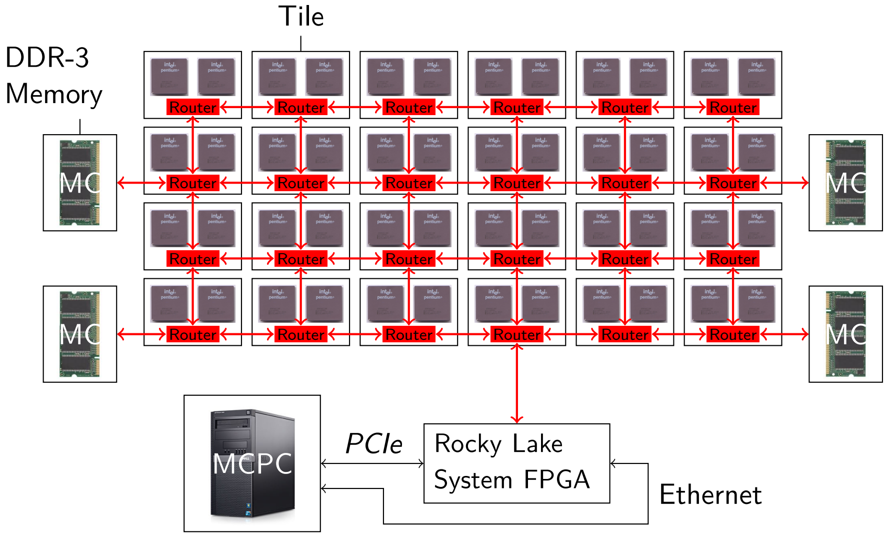


state-of-the-art 2D mesh interconnect

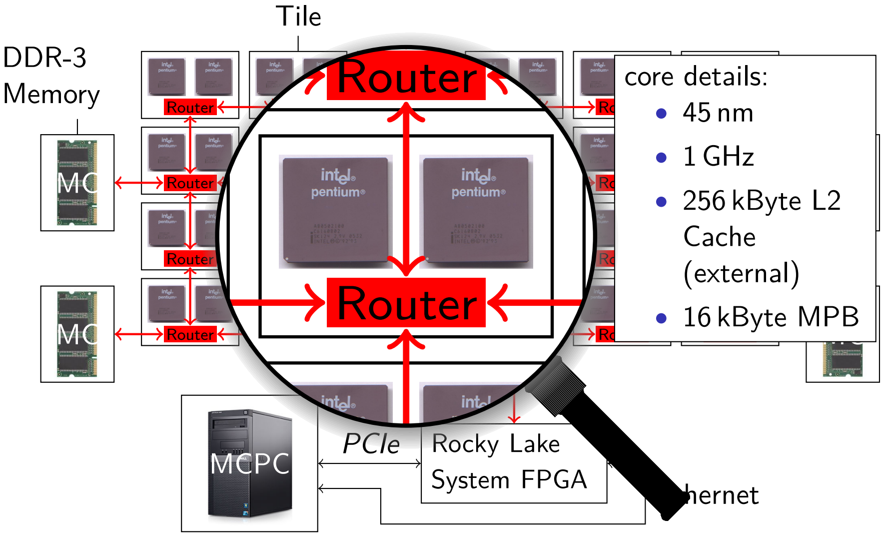
SCC Environment



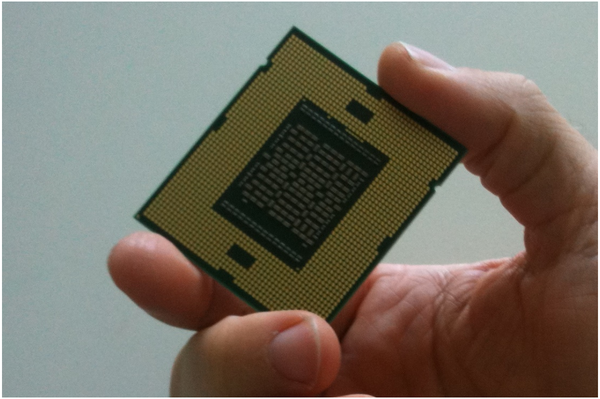
SCC Environment



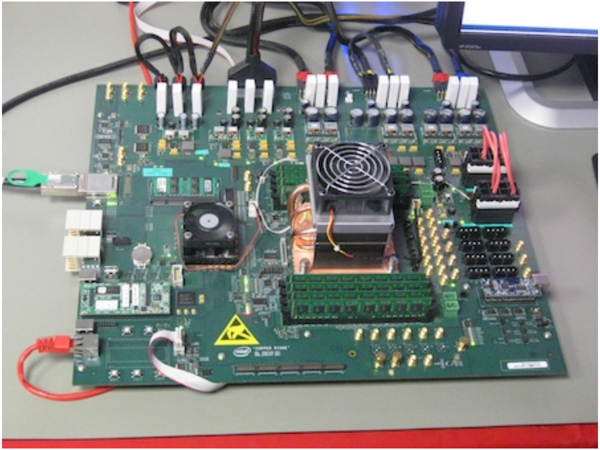
SCC Environment



Rocky Lake Processor



Rocky Lake Platform



Default Configuration

- SCC provides shared but not coherent memory
- Cluster-like programming environment
- Separate Linux booted on each core
- Shell-script to start processes
- RCCE ['rɒki] – light-weight message passing library

Default Configuration

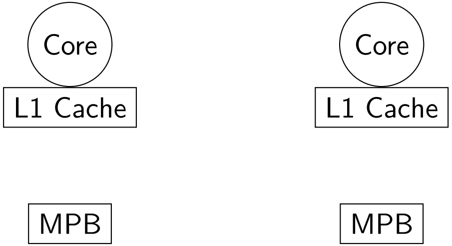
- SCC provides shared but not coherent memory
- Cluster-like programming environment
- Separate Linux booted on each core
- Shell-script to start processes
- RCCE [ˈrɒki] – light-weight message passing library

Default Configuration

- SCC provides shared but not coherent memory
- Cluster-like programming environment
- Separate Linux booted on each core
