Multi-scale, Multi-objective, Behavioral Modeling & Emulation of Extreme-scale Systems

* NSF Center for High-Performance Reconfigurable Computing (CHREC)

\+ Center for Compressible Multiphase Turbulence (CCMT)

University of Florida

CHREC
NSF Center for High-Performance Reconfigurable Computing

ModSim Workshop 2014

Nalini Kumar*+
Carlo Pascoe*+
Dylan Rudolph*+
Graduate Research Assistants
Dept. of ECE

Dr. Herman Lam*+
Assoc. Professor of ECE
Assoc. Director of CHREC

Dr. Alan George*+
Professor of ECE
Director of CHREC

Dr. Greg Stitt*+
Assoc. Professor of ECE

Outline

Overview

- Background and goal
- How to study Exascale w/o Exascale
- Related works

Behavioral Emulation

- Behavioral Emulation (BE) and BE flow
- Behavioral Emulation Objects (BEOs)
- Simulation/emulation platforms
- Novo-G emulation platform

Conclusions and Questions

Background & Goal

- Project conducted by researchers from CHREC*
  - As part of Center for Compressible Multiphase Turbulence (CCMT)
- CCMT supported by DOE, National Nuclear Security Admin
  - Advanced Simulation and Computing Program (PSAPP+ Program)
  - In first year of 5-year support

- CMT poses a grand-challenge problem
  - Significant importance in many environmental, industrial, & national security applications
  - Objective is for CMT simulation code to run on *Exascale systems* for fundamental breakthroughs

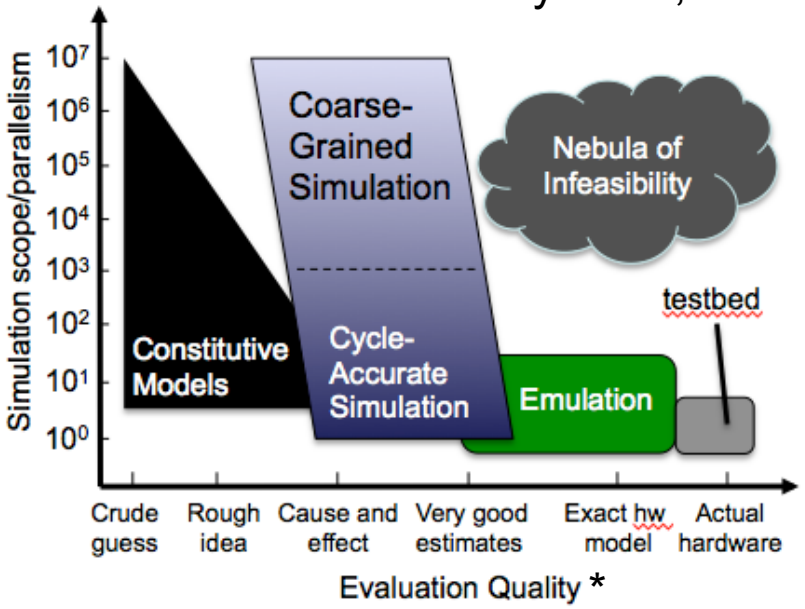
Project Goal: Study Exascale before existence of Exascale to provide advanced visibility for CMT studies

How to Study Exascale Systems?

- How may we study Exascale w/o Exascale?
  - **Analytical studies** – *systems too complicated*
  - **Software simulation** – *simulations too slow at scale*
  - **Behavioral emulation** – *to be defined herein*
  - **Functional emulation** – *systems too massive and complex*
  - **Prototype device** – *future technology, does not exist*
  - **Prototype system** – *future technology, does not exist*
- Many pros and cons with various methods
  - We believe **behavioral emulation** is foundation in terms of balance (accuracy, timeliness, scale, versatility)

Related Works

- System (macro-scale) simulators
  - **ROSS:** C. D. Carothers et al., 2013, 2002
  - **SST MACRO:** C. L. Janssen et al., 2010
  - **FASE:** Grobelny, Bueno, Troxel, George, and Vetter, 2007
  - **BIGSIM:** G. Zheng, G. Kakulapati, L. V. Kale, 2004
  - **ISE:** George, Fogarty, Markwell, and Miars, 1999.
  - **PARSIM:** A. Symons, V. L. Narasimhan,1995.

