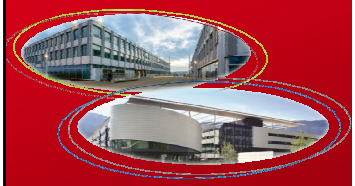


www.cea.fr

leti & list

Virtual prototyping acceleration on manycore architectures

N. VENTROUX
CEA LIST
Computing and Design Environment Lab
CEA-Saclay NanoInnov
nicolas.ventroux@cea.fr

Introduction

- SoCs are becoming more and more complex
  - Complexity in a chip is increasing x1.6 every 2 years (ITRS 2013)
    - e.g. Apple A8 SoC owns 2B transistors
  - Development costs/TTM must be contained
- Virtual prototyping can
  - Improve design quality and performances with fast design exploration
    - Performance, power, temperature, reliability
  - Provide a virtual HW platform to SW developpers
    - Parallelize design phases and increase design productivity
- Limitation of VP
  - Simulation performance does not scale with complexity *(at constant accuracy)*

➡ Inevitable need for VP acceleration

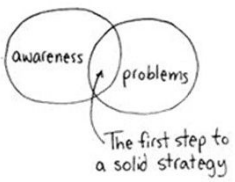
SystemC is mainly used as VP kernel

- SystemC
  - Accellera standard C++ library (IEEE Std. 1666™-2005)
  - A discrete event simulator (DES)
  - Current version is 2.3 + TLM 2.0.1

- SystemC is widely used for
  - Design-space exploration
  - Verification/validations

What about parallelizing SystemC?

- The evaluation of processes is potentially highly parallel
- But it is a very hard problem *(no industrial solution exists)*
  - A parallel implementation must keep the exact same behavior than the sequential one
    - It should not introduce new race conditions
    - Sequential phases modelling concurrency must be respected
  - Process lengths are usually shorts
    - Performances are very impacted by the kernel and synchronizations
  - Mainly irregular and control-oriented processing
    - Lots of function calls, branches and conditional executions
    - Only few simulations have data-parallelism



Related works

- Static scheduling (conservative)
  - Use of a global synchronization (at each δ-cycle)
  - Reduces synchronization and context switching overheads
  - Support shared variables between processes
    - Ensure that two processes accessing the same variable will never be run in parallel
  - Approaches
    - Local dependency analysis [BUCH07, MOU04, NAG07]
      - Suppose that a process only access to variables declared within its own module
    - Global static analysis [CHEN12, BLAN10]
      - Detect access to shared variables between Wait statements
      - Almost not practical with high level modeling (C++ or TLM)
    - TLM is not supported (e.g. RAM are shared array between components)
- HW acceleration (conservative)
  - Not fully compliant with SystemC (or limited use)
  - With the CELL processor [KAOU09]
  - With GP-GPU
    - Accelerate some processes on GPU [SINH12, VINC12, NANJ10]
    - Distribute multiple simulations [KUNZ12]
  - With a dedicated many-core [VENT14]

Related works

- Distributed SystemC (conservative and optimistic)
  - Use implicit communications (FIFO), synchronization or lock mechanisms or avoid shared variables
  - Works if few communications between clusters
    - Need to clusterize the simulated architecture
    - Modification of the simulated architecture to explicit the parallelism
  - Approaches:
    - Multiple SystemC schedulers
      - Conservative approaches [ZIYU09, CHOP06, CHUN14]
      - With speculation or relaxing methods [MEL10, COMB08, PESS10]
        - *Synchronizations on every communication, null messages...*
    - Distributed simulation on top of SystemC [HUAN08]
- Parallel SystemC kernel (conservative)
  - Use of a global synchronization (at each δ-cycle)
  - Execute the evaluation phase in parallel
  - Load-balancing techniques (work-stealing, work-sharing...)
  - Approaches
    - Not SystemC compliant [EZUD09, SCHU10, KHAL10]
      - Shared variable access atomicity is left to the user
      - Process order is not guaranteed
    - SystemC compliant
      - Run processes in parallel though they do not run at the same time [MOY14, CHEN12]
      - Partition simulation and preserve co-routine semantics within a partition [SCHU13]

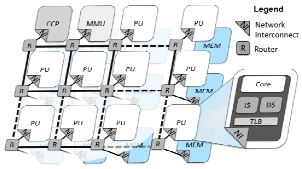
SCale

- Development of a new SystemC kernel, named SCale
  - Compliant with Accellera standards
    - TLM, TLM2.0, SystemC 2.3.1
  - Lightweight and think-parallel kernel
  - Conservative, in-order, global synchronization
  - Supports multiple architecture
    - Tilera64, AMD Bulldozer, i7 (for now)

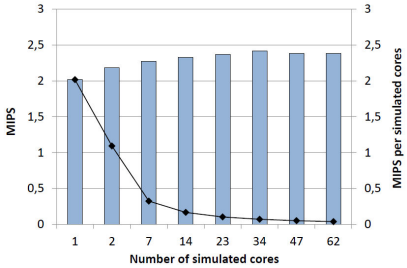
SCALE
Parallel SystemC Kernel

Results

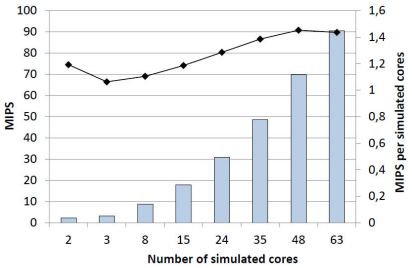
- Experiments with SESAM [VENT13] and SimSoC [HELM08] modeling a 64-core manycore architecture on a 64-core AMD Opteron machine
- Performances
  - 675 MIPS and 21x (SimSoC and BT)
  - 90MIPS and 36x (SESAM)



With Accelera (SESAM)



With SCale (SESAM)



Conclusion

- Designing large and complex systems need new methods and tools to accelerate simulations
- CEA LIST proposes a new SystemC kernel named *SCale* able to leverage the multiple cores of a many-core architecture
  - Guarantee accuracy and maintain sequential behavior and reproductivity
  - Up to 36x on a 48-core host x86

Bibliography

SystemC acceleration (static scheduling, parallel and distributed)

[BUCH07] R. Buchmann and A. Greiner, *A Fully Static Scheduling Approach for Fast Cycle Accurate SystemC Simulation of MPSoCs*, ICM 2007.