- Shell-script to start processes
- RCCE [ˈrɒki] – light-weight message passing library

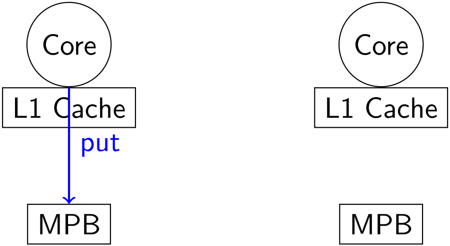
RCCE

- light-weight communication environment
- local put, remote get approach
- uses MPB to realize blocking, synchronous message passing



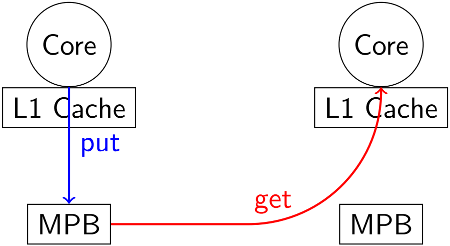
RCCE

- light-weight communication environment
- local put, remote get approach
- uses MPB to realize blocking, synchronous message passing



RCCE

- light-weight communication environment
- local put, remote get approach
- uses MPB to realize blocking, synchronous message passing

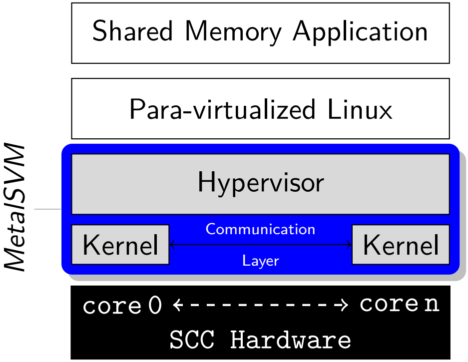


SVM

- VM?
  - virtual address space of a process is mapped onto a physical address space
  - almost all UNIX system implementations, including Linux, use demand paging to manage the allocation of physical memory
- SVM?
  - concept of a single address space shared by a number of processors
  - strategies to generate coherent but distributed memory

