Blackcomb Project: Year One

Presented by

Jeffrey Vetter, Dong Li, Gabriel Marin, Collin McCurdy

Oak Ridge National Laboratory

Robert Schreiber

HP Labs

Trevor Mudge

University of Michigan

Yuan Xie

Penn State University

SC11 Seattle, WA

OAK RIDGE National Laboratory

Blackcomb overview

- Funding awarded under the Office of Advanced Scientific Computing Research (ASCR) calls for Advanced Architectures and Critical Technologies for Exascale Computing
- Addresses the call's request for
  - Basic and applied research to address fundamental challenges in the design of energy-efficient, resilient hardware and software architectures and technology for high performance computing systems at exascale
- Solve exascale challenges
  - Power consumption
  - Reliability
  - Concurrency

Blackcomb objectives

- Blackcomb proposes a new distributed computer architecture that addresses the resilience, energy, and performance requirements of exascale systems:
  - Replaces mechanical-disk-based data-stores with energy-efficient nonvolatile memories (NVRAM)
  - Places low power compute cores close to the data store
  - Reduces number of levels in the memory hierarchy
- Addresses device scalability and energy efficiency of charge-based memories, while eliminating the problem of increasing DRAM soft-error rates
- Evaluates the impact of the proposed architectures on the performance of critical DOE applications

Nonvolatile memory (NVM)

- "Nonvolatile" in that it retains data even when not powered
- Currently used as a buffer for secondary storage in today's HPC systems
- However, questions about the future of DRAM scaling, combined with NVM technology advances, make it increasingly attractive as a primary storage substitute

	SRAM	DRAM	NAND Flash	PC-RAM	STT-RAM	R-RAM
Data Retention	N	N	Y	Y	Y	Y
Memory Cell Factor (F^2)	50-120	6-10	2-5	6-12	4-20	<1
Read Time (ns)	1	30	50	20-50	2-20	<50
Write / Erase Time (ns)	1	50	10^6-10^5	50-120	2-20	<100
Number of Rewrites	10^{16}	10^{16}	10^5	10^{10}	10^{15}	10^{15}
Power Read/Write	Low	Low	High	Low	Low	Low
Power (Other than R/W)	Leakage Current	Refresh Power	None	None	None	None