[MOU04] G. Mouchard, D. Gracia Pérez and O. Temam, *FastSysC: A Fast Simulation Engine*, DATE 2004.

[NAG07] Y.N. Naguib et R.S. Guindi, *Speeding up SystemC simulation through process splitting*, DATE 2007.

[CHEN12] W. Chen, X. Han and R. Dömer, *Out-of-order parallel simulation for ESL design*, DATE 2012.

[BLAN10] N. Blanc and D. Kroening, *Race analysis for SystemC using model checking*, TODAES, vol. 15, no. 3, 2010.

[CHOP06] B. Chopard, P. Combes, et J. Zory, *A Conservative Approach to SystemC Parallelization*, ICCS 2006.

[MEL10] A. Mello, I. Maia, A. Greiner, et F. Pecheux, *Parallel Simulation of SystemC TLM 2.0 Compliant MPSoC on SMP Workstations*, DATE 2010.

[ZIYU09] H. Ziyu et al., *A Parallel SystemC Environment: ArchSC*, ICPADS 2009.

[COMB08] P. Combes, E. Caron, and B. Chopard, *Relaxing Synchronization in a Parallel SystemC Kernel*, ISPA 2008.

[PESS10] I. Pessoa, A. Mello, A. Greiner and F. Pecheux, *Parallel TLM Simulation of MPSoC on SMP Workstation: Influence of Communication Locality*, ICM 2010.

[EZUD09] P. Ezudheen et al., *Parallelizing SystemC kernel for fast hardware simulation on SMP machines*, PADS 2009.

[SCHU10] C. Schumacher, R. Leupers, D. Petras and A. Hoffmann, *parSC: Synchronous Parallel SystemC Simulation on Multi-Core Host Architectures*, CODES+ISSS 2010.

[KHAL10] R.S. Khaligh and M. Radetzki, *A dynamic load balancing method for parallel simulation of accuracy adaptive TLMs*, FDL 2010.

[HUAN08] K. Huang, I. Bacivarov, F. Hugelshofer and L. Thiele, *Scalably distributed SystemC simulation for embedded applications*, ISIES 2008.

[MOY13] M. Moy, Parallel Programming with SystemC for Loosely Timed Models: A Non-Intrusive Approach, DATE 2013.

[CHUN14] M.-K. Chung, J.-K. Kim, and S. Ryu, *SimParallel: A High Performance Parallel SystemC Simulator Using Hierarchical Multi-threading*, ISCAS 2014.

[SCHU13] C. Schumacher et al., legaSCi: Legacy SystemC Model Integration into Parallel SystemC Simulators, IPDPSW 2013.

[WEIN14] J.H. Weinstock et al., Time-Decoupled Parallel SystemC Simulation, DATE 2014.

[PEET11] J. Peeters, N. Ventroux, T. Sassolas, *A SystemC TLM Framework for Distributed Simulation of Complex Systems with Unpredictable Communication*, DASIP 2011.

HW SystemC acceleration

[SINH12] R. Sinha, A. Prakash and H.D. Patel, *Parallel Simulation of Mixed-abstraction SystemC Models on GPUs and Multicore CPUs*, ASP-DAC 2012.

[KUNZ12] G. Kunz, D. Schemmel, J. Gross and K. Wehrle, *Multi-level Parallelism for Time- and Cost-efficient Parallel Discrete Event Simulation on GPUs*, PADS 2012.

[VINC12] S. Vinco, D. Chatterjee, V. Bertacco and F. Fummi, *SAGA: SystemC Acceleration on GPU Architectures*, DAC 2012.

[NAN10] M. Nanjundappa, H.D. Patel, B.A. Jose, et S.K. Shukla, *SCGPSim: A fast SystemC simulator on GPUs*, ASP-DAC 2010.

[VENT14] N. Ventroux, J. Peeters, T.Sassolas and James C. Hoe, Highly-Parallel Special-Purpose Multicore Architecture for SystemC/TLM Simulations, SAMOS 2014.

Virtual prototyping solutions

[VENT13] N. Ventroux, T. Sassolas, A. Guerre and C. Andriamisaina, *SESAM: A Virtual Prototyping Solution to Design Multicore Architectures*, Chapter in Multicore Technology: Architecture, Reconfiguration, and Modeling, edited by M.Y. Qadri and S.J. Sangwine, CRC PRESS, pp. 61-104, July 2013.

[HELM08] C. Helmstetter and V. Joloboff, *SimSoC: A SystemC TLM integrated ISS for full system simulation*, APCCAS, December 2008.

Thank you for your attention

INSTITUT CARNOT CEA LETI

INSTITUT CARNOT CEA LIST

MINATEC

digiteo

leti
Centre de Grenoble
17 rue des Martyrs
38054 Grenoble Cedex

List
Centre de Saclay
Nano-Innov PC 172
91191 Gif sur Yvette Cedex