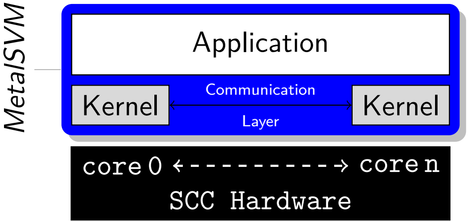
MetalSVM

- work in progress: research grant by Intel Labs Braunschweig
- shared virtual memory for many-core systems
- bare-metal hypervisor based approach



Project Goal

- work in progress: research grant by Intel Labs Braunschweig
- shared virtual memory for many-core systems
- bare-metal hypervisor based approach



First SVM Prototype

History

- tiny OS kernel for education: eduOS (since 2009)
- 1^{st} MARC Symposium (Braunschweig 2010)
  - presented basic ideas to integrate an SVM system into a bare-metal Hypervisor
- 3^{rd} MARC Symposium (Ettlingen 2011)
  - Comm. and Synch. Layer (Focus on HW Synch. Support and High Concurrency)
- 4^{th} MARC Symposium (Potsdam 2012)
  - SVM Prototype and first application benchmark

lguest

Lguest is a small x86 32-bit Linux hypervisor [...] serves as an excellent springboard for mastering the theory and practice of x86 virtualization [...] You should also be inspired to create your own hypervisor, using your own pets as logo. – Rusty Russell '07

lguest

Lguest is a small x86 32-bit Linux hypervisor [. . .] serves as an excellent springboard for mastering the theory and practice of x86 virtualization [. . .] You should also be inspired to create your own hypervisor, using your own pets as logo. – Rusty Russell '07

Prototype

- handles SVM related data:
  - use write trough strategy
  - enable Level 1 caching only
  - tag related Cache-Lines as `MPBT`
- Consequences:
  - \+ use write combining buffer
  - \+ hardware support for invalidation
  - – no use of Level 2 Cache

Memory Consistency Models

- Strong Memory Consistency Model:
  - Exactly one owner per page with read/write permissions
  - allow dynamical change of ownership
  - use messages to handle change

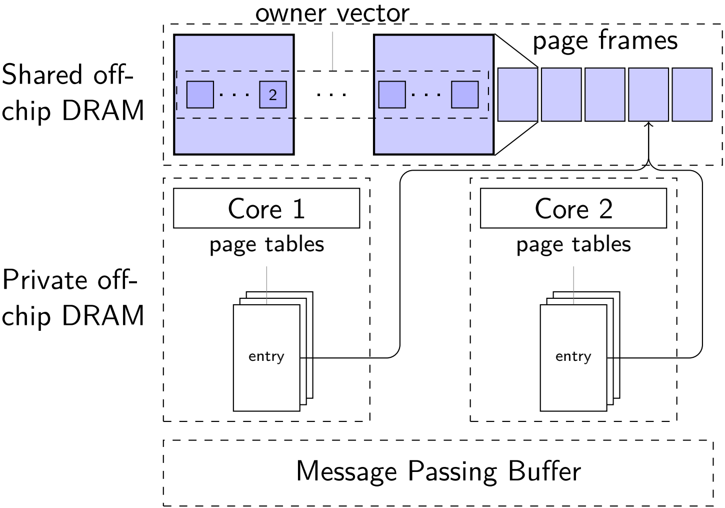
Memory Consistency Models

- Strong Memory Consistency Model:
  - Exactly one owner per page with read/write permissions
  - allow dynamical change of ownership
  - use messages to handle change
- Lazy Release Consistency:
  - Application explicitly controls Consistency (`svm_barrier`)

