Developing the Roadmap

Roughly based on the HEC FSIO roadmap

http://institutes.lanl.gov/hec-fsio/

THE ROADMAP

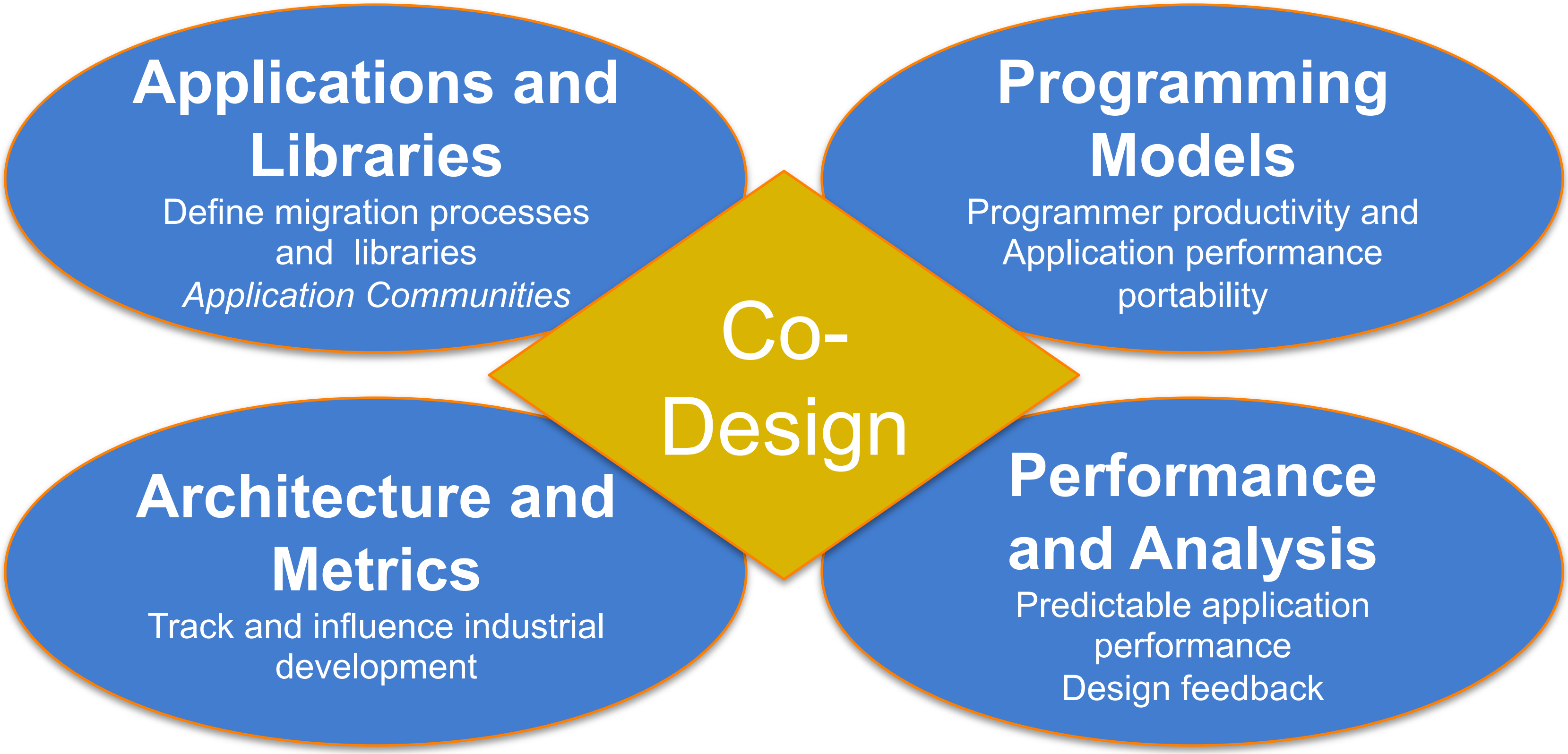
- What it isn’t:
  - A timeline, documenting when technologies will become available
  - A priority ordered list of technologies that need to be developed
  - Frankly, we’d never come to any kind of consensus if we try to do either of these
- What it is:
  - Technologies **we** believe need to be developed to make large-scale, accelerator based systems *production ready*
  - Document relevant projects
  - Identify **gaps** and provide **grades**
  - Dashboard might be a better name

