Traleika Glacier (X-Stack) Project

The Traleika Glacier project will research and mature software technologies addressing major Exascale challenges, and get ready to intercept by the end of the decade. The principals of this project are: Shekhar Borkar & Wilfred Pinfold (Intel), Rich Lethin (Reservoir Labs), Rishi Khan (ETI), Prof Guang Gao (Delaware), Prof Laura Carrington (UCSD), Prof Vivek Sarkar (Rice), Prof David Padua & Prof Josep Torrellas (UIUC), and John Feo (PNNL).

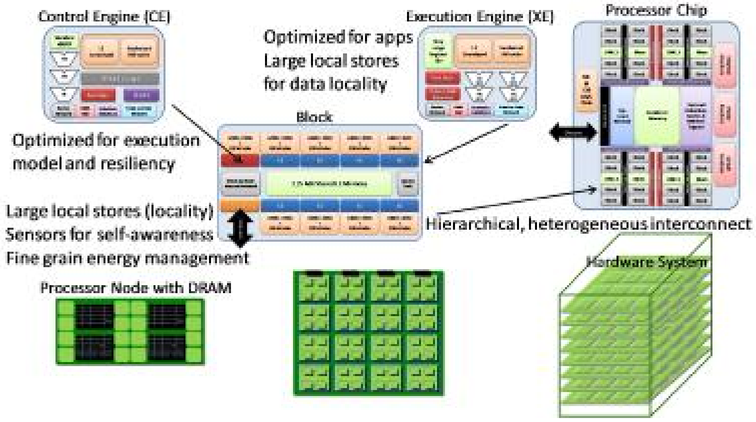
The research agenda of this team addresses the major Exascale challenges, namely (1) energy efficiency, (2) scalability, (3) data locality, (4) programmability, (5) execution model, and (6) resiliency. We plan to address these challenges across the entire stack, identifying solutions that are best implemented with an interdisciplinary approach, working closely with the entire X-Stack research community.

Our programming model primarily focuses on the disruptive part of MPI+X, although MPI cannot be ignored. The programming model adheres to separation of concerns, separating domain specification from HW mapping, expression of data locality, globally shared, non-coherent address space, and generation of HW specific event-driven-threads (EDT). The execution model is data-flow inspired, with self-contained tiny EDTs, with dynamic, non-blocking, event-driven scheduling, observation based adaptation, all implemented in a runtime environment.

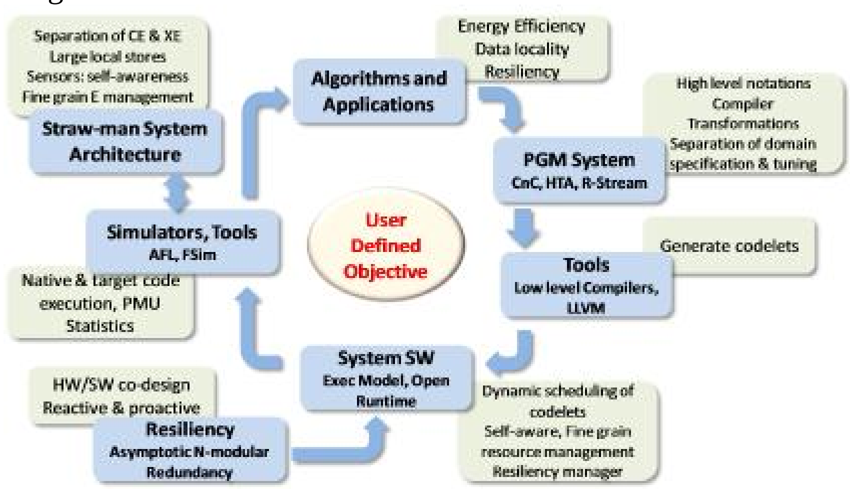
Our approach to the programming system rests on three major components: Concurrent-Collections (CnC), Hierarchical-Tiled-Arrays (HTA), and R-Stream optimizations. The user code is compiled into a parallel intermediate language (PIL, GDG), and EDTs are generated from this intermediate representation for various runtime systems.

Presently we are experimenting with four different runtime systems: (1) Intel research runtime (IRR) targeting the Intel Straw-man architecture, (2) ETI's SWARM for a wide range of parallel machines, (3) Delaware's DAR^{3}TS exploring EDT program execution model using portable C++, and (4) Rice's Habanero-C which interfaces to IRR and CnC. We are working towards converging to a single Open Collaborative Runtime (OCR), adopting salient features from these disparate runtime technologies, and make it available to the community for research; a preliminary version of OCR is already released. Our runtime research will focus on locality aware scheduling, adaptive performance tuning of the hardware, dynamic data movement, and introspective dynamic optimization of the system.

We have instrumented a straw-man architecture adopting the programming and execution models, and the HW implementation of this architecture is evaluated to ensure implementation capable of meeting the Exascale goals. This architecture is captured in two simulators for research, namely AFL and FSIM. AFL works with the Intel research runtime API's and extensions, executes native code on the host processor, runs fast, and also generates useful statistical information for tuning. However, it does not model the straw-man architecture, and is primarily targeted for rapid application development.



FSIM, the functional simulator, models the straw-man architecture, memory and network hierarchies, runs reasonably fast, generates useful statistics, and runs massively parallel and distributed. It is not clock accurate; nevertheless, it is good enough for the entire system evaluation. We are working towards implementing simple timing information in this simulator. These simulators are accompanied by a set of tools for post processing statistics into meaningful results such as energy consumption, data movement, and so on. The simulators are fairly accurate for evaluation using comparisons rather than absolute, and thus will be used to compare architectural features, programming constructs, runtime features, and algorithms.



The flowchart depicts our evaluation flow. We will work closely with the co-design centers to understand proxy applications, extract small idioms and kernels, program them using our programming system, run them in the simulation environment using the runtime system, and evaluate the entire system towards effectiveness of meeting the Exascale goals. This will be an iterative process, with learning incorporated into changes in the architecture, programming system and the runtime. Our research in all disciplines will be evaluated as a complete system using this infrastructure, and thus will be matured for a timely intercept to the Exascale system.

For details, please visit our website: https://sites.google.com/site/traleikaglacierxstack/