

Characteristics of Programming Models

- Discuss the following abstractions and their semantics, and how they should be represented by the programming model
 - Locality/hierarchy
 - Parallelism
 - Global name space(s)
 - Synchronization
 - Introspection
 - Resilience
 - Communication
 - Time/space/energy management
 - Roles of runtime and compiler support
- What is missing from this list?
- Map these abstractions to application requirements
- What dependencies to PMs have on software stack What are the natural separations of functionality

Abstractions needed by programming models depend on the development level

High Level

(domain scientist)

- Science Geometry
- Locality (implicit?)
- Equations
- Dependencies instead of parallelism
- Data abstraction to express dependencies (refs, aliases)
- Algorithmic name space(s)
- Introspection support (tools, diagnostics)
- Data Flow
- Problem state

Mid Level

(Algorithm specialist)

- Locality/hierarchy
- Dependencies
- Parallel patterns
- Scan, foreach, task DAG
- Introspection
- Error recovery/sensitivity hints
- Containment
- Communication
- Global name space(s)
- Time/space/energy management

Low Level

(Tuning Specialist)

- Explicit placement/movement
- Sync/locking
- Pattern compiler/runtime
- Introspection
- Resilience Notification/Action
- Idempotence
- Global name space(s)
- Communication
- Time/space/energy management
- Roles of runtime and compiler support

Just as there no single exascale programming model, there is no single set of abstractions.

Stockholm Syndrome

- Must identify applications/test problems to test programming models.
 - Map applications/algorithms to prototype PMs
 - Right timing is essential. Managed expectations. Partnerships are critical.
 - Multiple levels & timings for application interaction. Roadmap/milestones.
 - Can't forget that machines will be delivered.
- What is hard/tedious/painful in your current development process. How could a PM help?
 - Most problems are on node. Too much complexity. Can't measure anything. Tools insufficient. No hardware support. Debugging tools. Once measured, how do I analyze/respond? This is just the start of a long list....