John Shalf, David Donofrio: Lawrence Berkeley National Laboratory
Curtis Janssen, Helgi Adalsteinsson: Sandia National Laboratories
Dan Quinlan: Lawrence Livermore National Laboratory
Sudhakar Yalamanchili: Georgia Tech

http://www.nersc.gov/projects/CoDEx
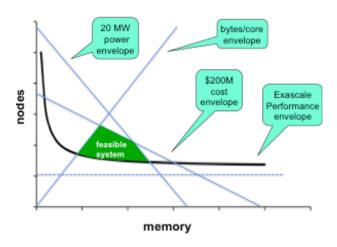
Lawrence Berkeley National Laboratory

CoDEx: CoDesign for Exascale

Architectural Simulation and Modeling for Exascale Platform Development

Exascale computing will require a radical redesign of HPC node architectures to fit within emerging constraints of power, the increasing cost of data movement, and stalled processor clock rates. Applications and algorithms will need to adapt to the evolution of node architectures. The codesign of applications, architectures and programming environments will enable navigation of the increasingly daunting constraint space for feasible exascale system designs to achieve more optimal, balanced systems (see Figure 1). We are assembling a comprehensive hardware/software co-design environment, called CoDEx (CoDesign for Exascale) that will enable an unprecedented opportunity for application and algorithm developers to influence the direction of future architectures so that they meet DOE mission needs. CoDEx combines highly-configurable, cycle accurate simulation of node architectures, developed through the LBNL Green Flash project, with novel automatic extraction and exascale extrapolation of memory and interconnect traces using the LLNL ROSE compiler framework, and scalable simulation of massive interconnection networks using the Sandia-developed SST/macro

coarse-grained simulator. These tools will enable a tightly-coupled software/hardware co-design process can be applied effectively to the complex HPC application space.

The CoDEx project will provide a valuable research vehicle to understand how the evolution of massively parallel chip architectures can be guided by close-coupled feedback with the design of the application, algorithms, and hardware together. This hardware/software co-design process, driven by exascale-driven DOE applications, ensures hardware design decisions do not evolve in reaction to hardware constraints, without regard to programmability and delivered application performance. Further, our architectural simulation process, unencumbered by existing vendor roadmaps, will provide a powerful tool to validate and potentially influence vendor design decisions. Our unique co-design methodology will ensure development of exascale computing systems that deliver high impact across science domains and demonstrate a new model for interaction between laboratories and vendors to create highly effective computing platforms.
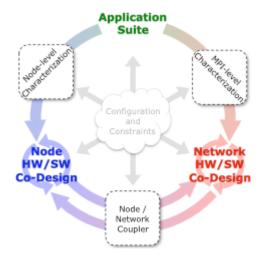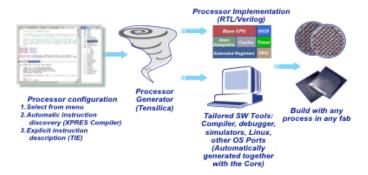
Figure 1: (left) The notional diagram presented by Andy White at the Exascale Crosscut Workshop to describe the complex technology trade-offs at exascale that we hope to navigate using an iterative codesign process. (right) The CoDEx environment provides tools to accelerate the application-driven codesign process.

Key Technology Components of CoDEx

RAMP/Green Flash: The Green Flash platform provides a configurable, cycle-accurate, validated node design to understand the impact of node-architecture choices on application performance. Our research has built a foundation for hardware exploration that enables rapid synthesis of alternative CPU designs, methods for estimating peak power consumption, and benchmarking of full applications using validated hardware accelerated cycle accurate models of the resulting node design. Green Flash takes advantage of the RAMP FPGA-based hardware emulation platforms, which have emerged as a cost effective tool to prototype and run gate-level hardware implementations at near real-time speeds. In this project we will extend our cycle-accurate, node-level simulation tool to enable exploring application performance and practical advanced programming models, together with novel hardware support mechanisms that allow programmers to utilize massive on-chip concurrency, such as direct hardware support for PGAS on a chip. We will also extend our tools to support expanded instrumentation for performance introspection, more accurate power modeling, more scalable synthesis of manycore node designs, and targeted fault-injection to simulate transient errors in order to support the X-Stack resilience research.

Processor configuration
1. Select from menu
2. Automatic instruction discovery (XPRES Compiler)
3. Explicit instruction description (TIE)

Processor Generator (Tensilica)

Tailored SW Tools: Compiler, debugger, simulators, Linux, other OS Ports (Automatically generated together with the Core)

Build with any process in any fab

ROSE Compiler: The ROSE compiler framework enables rapid development of source-to-source translators from Fortran, C, C++, UPC, CUDA, and OpenMP to facilitate deep analysis of complex application codes and code transformations. ROSE is also the basis of important tools such as the Rice CoArray Fortran compiler, and the Utah/ISI CHiLL autotuning compiler. ROSE analysis tools provide architecturally relevant parameters to inform chip designers where to focus their attention,

such as byte-flop ratios, on-chip cache-size requirements, maximum extractable parallelism for a kernel, and data flows required for interprocessor communication. In this project, we will develop the technology to automate the generation of communication models (MPI and PGAS) from real applications. We will extend our simulation and co-design methodology to use the ROSE compiler framework, enabling a new capability that automates the extraction of memory trace and interprocessor communication patterns that can be extrapolated to exascale-class systems. ROSE will also expand our ability to explore new language constructs and programming model options.

SST/macro: Any extreme scale solution requires networks substantially larger than today's petascale machines. As construction of such a network is extremely expensive, it is imperative that architects have access to simulation tools that accurately model large-scale behavior. The SST/macro discrete event simulator enables the evaluation of large-scale interconnect designs that have millions of client endpoints. In this project, we will develop tools utilizing the ROSE compiler framework to generate skeleton applications that can directly drive SST/macro simulations of extreme-scale systems. SST will enable us to evaluate the performance of alternative large-scale interconnects, so that we can answer questions like "what interconnect latency, bandwidth, and topology will offer the best delivered energy and time performance for leading-edge DOE applications." We will develop accurate time and energy models to enable a complete assessment of the trade-offs and costs of different architecture and algorithm choices.

CSO 20952