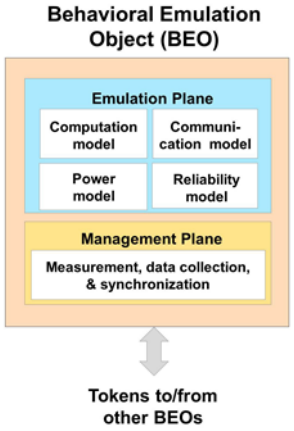


Reference list in Appendix

- Device (micro-scale) & node (meso-scale) simulators
- Object-oriented system simulation
- Hardware emulation
- Analytical modeling
- Supercomputer-specific simulation

* Gilbert Hendry, Joseph Kenny, Jeremy Wilke, and Benjamin Allan. SST/macro Tutorial: IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2013

Behavioral Emulation (BE)

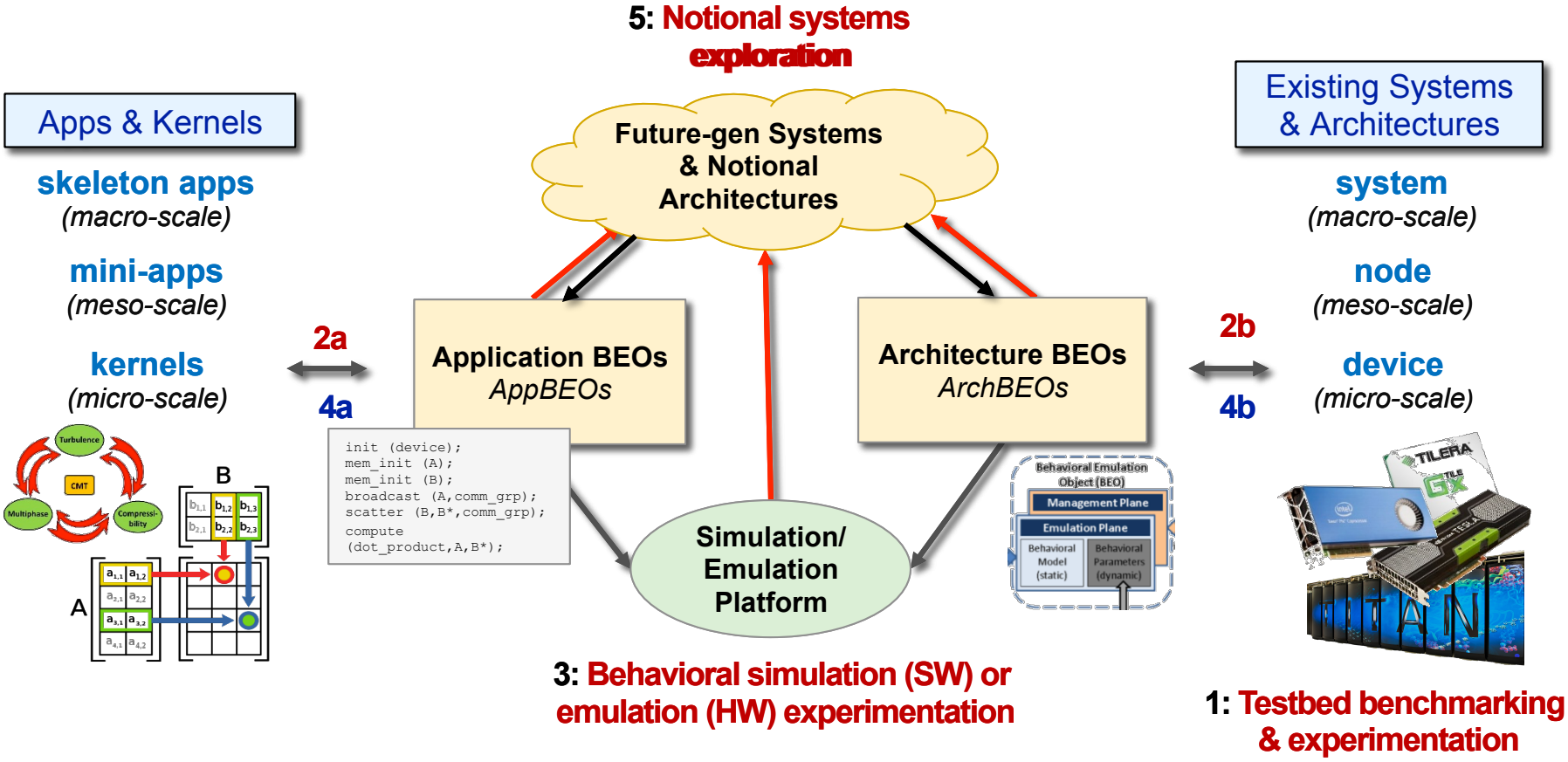


- ***Component-based*** simulation
  - Fundamental constructs called Behavioral Emulation Objects (BEOs)
  - Characterize & represent Exascale application, devices, nodes, & systems as fabrics of interconnected Architecture BEOs & Application BEOs
- ***Multi-scale*** simulation
  - Hierarchical method based upon experimentation and exploration

	Apps	Arch	BEO Models
Macro Level	Skeleton-apps	***System*** BEO fabrics	▪ Models abstracted from Meso-scale ▪ Testbed experimentation in support ▪ Notional ***Exascale system*** exploration
Meso Level	Mini-apps	***Node*** BEO fabrics	▪ Models abstracted from Micro-scale ▪ Testbed experimentation in support ▪ Notional ***Exascale node*** exploration
Micro Level	Kernels	***Device*** BEO fabrics	▪ Architectural studies ▪ Testbed experimentation as foundation ▪ Notional ***Exascale device*** exploration

- ***Multi-objective*** simulation
  - Performance, power, reliability, and other environmental factors

