CCMT

SAND2015-3547PE

Scalable Network Simulations

Nalini Kumar

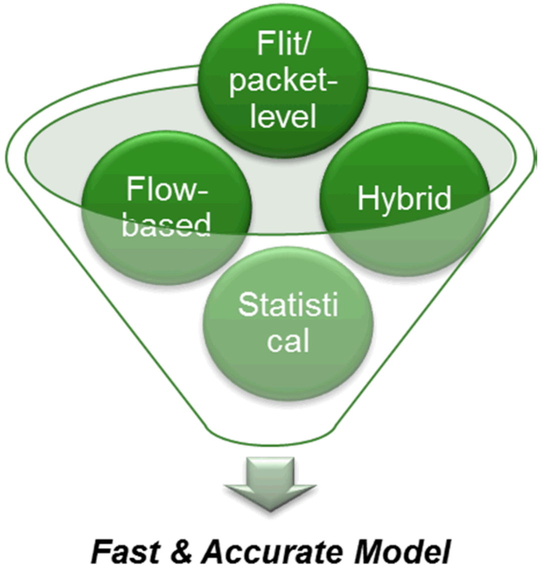
PhD Student, ECE, University of Florida

Sandia National Laboratories

Scalable Network Simulation

- Explore existing congestion models for use in Behavioral Emulation
  - Most recent simulators use low-level network models
  - SST* (Micro) uses high-fidelity component models for system simulations
  - SST (Macro) uses very coarse-grained models for system networks
  - FSIM allows functional network simulation and BigSim allows high-level latency models and detailed model of communication fabric

- Developing highly-scalable parallel simulator is a big-task
  - We are looking at leveraging existing simulator cores/frameworks to support network modeling using our Behavioral Emulation approach
  - Reduce development and support effort, and possibly leverage existing models developed by other users of the tool



* Structural Simulation Toolkit

Characterizing Communication in CMT-Nek

- First we need to understand communication behavior of target CMT-Nek app
  - Since full application is too complex and cumbersome to do targeted study, we are using **'CMTBone'** miniapp

- Nearest-neighbor update using pairwise exchange:
  - No. of transfers per MPI rank = 6
  - Best-case, all exchanges across all MPI ranks occur in parallel
  - Worst-case, all transfers are serialized = $6P$
  - Average transfer size = $6N^2\left(\frac{E}{P}\right)^{\frac{2}{3}}$; total data transferred = $30N^2\left(\frac{E}{P}\right)^{\frac{2}{3}}$

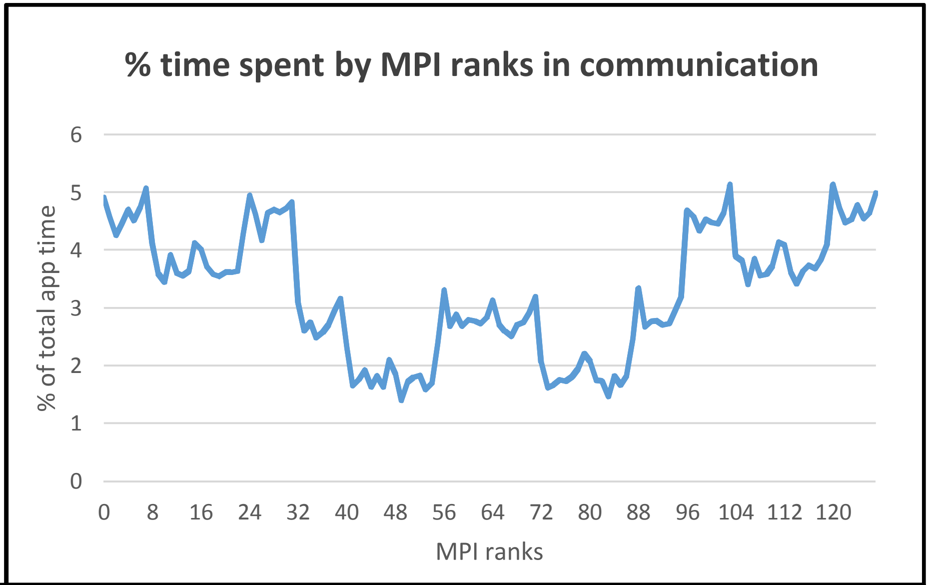
- Nearest-neighbor update using crystal router:
  - No. of transfers per MPI rank = Optimal no. of transfer steps = log_2P
  - Transfers at each comm stage = P ; Total no. of transfers = $P\ log_2P$
  - At each transfer stage, largest transfer size = $6N^2\left(\frac{E}{P}\right)^{\frac{2}{3}}$; total data transferred > $30N^2\left(\frac{E}{P}\right)^{\frac{2}{3}}$