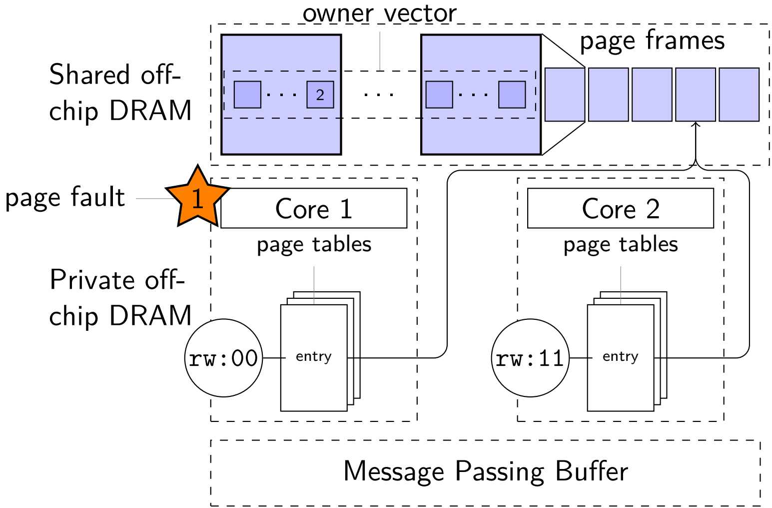
SVM subsystem

- First steps to apply Shared Memory Programming on the SCC use a small subset of SMI:
  - `svm_alloc`
  - `svm_flush`
  - `svm_invalidate`