BEOs & Behavioral Emulation Flow

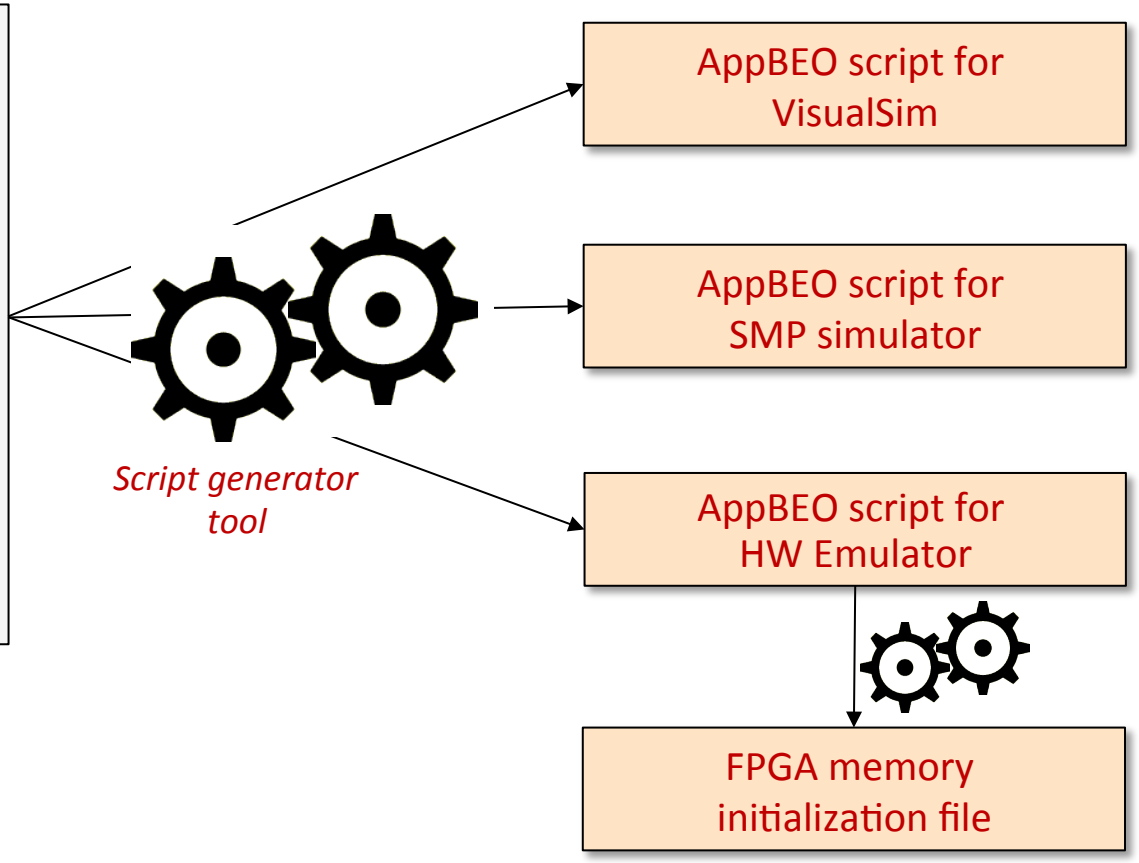


Example: AppBEO for Matrix Multiply

- Mimic application behavior
  - Pseudo-code like sequence of high-level instructions with custom API

```
if (node==0) {
   init (device);
   mem_init (A);
   mem_init (B);
   broadcast (A,comm_grp);
   barrier ();
   scatter (B,B*,comm_grp);
   compute (dot_product,A,B*);
   gather (result,comm_grp);
} else {
   recv (A,node_0);
   barrier ();
   recv (B,B*,node_0);
   compute (dot_product,A,B*);
   send (result,node_0);
}
```

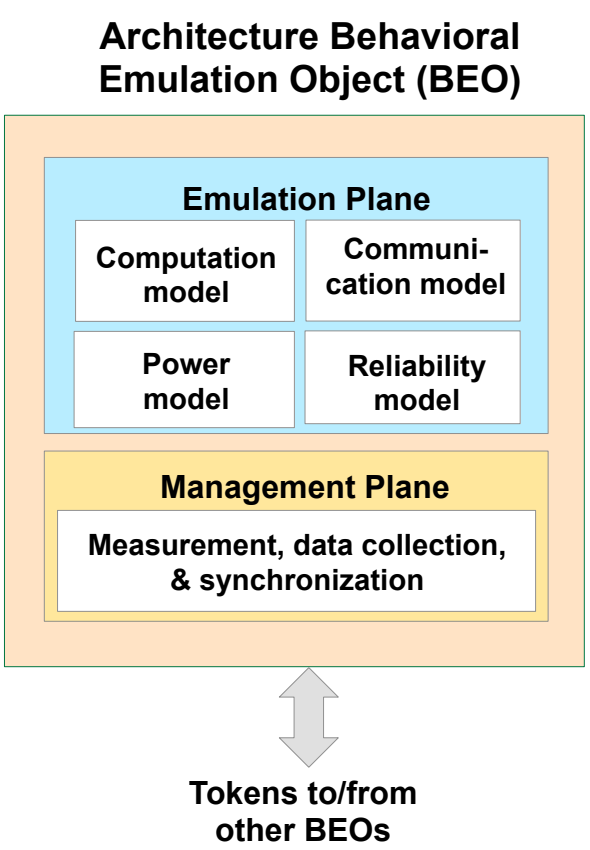
High-level AppBEO script showcasing parallel matrix multiply (C=BxA)



