Charm++ helps creators of compute-intensive software deliver better performance and scalability in shared and distributed memory parallel applications.

Through features enabled by the core principle of processor virtualization, it delivers higher scalability, increased throughput, and reduced development and maintenance effort. Charm++ provides job-size flexibility, fault tolerance, and dynamic load balancing.

**Coded on Laptops, Run on Supercomputers: Numerical Relativity**

The SpECTRE code runs Discontinuous Galerkin space-time simulations around black holes. It was coded and debugged on the devs’ 4-core laptops.

The same code scaled unchanged to NCSA's Blue Waters supercomputer with 100,000 cores!

**When MPI stutters, Charm++ Scales 1000x: Agent-Based Epidemiology**

EpiSimdemcs was ported from MPI to Charm++ when scaling stopped at under 512 cores. The switch enabled efficient fine-grained communication, asynchronous execution, overlapped communication with computation, and high-quality load balancing.

After porting, the developers were able to scale the code a thousand-fold, to run larger problems efficiently on full supercomputers. The code exploits Charm++ MPI interoperability support to reuse existing IO library code, sparing developer effort.

**From TBB to Charm++: Asynchronous Contact Mechanics**

The ClothSim Application presents massive dynamic load imbalance. Charm++ enabled 2x speedup over TBB on a single node. The same code scaled to distributed memory, providing an overall 12x speedup from baseline.

**Blending OpenMP with Charm++: Sparse Approximate Matrix Multiply**

The SpAMM kernel dynamically forms an octree of matrix products, using sub-block norms to occlude negligible contributions. This runtime structure benefits from Charm++'s ability to efficiently construct and work on sparse collections, and to load balancing them based on observed workload. This application runs in a hybrid Charm++-OpenMP execution model, showing further interoperability.

**Summary**

Charm++ provides a wide range of benefits to parallel programmers. They can express sophisticated parallel algorithms, to quickly solve bigger problems and get faster solutions. The resulting programs scale smoothly to the largest systems, and port to new architectures with minimal burden. Charm++ interoperates well with the broader HPC code ecosystem. Applications run flexibly and resiliently in diverse environments.

- Bhatele et al., “Massively Parallel Simulations of Spread of Infectious Diseases over Realistic Social Networks”, CCGrid 2017.