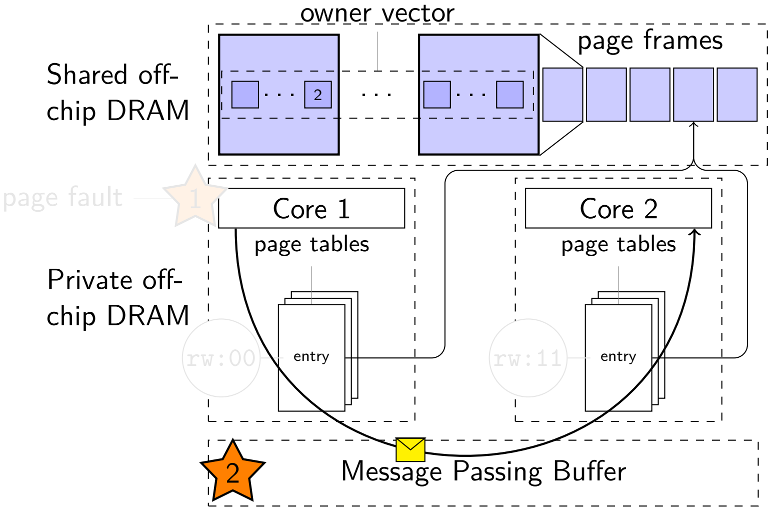
Visualize Change of Ownership



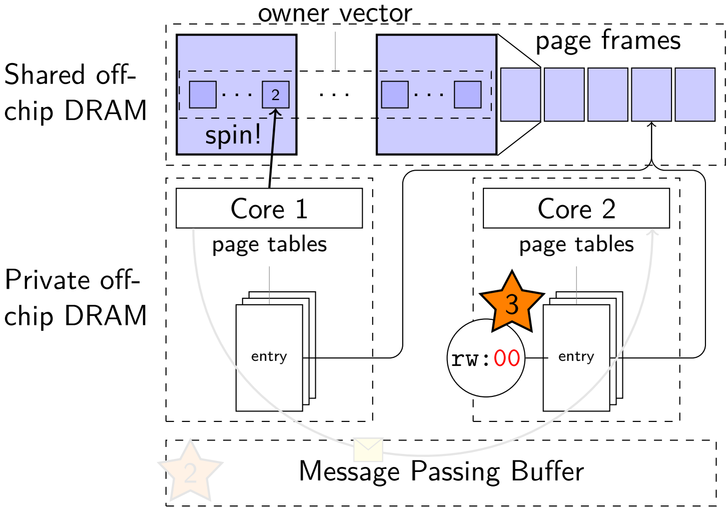
Visualize Change of Ownership



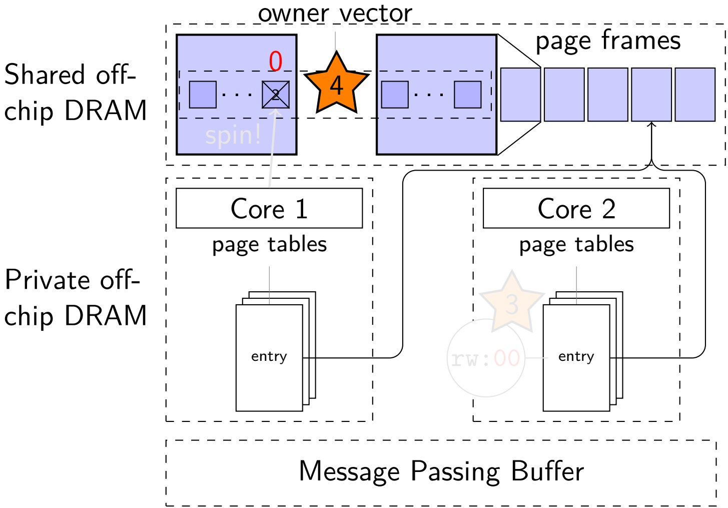
Visualize Change of Ownership



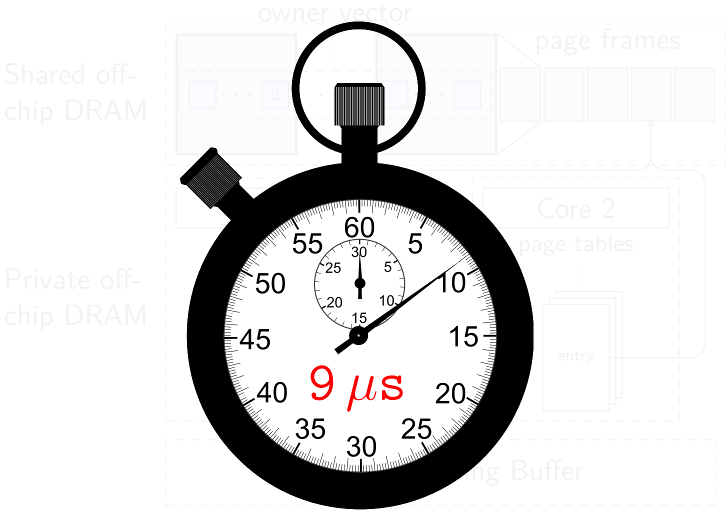
Visualize Change of Ownership



Visualize Change of Ownership



Visualize Change of Ownership

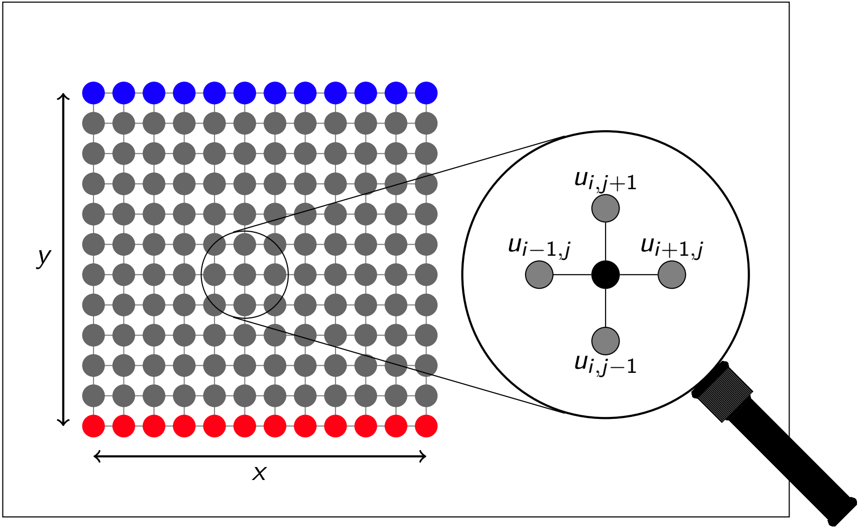


Application

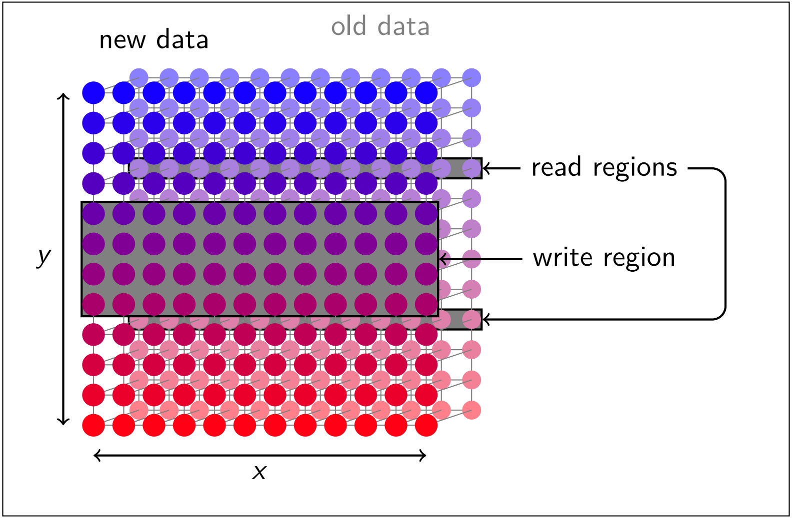
- Stencil app (part of RCCE)
- Dirichlet Bounding Condition
- Solver: Jacobi Over Relaxation algorithm
- Synchronous behavior

  Shared Memory: Barrier between iterations
  Message Passing: implicitly
- SCC Platform running with 533 MHz core and 800 MHz memory/mesh
- double precision

access pattern

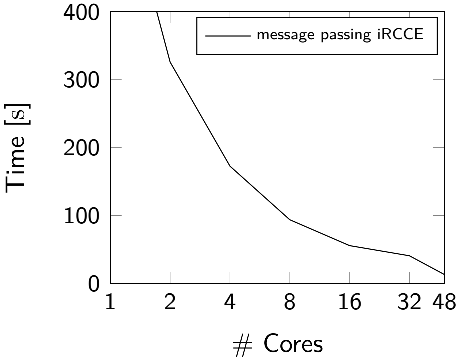


access pattern