Fundamental Design of an Arch BEO

Arch BEO: *Abstract model (surrogate) of an architecture object*

- Basic primitive in BE approach to studies of Exascale systems



Emulation Plane

- *Mimic* appropriate behavior of BEO
- *Interact* with other BEOs via tokens to support emulation studies

Management Plane

- Measure, collect, and/or calculate *metrics and statistics*
- Support architectural exploration

Metrics

- *Performance factors* (execution time, speedup, latencies, throughputs, etc.)
- *Environmental factors* (power, energy, cooling, temperature)
- *Dependability factors* (reliability, availability, redundancy, overhead)

Example: ProcBEO for TILE-Gx36*

- Mimic behavior of TILE-GX36 device
  - Read and decode AppBEO instructions
  - Resolve computes (determine performance)
  - Update local clock
  - Assign communication instructions to CommBEO

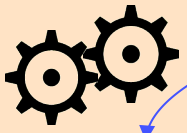
Pseudo-code for ProcBEO

```
if (init) {
   clock=clock+t_init}
if (mem_init){…}
if(compute_dot_product){…}
if(scatter){…}

          ...
```

data size	Time (ns)
8	487.47
16	917.48
32	1,781.68
64	3,509.27
128	6,965.78
256	13,877.84
512	27,703.63
1024	55,401.93

TILE-Gx36 training data (testbed benchmarking) for dot-product parameters: data_size,int64, local mem



Iteratively refine & celebrate model

Model

`execution_time = f()`

Train interpolation model

Predicted execution time

Exceeds error threshold?

- Radial Basis Function
- Table Lookup
- K-Nearest Neighbor
- Kriging
- Linear Interpolator
- …

Interpolation techniques

data size	Time (ns)
100	5,455.77
200	10,855.59
300	16,255.47
700	37,915.54

Test data (different than training data)

* TILE-Gx36: 2D-mesh, many-core processor

Example: CommBEO for iMesh

- Mimic Tilera iMesh network behavior
  - Topology, routing policy, arbitration, etc.

	Time (ns)	Throughput (Mbps)
Neighbors	20.5	3,117.355
Side-to-Side	24.5	2,608.717
Corners	30	2,129.44

iMesh one-way latencies and throughput

Direction	Time (ns)
x-x	1
y-y	1
x-y	1

Switching time

TILE-Gx36 iMesh benchmarking data

Pseudo-code for CommBEO

```
if (input_buffer!=empty) {
   read_event;
   if(output_buffer !=full) {
      forward(x_dir, y_dir);
   }
}
            ...
```

```
Topology: 2D mesh
Routing policy: dim-order
Routing policy: cut-through
X-dir latency: testbed data
Y-dir latency: testbed data
Arbitration: round-robin
            ...
```

Network configuration parameters for TILE-Gx36 iMesh

Exascale Simulation/Emulation Platforms

Emulation Platform

- BEOs representing Exascale devices, nodes, or systems mapped to emulation platform
- BEO method independent of platform types
  - Discrete-event modeling & simulation tool (e.g., VisualSim)
  - Parallel simulation tool on conventional (many-core) computer
  - Software-defined hardware on reconfigurable supercomputer (e.g., Novo-G)

- *Commercially available, flexible, ease of use*
- *For small-scale devices, nodes, and systems*

- *Emulation platform to be developed in software*
- *Higher performance than simulators, but insufficient for Exascale*

- *Even the proposed BEO approach to emulation is challenging for studying Exascale systems*
  - ✓ Exascale, multiscale, multiobjective
- *Reconfigurable hardware to provide performance and scalability required for study of extreme-scale systems*

Novo-G Reconfigurable Supercomputer

- Developed and deployed at CHREC
  - Most powerful reconfigurable computer in academic world
  - 2012 Alexander Schwarzkopf Prize for Technology Innovation @ NSF
- App acceleration
  - In key science domains: bioinformatics, finance, image & video processing
- Hardware emulation
  - Behavioral emulation of future-gen systems, up to Exascale
- 2014 upgrade
  - 32 GiDEL ProceV (Stratix V D8)
  - 4x4x2 3D-torus or 5D-hypercube
  - 6 Rx-Tx links per FPGA
  - 4x 10 Gbps per link

