2.2 Findings – 2011-2012

Our major findings in this period (September 2011 to August 2012) were:

- **Charj:** The use of a programming language specific to the message-driven programming model can result in significantly shorter applications with equivalent performance and functionality. It also provides an opportunity for improved warning and error messages based on semantic knowledge of the programming model, reducing the chances for model-specific errors. Furthermore, it exposes opportunities for compiler optimization that cannot be performed by compilers targeted at sequential languages because of their model-specific nature.

- **Charisma:** The addition of three simple constructs, namely (1) *publisher-directed* communication patterns, (2) index subspaces and (3) parameter value range publication significantly broaden the scope of expression of Charisma, allowing us to write programs that could not otherwise have been written in the language.

- **Divide-and-Conquer:** A framework that automates task grain size control and provides dynamic agglomeration of data-parallel operations for efficient communication can improve parallel performance significantly. The framework engenders productivity by allowing the natural, fine-grained of divide-and-conquer algorithms.

- **Generic Programming:** More thorough support for template metaprogramming in Charm++ enables clearer, more concise expression of application logic in the OpenAtom code. For example, algorithms that can operate in terms of various numeric types (single versus double precision, real versus complex) now only need to be expressed once, with the attendant benefits for readability and correctness.

- **HPC Challenge:** The suite of developed benchmarks achieved admirable performance on multiple platforms, with substantially less code than reference implementations. The necessary code also exhibited better separation of concerns between algorithmic logic necessary for correctness and tuning logic necessary for performance. The separation was measured not just through the volume and structure of the resulting code, but also through version control logs recorded over the course of development. These benchmarks were submitted to the HPC Challenge Competition at Supercomputing 2011, where they were awarded a Class 2 (Productivity) prize for overall performance. Additionally, CharmLU was used to study the performance effects of various mapping schemes on performance.