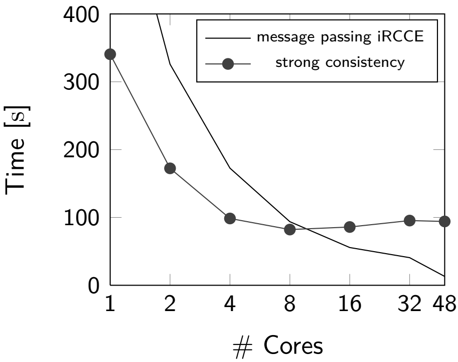
First Results

- Five Stamp Stencil
  - Problem size 1024×512



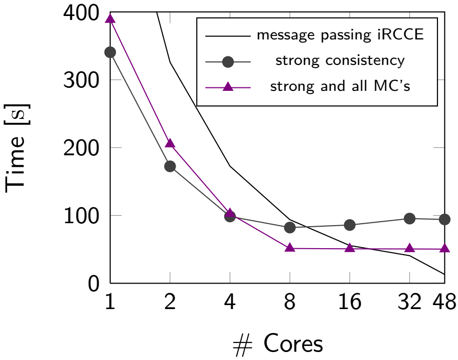
First Results

- Five Stamp Stencil
  - Problem size 1024×512



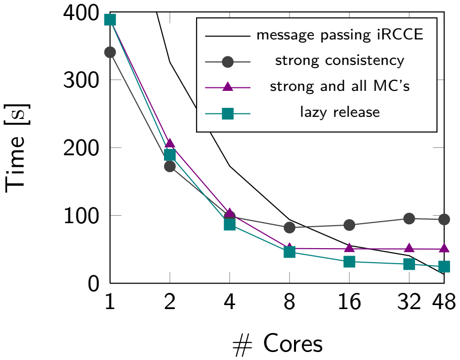
First Results

- Five Stamp Stencil
  - Problem size 1024×512



First Results

- Five Stamp Stencil
  - Problem size 1024×512



Conclusion and Outlook

Conclusion

- First Prototype of MetalSVM is running
- Results are promising

Outlook

- Boot Linux on multiple hypervisor instances
- Connect two SCCs
- plan a release in 2012 `metalsvm.org`

Conclusion and Outlook

Conclusion

- First Prototype of MetalSVM is running
- Results are promising

Outlook

- Boot Linux on multiple hypervisor instances
- Connect two SCCs
- plan a release in 2012 `metalsvm.org`

Demo