SAND2014-16524PE

Sandia National Laboratories

Exceptional service in the national interest

DHARMA: Distributed asyncHronous Adaptive Resilient Management of Applications

Janine Bennett, Robert Clay, John Floren, Ken Franko, Saurabh Hukerikar, Samuel Knight, Hemanth Kolla, Greg Sjaardema, Nicole Slattengren, Keita Teranishi, Jermiah Wilke

PSAAP II CCMSC Programming Models Deep Dive
July 28, 2014

U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Mission

- Assess & address fundamental challenges imposed by the need for performant, portable, scalable, fault-tolerant programming models at extreme-scale
- Two focus areas
  - Programming model analysis for next generation platforms
  - Demonstration of fault-tolerant programming model at extreme-scale

DHARMA is a fundamental Hindu concept referring to
- the order and custom which make life and a universe possible
- the behaviors appropriate to the maintenance of that order

The classical Sanskrit noun DHARMA derives from dhr
- meaning to hold, maintain, keep

Programming model analysis for next generation platforms

Sandia National Laboratories

- Asynchronous many-task programming models are a leading new paradigm with many variants
- Goal: Address knowledge gaps
  - Comparative analysis of leading candidate solutions
  - Quantitative & qualitative tests using ASC-relevant codes
- Outcome: Guidance to code development road map for next generation platforms for ASC/Integrated Codes

Runtimes
Uintah
Legion
HPX
Charm++
STAPL
Swift/T

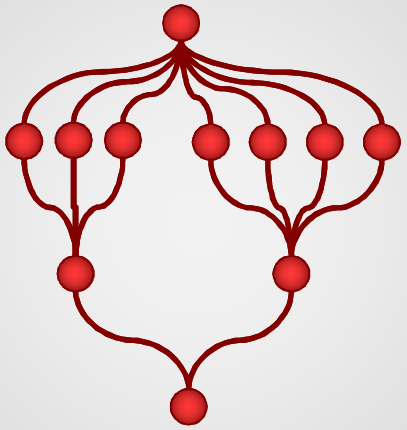
Assess
Scalability
Performance
Resilience
Interoperability with MPI

Demonstration of fault-tolerant programming model at extreme-scale

Sandia National Laboratories

- Asynchronous many-task programming models
  - + Show promise at sustaining performance
  - + Work stealing enables load balancing
  - + Failed tasks can be re-executed
- Recovery (beyond checkpoint/restart) is challenging
  - - Distributed coherency problem
  - - Care is required to identify lost tasks due to work-stealing and asynchrony

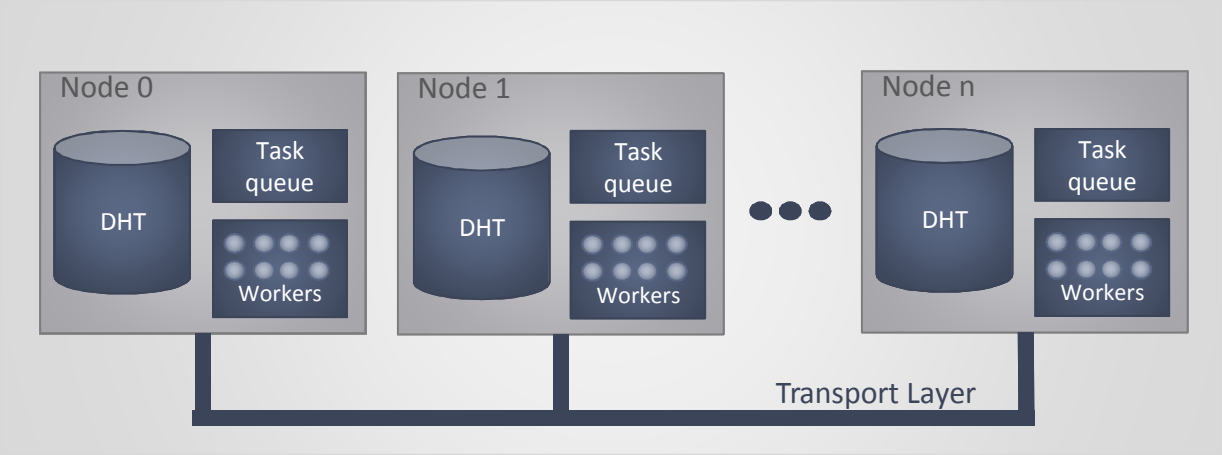
Task Graph



Nodes are tasks
Edges are data

A holistic solution requires a number of fault-tolerant components

- Distributed Hash Table (DHT): Store task descriptors/data pointers
- Collection/task queue: Maintain state & work assignments
- Resilient Transport Layer
  - Fault-aware collectives: terminate cleanly with no result
  - Fault-tolerant collectives: heartbeat via overlay network to rigorously agree on which nodes are alive

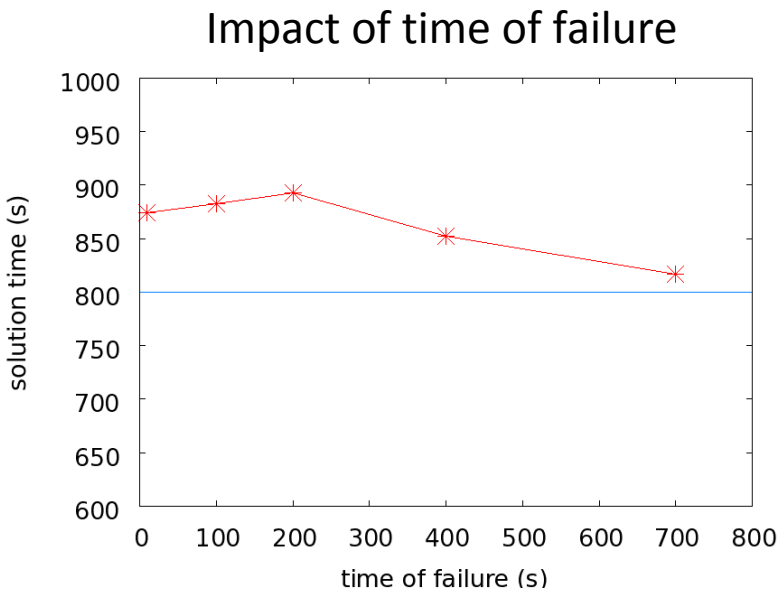


We have implemented the DHARMA runtime system in SST

Sandia National Laboratories

- Coarse-grained simulation allows for system-level exploration
- Skeletonized mini-apps of explicit and implicit solver

Runtime studies	Algorithmic studies
Scalability with no faults (strong and weak)	Task-granularity and decomposition
Performance in the presence of faults	Classification of performance according to compute/communication ratios
Node degradation tests	Algorithmic tradeoffs
Comparison against baseline MPI skeleton	Matrix assembly variants made possible by shift to many-task model



- Demonstration of full-scale implementation of run-time and associated mini-apps on capability-class system next year