Chapel Hierarchical Locales:
Adaptable Portability for Exascale Node Architectures

Complicated compute node architectures got you down?
Wishing for a programming model that adapts to new architectures so your applications don’t have to?

Our Solution:

Chapel’s Hierarchical Locales:

- **Describe target architecture**
  - Resources: CPUs, memories
  - Hierarchical structure: children, parents
- **Implement standardized interface**
  - Tasking, memory, communication
  - Instance of predefined LocaleModel class
  - Implemented in terms of other resources: CPUs, memories
- **Typically coded by affinity/locality expert**
- **Current and planned implementations:**
  - numa (needs tuning)
  - Intel Xeon Phi Knights Corner (KNC, from flat; KNL will be more like numa)
  - numa + accelerator/KNL (to be done)

Chapel’s Domain Maps:

- **Describe distribution of indices (block, cyclic, etc.)**
- **Tie together locality, parallelism, affinity**
- **Interrogate locale model regarding resources**
- **Translate abstract parallelism into actual parallelism**
- **Typically coded by affinity/locality expert**
- **Reusable!**

Chapel Applications:

- **Specify domain maps in application code**
- **Express parallelism abstractly**
- **No reference to physical architecture**
- **Specify locale model as part of Chapel configuration when compiling application**

Chapel in a Nutshell:

- **Compartmentalized for running locale applications**
- **Separation of concerns**
- **Change any of these without changing the others!”

Next Steps:

- Enhance and tune NUMA locale model
- Tune Intel KNC locale model
- Start on Intel KNL and accelerator