More Information

- **http://charm.cs.illinois/edu**
  - Papers, downloads, manuals, tools
- **Mailing list: charm@lists.cs.illinois.edu**
  - [https://lists.cs.illinois.edu/lists/info/charm](https://lists.cs.illinois.edu/lists/info/charm)
- **http://charmplusplus.org**
  - Tutorial material, mini apps, ...
  - A series of programming exercises designed to teach basic concepts in Charm++
    - [http://charmplusplus.org/exercises/](http://charmplusplus.org/exercises/)
- **http://charmplusplus.com**
  - Charmworks Inc., supporting Charm++
Day 2 and beyond: Advanced Concepts

- Projections: Performance Analysis and Visualization, really nice, and a workhorse tool for Charm++ developers
- Application Design case studies
- Threaded entry methods, Futures, …
- Libraries and Modules (Modularization): bound arrays, callbacks
- Priorities
- Entry method attributes
- Quiescence detection
- LiveViz: visualization from a parallel program
- CharmDebug: a powerful debugging tool
- Messages (instead of marshalled parameters)
Advanced Concepts continued

• Processor-aware constructs:
  – Groups: like a non-migratable chare array with one element on each “core”
  – Nodegroups: one element on each process

• Within node parallelism: CkLoop(), taskQueue, nodeGroup

• Using GPGPUs: HAPI: async interface, using streams effectively, ..

• Sections