Novo-G Annual Growth

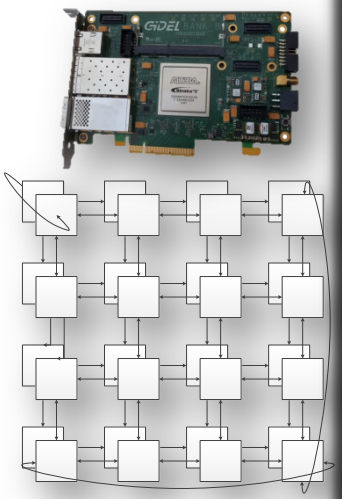
2009: 24 GiDEL ProcStar III cards (96 top-end Stratix-III FPGAs), each with 4.25GB SDRAM

2010: 24 more ProcStar III cards (96 more Stratix-III FPGAs), each with 4.25GB SDRAM

2011: 24 ProcStar IV cards (96 top-end Stratix-IV FPGAs), each with 8.50GB SDRAM

2012: 24 more ProcStar IV cards (96 more Stratix-IV FPGAs), each with 8.50GB SDRAM

2014: 32 ProceV cards (32 top-end Stratix-V FPGAs), with high-speed 4x4x2 torus



Funded by U of Florida w/ generous help from Altera and GiDEL

Conclusions & Questions

What is the major contribution of your research?

- A novel approach for behavioral simulation & emulation of large systems and applications *up to Exascale*
  - At multiple scales & for multiple objectives
- Use of reconfigurable hardware (FPGAs) to provide performance and scalability required for study of extreme-scale systems

What is bigger picture for your research area? (ident. synergistic projects, complementary projects in technical sense, etc.)

***Big picture**: DOE's $100M effort on Exascale arch exploration*

- Coarse-grained simulation approach (*rapid virtual prototyping, RVP)*
- Provide a first-order approximation for design-space exploration
- Complementary to other (detailed & slow) *fine-grained* simulation/ emulation efforts

What are gaps you identify in the research coverage in your area?

(Looking for synergistic & complementary projects for leveraging & collaboration!)

1) Characterizing processors, networks, apps, et al. at *multiple scales* (single device to exascale system) with *behavioral objects* as *surrogates*
   - *DOE Co-design centers; FastForward and DesignForward for vendor roadmaps to Exascale; parallel coarse-grained and fine-grained simulators*
2) Mapping these behavioral objects onto systems of *reconfigurable processors* to maximize the number and speed of these objects
3) Adapting *synchronization* and *congestion-modeling* techniques to support simulation experiments with millions of these behavioral objects
   - *Parallel large-scale network simulators*
4) Measuring, managing, and *visualizing complex behaviors* in performance, resilience, and energy of systems and apps up to Exascale
   - *Visualization tools for extreme-scale systems*
5) Augmenting initial focus upon performance evaluation of systems and apps to include evaluation of *resilience* and *energy consumption*
   - *Modeling and simulation tools for resilience and energy consumption*

Appendix

References (1)

