CharmROSS

Empowering PDES with an Adaptive Runtime System

UIUC: Eric Mikida, Nikhil Jain, Laxmikant Kale
RPI: Elsa Gonsiorowski, Chris Carothers
LLNL: Peter Barnes, David Jefferson
Overview
Charm++ Model

- Asynchronous objects
  - Migratable
  - Communicate via remote method invocation
  - Over-decomposition: many objects per core
  - Location and naming managed by RTS

- Message-driven communication
  - Only objects with work get scheduled
  - Overlap of computation and communication
Motivation and Goals

● Achieve similar performance to MPI ROSS
● Add new capabilities
  ○ Asynchrony (GVT)
  ○ Load balancing
  ○ Fault tolerance
  ○ Checkpoint restart
  ○ Fine-grain message aggregation
Port Status

- Sequential, Conservative, Optimistic all work
- Deterministic and consistent with original
- 3 models (PHOLD, PCS, Dragonfly)
- Charm: 4k SLOC, MPI: 8.8k SLOC
- Some extra features implemented
Features
New Features

- GVT Asynchrony
  - Async broadcasts
  - Async reductions
  - Fully async GVT

- Migratability
  - Load balancing
  - Checkpoint/Restart
  - Fault tolerance
Asynchronous Start

Idle Time

PE 0

PE 1

PE 2

PE 3

Time
Asynchronous Start

Start GVT with asynchronous broadcast

PE 0
PE 1
PE 2
PE 3

Time
Asynchronous Reductions

Events

GVT

QD Red FC

Time
Asynchronous Reductions

Events

GVT

QD  Red  FC

Time
Fully Asynchronous GVT
Migratibility

- LPs are migratable
- Load balancing
- Checkpoint/Restart
- Fault Tolerance
Performance
Initial Performance

- Runs done with PHOLD benchmark
- 1 rack of Vesta (1024 BG/Q nodes)
- 64 threads per node
- No new features included
Async GVT Comparison

- Runs done with PHOLD benchmark
- 512 nodes on Vesta
- 64 threads per node
- 50% remote event rate
GVT Comparison

Event Rate (billions/second)

- MPI ROSS: 1.0650
- CharmROSS: 1.0941
- Async Start: 1.0674
- Async Red: 1.0911
- Both: 1.0634
- Fully Async: 1.0429
Conclusion
Future Work

- Tuning/optimization of async features
- PDES specific load balancing
- Topological Routing and Aggregation Module