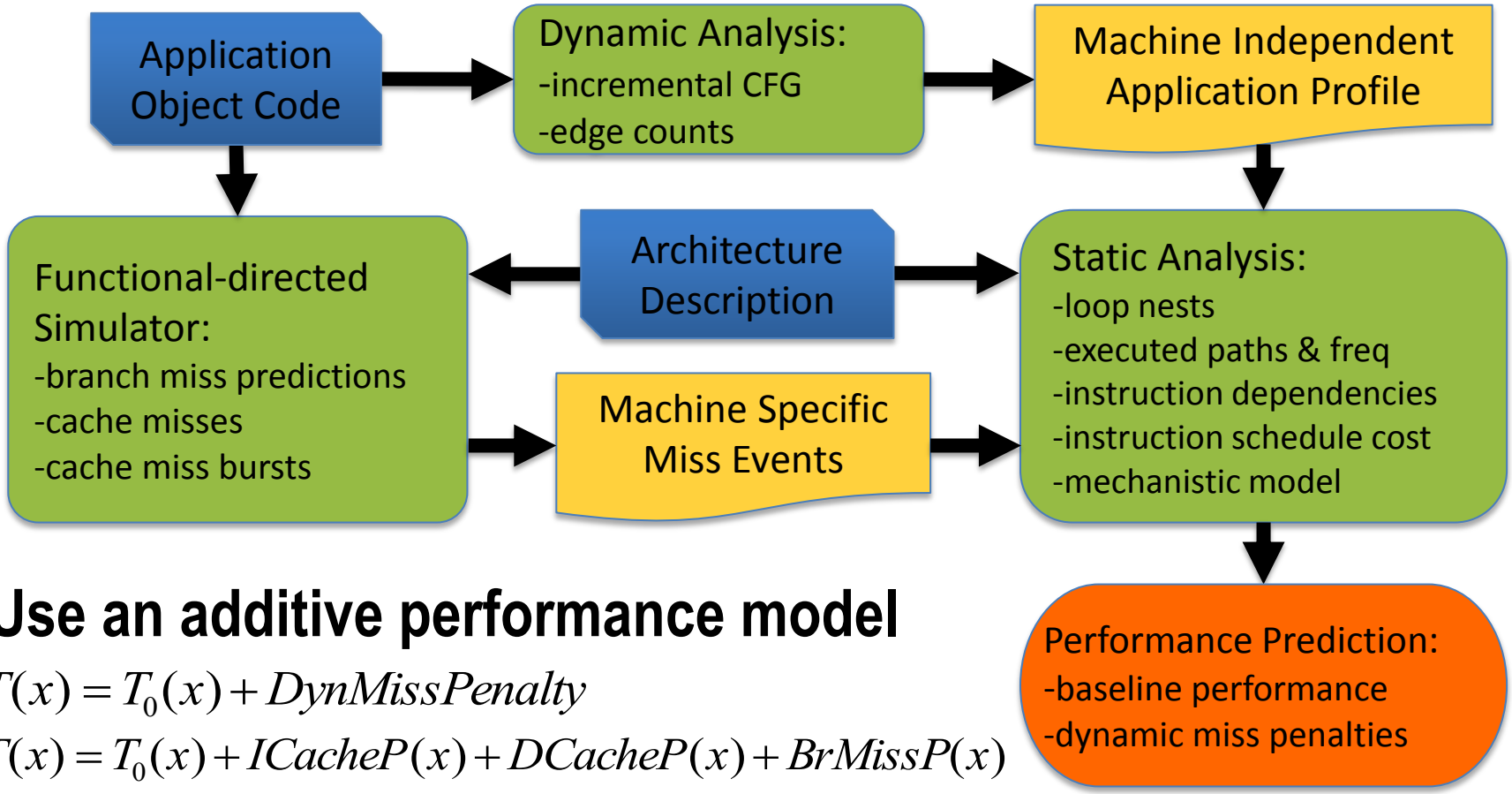
Blackcomb approach

- Identify and evaluate the most promising nonvolatile memory (NVM) technologies
- Explore assembly of NVM technologies into a storage and memory stack
- Propose an exascale HPC system architecture that builds on our new memory architecture
- Build the abstractions and interfaces that allow software to exploit NVM to its best advantage
- Characterize key DOE applications and investigate how they can benefit from these new technologies

Blackcomb challenges

- Understanding and mitigating limitations of NVMs as a general-purpose memory: higher write overheads and lower life endurance than SRAM/DRAM
- Requires novel analytical/simulation hybrid model to understand trade-offs between energy efficiency, resilience and performance
- Requires methodology for evaluating productivity of proposed programming models that exploit NVM to improve fault-tolerance of distributed applications

Blackcomb performance simulator



- **Use an additive performance model**

$$T(x) = T_0(x) + DynMissPenalty$$

$$T(x) = T_0(x) + ICacheP(x) + DCacheP(x) + BrMissP(x)$$

- **$T_0(x)$ – instruction schedule cost w/o dynamic misses**
  - **Use modeling and static analysis to reduce simulation overhead**

Understanding dynamic miss penalties

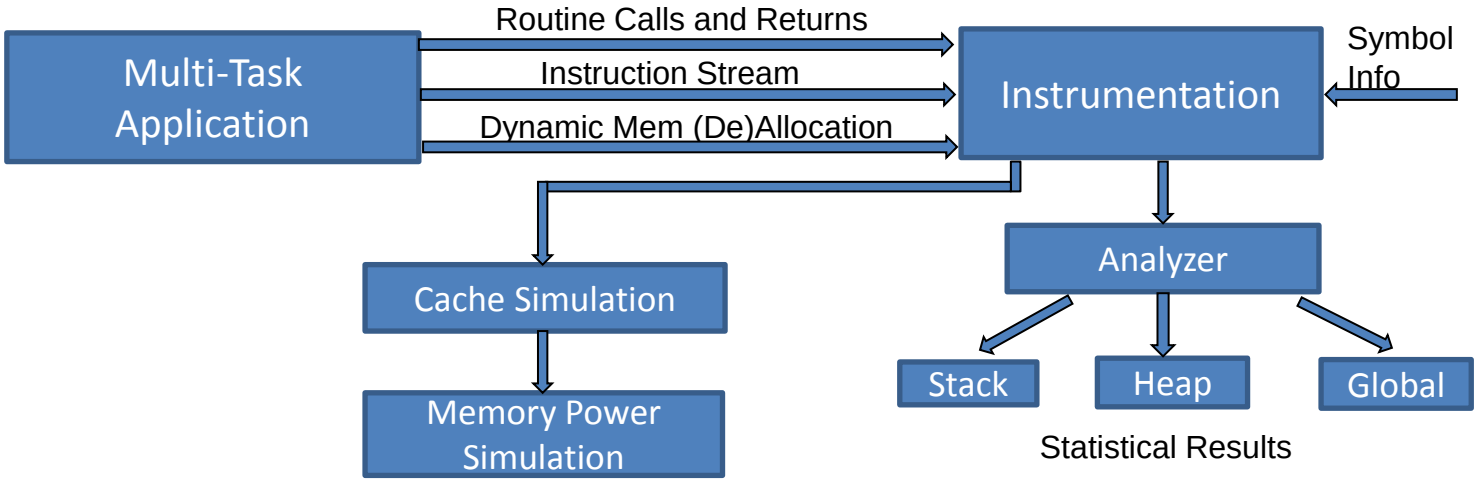
- Use a functional directed simulator to understand overlapped miss events
  - First order model to estimate computation overlap
- Use COTSon infrastructure
  - HP's open source research simulator, built on top of SimNow
  - Supports multicore systems and multi-threaded applications
  - Extensible using plugins
- Functional-directed simulation
  - Cache simulator, branch predictor
  - Simple dependence tracking
    - Understand serialized vs. parallel miss events

NV-SCAVENGER

- Target: understand characteristics of mission critical scientific applications
- A pin-based dynamic binary instrumentation tool
- Monitor heap data, global data and stack data
- Collect memory access patterns at the granularity of memory regions
  - e.g., a memory block coming from heap-based memory allocation
  - e.g., a global data array
  - e.g., a common block in Fortran

Tool Functionality

- Collect memory references from instruction streams
- Simulate a multi-level configurable cache
- Associate memory references info with application code
- Statistically report results for stack, heap and global data



NV-SCAVENGER Observation

- Read-only data widely exists in scientific applications
  - Computational auxiliary data structures
  - Computing-dependent read-only data
  - Physical invariants
- Data structures can be unevenly touched or not touched at all across computation iterations
- Most of the data structures we examined in our experiments have #read_refs > #write_refs

Contacts

Jeffrey Vetter

Future Technologies Group
Computer Science and Mathematics Division
(865) 356-1649
vetter@ornl.gov