

## **XPRESS: Collaboration on System Performance**

LSU, Baton Rouge, February 6th – February 8th, 2013

### **Participants**

- Sandia National Labs: Ron Brightwell
- University of Oregon: Sameer Shende, Kevin Huck
- University of North Carolina (RENCI): Allan Porterfield
- LSU (CCT): Hartmut Kaiser, Steven Brandt, Vinay Amatya, Bryce Lebach

### **Accomplishments:**

- 1) Presentations, discussions, technology reviews and project plans from each team
  - HPX (LSU/CCT)
  - APEX/TAU (UO)
  - RCR (UNC/RENCI)
- 2) Software configured, compiled installed on LSU cluster and test machine at RENCI.
- 3) Demonstrated integrations of APEX & HPX. Integration of APEX and RCR in progress.
- 4) Agreed on possible architecture for performance monitoring subsystem

Workshop page at: <http://stellar.cct.lsu.edu/projects/xpress-collaboration-on-system-performance/>

## Future Work

Fig. 1 shows an overview for a possible architecture and how to integrate APEX into the system. This corresponds very much to the system structure as outlined in the initial XPRESS proposal.

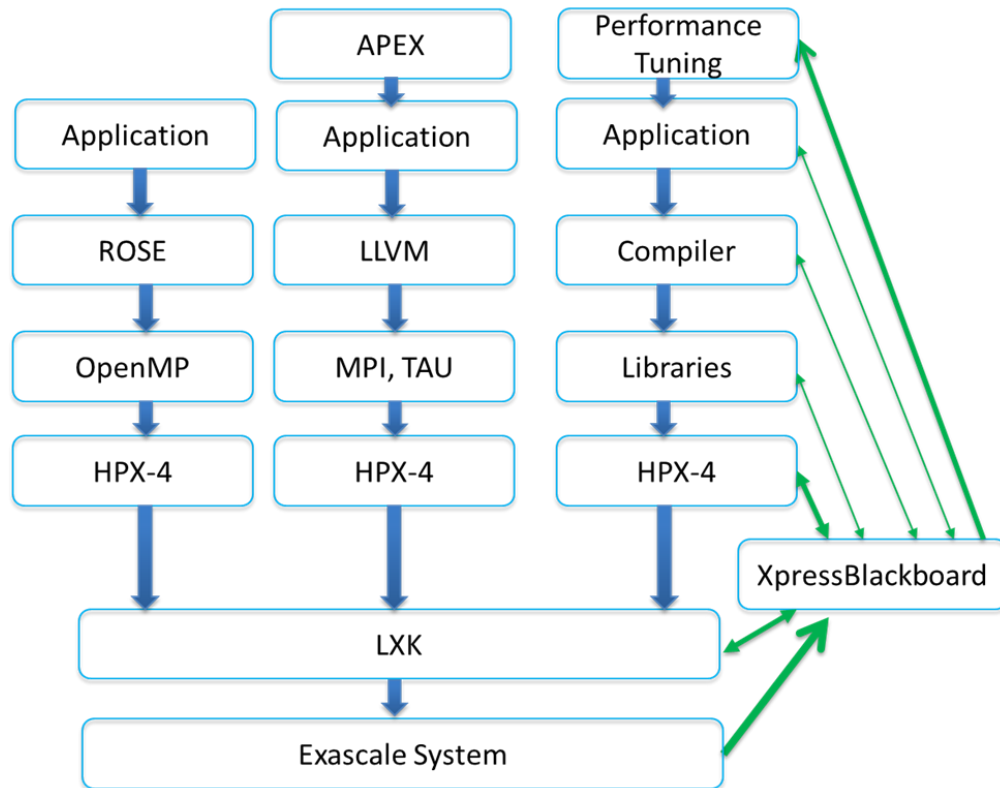


Figure 1: Possible architecture for integrating APEX into the whole system stack.

It demonstrates the need for the performance monitoring subsystem to be integrated with all layers of the XPRESS software stack. From here, after the discussions during the workshop the participants came up with a consensus for a possible structure of the performance monitoring subsystem for XPRESS (see Fig 2). This is based on existing software packages, allowing to leverage existing code for fast prototyping results. We expect to have a first working version available by the end of this year.