Grading Criteria

Urgency How soon is it needed?	**Duration** How long will it be useful?	**Responsive** How much will money help?	**Applicability** How broadly can it be used?	**Timeline** How soon can we expect it?
Critical Needed now	**Long** Useful for the foreseeable future	**High** Funding enables significant progress	**Broad** Applicable beyond HPC	**Immediate** Results within 1-2 years
Important Needed within 3 years	**Medium** Useful for Exascale	**Moderate** Funding enables progress	**HPC** Applicable to all of HPC	**Soon** Results within 2-5 years
Useful Needed after 3 years	**Near** Only useful for immediate systems	**Low** Funding has little affect on progress	**Narrow** Only applicable to immediate systems	**Eventually** Results after 5 years

Technical Committees (TC)

Pick you poison



Applications and Libraries

Define migration processes and libraries

Application Communities

Programming Models

Programmer productivity and Application performance portability

Architecture and Metrics

Track and influence industrial development

Performance and Analysis

Predictable application performance

Design feedback

Co-Design

<<Topic Area – at most 5 of these>>

- Description
  - <<brief description of the topic area>>
  - Sub areas (if any)
- Notes from Discussion

- Relations to other TCs
  - Fill this in
- Related Projects
  - <<list of existing projects that are relevant to this topic>>

Urgency	Duration	Responsive	Applicability	Timeline

This is a COMMUNITY Effort

- We’d like to keep the grading criteria
  - The criteria and the grades are somewhat arbitrary
  - So is every grading system
  - We’ll change it if it really doesn’t work
- The breakout sessions will start with “strawmen” roadmaps
  - These are intended to facilitate, not stifle discussion
  - Tell us what we’re missing: both related projects and needed technologies
  - Tell us how things ought to be graded
- We’d like consensus…..
  - Make sure that minority views are captured when there are significant differences of opinion

After the Workshop

- Goal is to give everyone an opportunity get their input into the document
- Draft 0.01 (slides from Thursday outbrief) will be on the web site next week
- Draft 0.1 available on consortium website mid-February
- Final version
  - Target: March 15
  - Drop dead: March 30
- Web site and roadmap will have section contact points
- Schedule will be posted and updated on HMC Website
- Announcements sent to hybrid-announce

APPLICATIONS AND LIBRARIES TOPICS

- Critical Libraries
  - BLAS, LAPACK, FFT(W), Sparse MV, C++/STL)
- Early access to systems
  - cluster systems or desktops? will desktops prove viable?
- Workflow for porting/re-factoring applications
  - Is porting sufficient or is rewritting required? Coders need to understand algorithms. Are alternate algorithms better?
- Application Communities
  - Motifs - 13 still?
- Approaches to co-design
  - Apps & Arch are the two legs, but Prog. Model also affect design; Perf. efforts provide input to the process; is 64-bit F.P. always required?

ARCHITECTURES AND METRICS TOPICS

- Components
  - ???
- Accelerator/CPU Coexistance
  - Disjoint/attached or on-die accelerators; accelerator devices per CPU; separate or shared memory; threaded vs. instruction level acceleration
- Accelerator Design
  - SIMD width, threads, ganged vs. separate thread progress, SIMD shuffle/masks, memory latency hiding, memory gather/scatter
- Simulation and Modeling
  - connectivity BW & latency, memory performance; flops
- Node Operations
  - power efficiency, W/flop, packaging, RAS, error detection/ correction; accelerators per CPU

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

Performance and Analysis Topics

- Monitoring, observation and Analysis Tools for systems and applications
  - Memory, node, interconnect, apps
- Code optimization
  - Autotuning, compilation
- Predictive modeling
  - Optimal application-architecture mapping for hybrid
  - Application/architecture co-design
  - Methodology development (modeling of many flavors, simulation)
  - Dynamic (runtime) model-driven system/application optimization

PROGRAMMING MODELS TOPICS

- Data layout
  - application data patterns, memory placement, SIMD considerations
- Desktops to clusters
  - adds internode communication layers; MPI is the defacto standard for clusters; PGAS may be possible but languages slow to develop; easy to bring cluster code back to desktop
- Backward and forward (performance) portability
  - runs on multi-core as well as on hybrid; runs well on both; runs on existing and future
- Existing and emerging language standards
  - compilers, MPI + something else, remote node data access (MPI, PGAS), exposing threads & SIMD
- Low level (OS) APIs for accessing different cores
  - attaching, releasing, & controlling accelerator devices; OS initialization, RAS reporting, resource reporting/selection
- Debuggers & Tools
  - PRINTF