# XPRESS

## *eXascale PRogramming Environment and System Software*

## GOAL

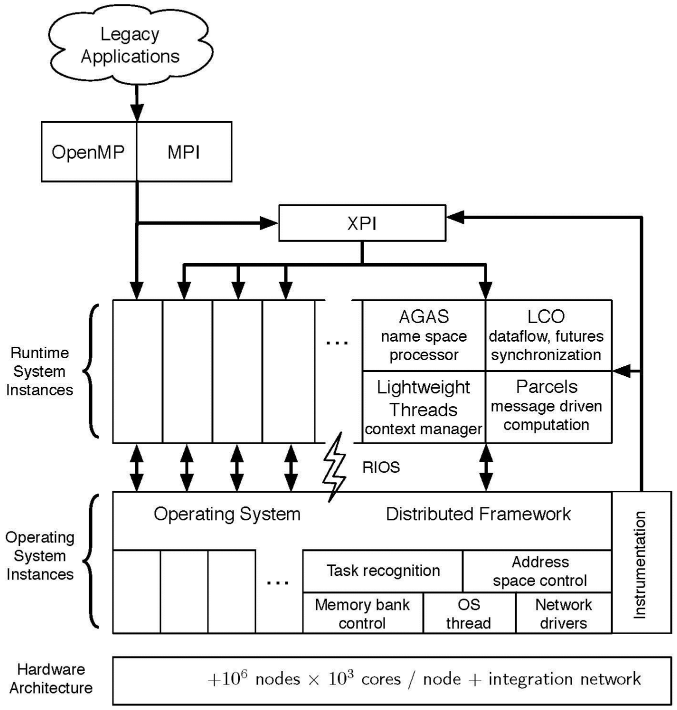
The goal of the XPRESS project is to enable extreme scale computing including exascale by the end of the decade and strong scaling by mid decade. To this end, the XPRESS project is to conduct the research and development of OpenX, a complete system software architecture for Exascale computing. The four principal R&D thrusts include:

1) an Exascale lightweight kernel operating system (LXK), based on the Kitten OS, to manage billion-way hardware parallelism, management of faults and power, management of global virtual name space, and other features of future system architectures;

2) a runtime system (HPX-4), co-designed with LXK, which will be based on the ParalleX execution model and will support dynamic resource management and task scheduling;

3) system interfaces for interoperability between the runtime system and both the OS and APIs; and

4) compilation strategies and systems to translate MPI and OpenMP legacy codes to a form that can be run by OpenX with performance at least as good as a native code implementation.



OpenX Architecture

## TECHNICAL STRATEGY

XPRESS is organized as a set of cooperative tasks to develop and test OpenX. These major tasks include:

- **OpenX software architecture:** a conceptual framework for the co-design and interoperability of proof-of-concept XPRESS software stack
- **ParalleX execution model:** guiding principles for co-design of components of OpenX stack with advances in locality management and task prioritization through introspection
- **HPX runtime system:** support of application dynamic adaptive resource management, task scheduling, and introspective control policies
- **LXK operating system:** lightweight kernel operating system for order constant scalability and low/no noise to manage resources
- **RIOS:** a realm of Exascale system operation unique in the X-stack program. The full system stack including the relationship between the new generation of lightweight kernel operating systems and runtime system software
- **XPI advanced programming model:** intermediate form and low-level (readable) programming interface reflecting the ParalleX model, providing a target for source-to-source high level parallel language translation, and supporting early direct programming experimentation, measurement, and evaluation
- **Performance measurement:** provide parameters and their mutual sensitivities to guide co-design and quantify operational behavior
- **Legacy application mitigation:** ensuring seamless transition of legacy codes in MPI and OpenMP to the future generation of ParalleX based Exascale systems
- **Experiments and evaluation:** critical to determining degree of effectiveness and likelihood of ultimate success as well as guiding corrective design changes to achieve DOE objectives
- **Applications:** collaborations with Co-Design Centers and other mission critical codes
- **Documentation:** as well as reporting to DOE X-stack program management, to provide early adopters with sufficient information to apply prototype programming and execution environment

# XPRESS

## *eXascale PRogramming Environment and System Software*

## YEAR ONE: PROGRESS

- **OpenX software architecture:** defined the components of the for Exascale computing, including ParalleX execution model, LXK operating system, HPX-4 runtime system, interface protocol, compilation methods, debugging tools, instrumentation, fault tolerance, and power management
- **ParalleX:** devised an initial simple model of locality that distinguishes among pair-wise associations with respect to their relative locality
- **HPX:** released HPX V0.9.5 (API and performance counter improvements) and HPXC V0.2 (pthreads). Developed the top-level software architecture for the HPX-4 parcel handler and synchronization
- **LXK:** completed an initial analysis of the operating system (OS) requirements for the HPX-3 runtime system
- **RIOS:** explored the parcel interface which supports message-driven computation to be conducted across the system and between nodes. This is helping to define the emerging protocol
- **XPI:** developed first full specification of the XPI programming interface
- **Introspective adaptation of system & application code:** designed how the information flow between the initial performance tools will occur. Started the implementation to merge the tools and performance models
- **APEX performance:** developed APEX prototype using TAU as the core measurement infrastructure. Instrumented HPX-3 runtime thread manager with APEX timers
- **Legacy migration:** defined strategy for adapting legacy MPI and OpenMP programming models to HPX. Started development of OpenACC compiler. Completed a prototype implementation of data-driven OpenMP execution model

## YEAR TWO: PLANNED ACCOMPLISHMENTS

- **OpenX software architecture:** complete the OpenX software architecture
- **ParalleX:** finalize and document the new generation ParalleX execution model with locality management and introspective scheduling prioritization policies
- **HPX:** complete HPX-3 to LXK stacking and design, and build new parcel handler and LCO operators for HPX-4
- **LXK:** develop functional scalable version of OS for operation, testing, and evaluation
- **RIOS:** publish specification of interface protocols between OS and runtime system
- **XPI:** finalize API specification and implement first reference implementation with interface to HPX-3
- **Introspective adaptation of system & application code:** develop and integrate contention/energy models into HPX and APEX. Improve/increase data sources for models by integrating into LXK. Finish design and start implementing multi-node data collection and contention/energy models
- **APEX performance:** continue to develop and extend the APEX prototype to expose the API for all layers of the XPRESS stack to probe the performance data at the system-wide resolution, with resource-centric reflection and the implicit communication capabilities within HPX
- **Legacy migration:** finish the integration of OpenMP/MPI runtime with HPX. Initial implementation of OpenMP/MPI runtime using XPI interface. Release a prototype implementation of OpenACC compiler in OpenUH
- **Experiments and evaluation:** explore with application codes in XPI, MPI, and OpenMP on top of the OpenX software stack
- **Applications:** conduct initial ports of test applications including Codesign center proxy apps
- **Documentation:** write specification reports for XPI, ParalleX, and RIOS, with Principles of Operation for LXK and HPX-4