System (macro-scale) Simulators

- C. Engelmann, and T. Kaughton, "A Hardware/Software Performance/Resilience/Power Co-Design Tool for Extreme-scale Computing", Workshop on Modeling & Simulation of Exascale Systems & Applications, September 18th-19th, 2013. **xSim**
- C. D. Carothers, R. B. Ross, J. S. Vetter, et.al., "Combining Aspen with Massively Parallel Simulation for Effective Exascale Co-Design", Workshop on Modeling & Simulation of Exascale Systems & Applications, September 18th-19th, 2013. **ROSS**
- R. Cledat, J. Fryman, I. Ganev, S. Kaplan, R. Khan et.al., "Functional Simulator for Exascale System Research", Workshop on Modeling & Simulation of Exascale Systems & Applications, September 18th-19th, 2013. **FSim**
- C. L. Janssen, H. Adalsteinsson, S. Cranford, J. P. Kenny, A. Pinar, D. A. Evensky, and J. Mayo, "A simulator for large-scale parallel architectures" International Journal of Parallel and Distributed Systems, vol. 1, no. 2, pp. 57-73, 2010. **SST MACRO**
- E. Grobelny, D. Bueno, I. Troxel, A.D. George, and J.S. Vetter, "FASE: A Framework for Scalable Performance Prediction of HPC Systems and Applications, Simulation", Simulation, Vol. 83, No. 10, pp. 721-745, Oct. 2007. **FASE**
- L. Carrington, A. Snavely, and N. Wolter, "A performance prediction framework for scientific applications". Future Generation Computer Systems, 22(3), 336–346 **PMaC**
- G. Zheng, G. Kakulapati, L. V. Kale, "Bigsim: A parallel simulator for performance prediction of extremely large parallel machines", 18th IPDPS, pp. 78, 2004. **BIGSIM**

References (2)

System (macro-scale) Simulators, continued

- A. D. George, R. B. Fogarty, J. S. Markwell, and M. D. Miars, “An Integrated Simulation Environment for Parallel and Distributed System Prototyping”, Simulation, vol. 72, pp. 283-294, May 1999. **ISE**
- A. Symons, V. L. Narasimhan, "Parsim-message PAssing computeR SIMulator," IEEE First International Conference on Algorithms and Architectures for Parallel Processing, vol. 2, pp. 621, 630, 19-20, ICAPP, 1995. **PARSIM**

Device (micro-scale) & Node (meso-scale) Simulators

- J. Wang, J. Beu, S. Yalamanchili, and T. Conte. “Designing Configurable, Modifiable and Reusable Components for Simulation of Multicore Systems”, 3rd International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems, November 2012. **MANIFOLD**
- M. Hseih, R. Riesen, K. Thompson,W. Song, A. Rodrigues, “SST: A Scalable Parallel Framework for Architecture-Level Performance, Power, Area and Thermal Simulation”, Computer Journal, vol. 55, no. 2, pp. 181-191, 2012. **SST MICRO**
- N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, “The gem5 simulator”, SIGARCH Comput. Archit. News 39, 2 (August 2011), 1-7. **GEM5**

References (3)

Supercomputer-specific Modeling & Simulation

- S. R. Alam, R.F. Barrett, M. R. Fahey, J. M. Larkin, and P.H. Worley, "Cray XT4 : An Early Evaluation for Petascale Scientific Simulation", 2007.
- A. Hoisie, G. Johnson, D. J. Kerbyson, M. Lang, and S. Pakin, "A Performance Comparison Through Benchmarking and Modeling of Three Leading Supercomputers : Blue Gene / L, Red Storm , and Purple", (November), 1–10, 2006.

Hardware Emulation

- Z. Tan, A. Waterman, H. Cook, S. Bird, K. Asanovi, and D. Patterson, "A Case for FAME: FPGA Architecture Model Execution", ISCA'10, June 19–23, 2010, Saint-Malo, France, 290–301.
- J. Wawrzynek, D. A. Patterson, S. Lu, and J. C. Hoe, "RAMP: A Research Accelerator for Multiple Processors", 2006.
- http://www.cadence.com/products/sd/palladium_series/pages/default.aspx
- http://www.mentor.com/products/fv/emulation-systems/veloce

Object-oriented System Modeling

- J. C. Browne, E. Houstis, and J. R. Purdue, "POEMS – End to End Performance Models for Dynamic Parallel and Distributed Systems"

Analytical Modeling

- N. Jindal, V. Lotrich, E. Deumens, B.A. Sanders, and I. Sci, " SIPMaP : A Tool for Modeling Irregular Parallel Computations in the Super Instruction Architecture", IPDPS 2013