Figure 2 also outlines a possible work plan (phase 1, 2, and 3) for the foreseeable future:

Phase 1) at least prototype during workshop:

- 1) Install APEX and RCRdaemon on HPX system at LSU. (needed for use as single tool stack)
- 2) Partially manually instrument HPX to provide information to APEX for display.(demo interaction between APEX and HPX)

Phase 2) started:

- 1) Build HPX with RCRblackboard/provide HPX counter visibility for values in blackboard (demo interaction – provide counters at execution conclusion (?))
- 2) APEX access RCRblackboard counters through HPX counter interface

Phase 3) this year:

- 1) Build contention detector for shared hardware resources (models architecture specific – but need a generic interface for APEX/HPX thread scheduler)
- 2) Connect contention model with APEX data
- 3) Connect contention model with HPX thread scheduling
- 4) Write location/phase information from APEX/Tau instrumentation into RCRBlackboard (demo multi-writer capabilities)

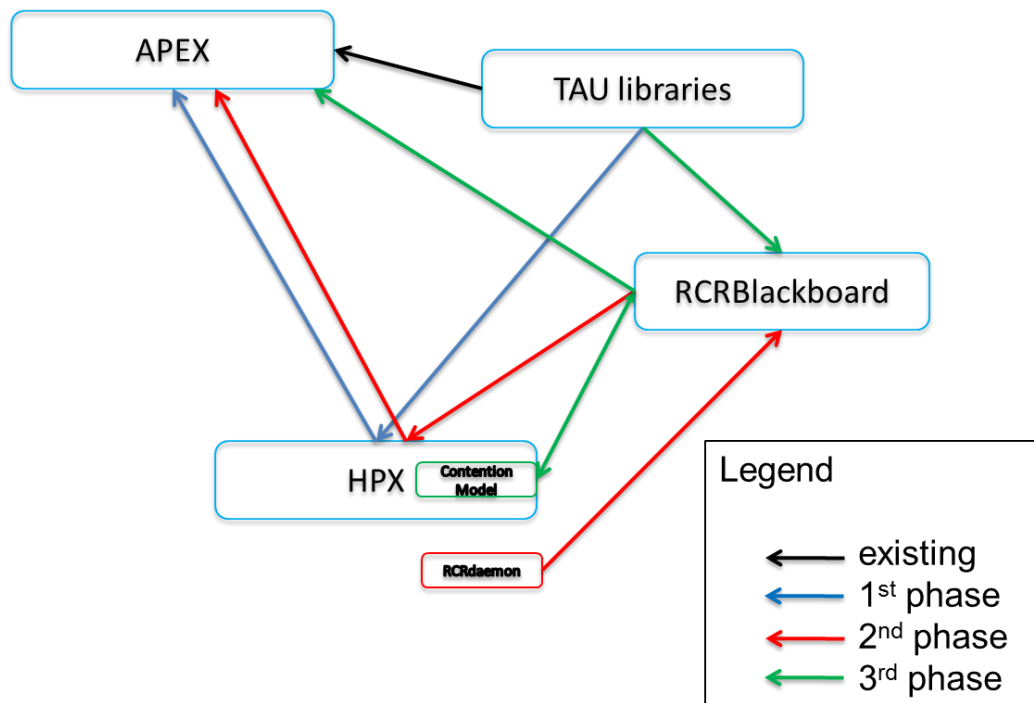


Figure 2: A possible structure of the performance monitoring subsystem

## Work assignments:

### LSU:

- Add support to HPX build system to find librcrtool.a and related headers (HPX ticket [#703](#)).
- Add RCR performance counter component utilizing the rcrBlackboard interface (HPX ticket [#701](#))
  - The HPX PAPI performance counter component is similar in nature, so we can use that as a starting point.
- Abstract performance hooks to ITT (HPX ticket [#699](#))
  - Write performance hooks to APEX using ITT design
- Find reasonable locations to instrument AGAS (HPX ticket [#700](#))
- Instrument HPX parcel transport layer (HPX ticket [#702](#))
- Create performance counter to expose thread queue waiting time (allocation -> schedule), (HPX ticket [#704](#))
- Create aggregating performance counters exposing the rolling average and the median of any underlying performance counter (HPX ticket [#705](#))

### RENCI:

- Add support to HPX build system to find librcrtool.a and related headers.
- Add RCR performance counter component utilizing the rcrBlackboard interface.
  - The HPX PAPI performance counter component is similar in nature, so we can use that as a starting point.
- Come up with a way to allow non-root users to launch RCRdaemon.
- Investigate possibilities of integrating the Sherwood scheduler with HPX

### OREGON:

- Consolidate and cleanup PBS-related scripts for running HPX under APEX
  - Add those scripts to the APEX repository.
- Write a quickstart document for using HPX and TAU.
- Model annotation method for identifying worker threads within “preemption” context.
- “Postprocessing” of node/thread/timer data to locality/stream/thread model

### GENERAL:

- Consolidate workshop notes and distribute to all participants.
- Collect slides and post to the STE||AR group website.
- Put material on the XPRESS website?
- Identify what each group can do over the next few months to help each other and to converge towards project goals.
- Set up APEX mailing list.
  - Instead of doing something like a monthly APEX teleconference, let's have everyone post a monthly/biweekly update to this proposed APEX mailing list (asynchrony!)