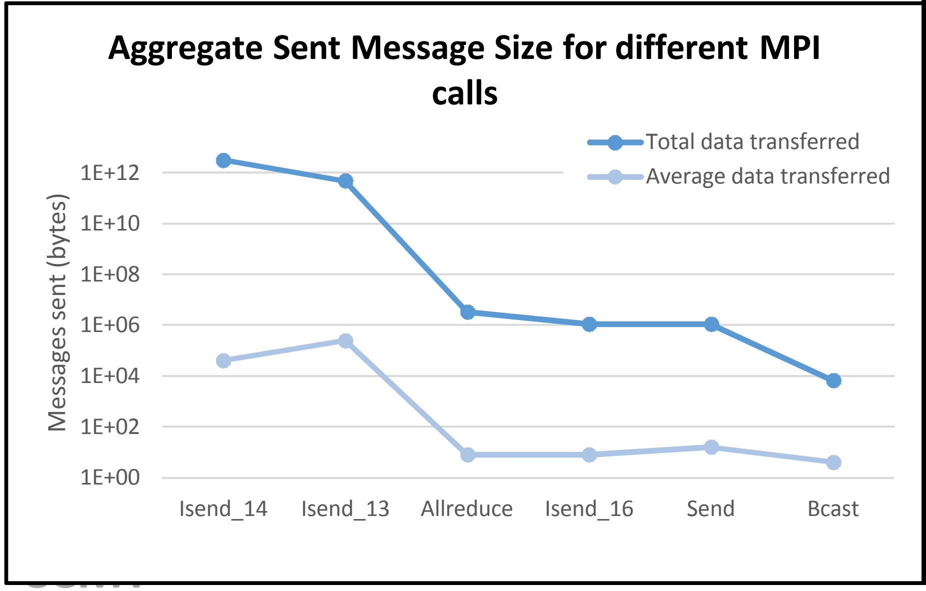
- Polynomial degree of Nx=Ny=Nz=N
- Total no. of elements, E
- No. of MPI ranks, P
- Physical quantities, Q = 5
- No. of bytes, B

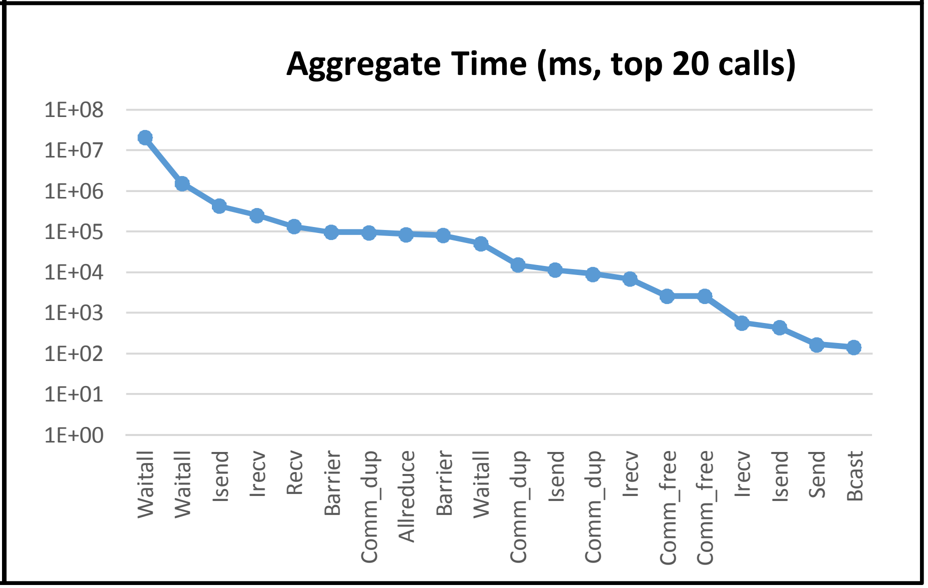
CMT-Bone MPI Profiling Data

- Experimental setup:
  - 128 MPI ranks, 1 rank/node
  - mpiP profiling data
  - Best-case, all exchanges across all MPI ranks occur in parallel

These experiments were run on Intel Sandy Bridge based ASC testbed at Sandia National Laboratories, Albuquerque, NM.



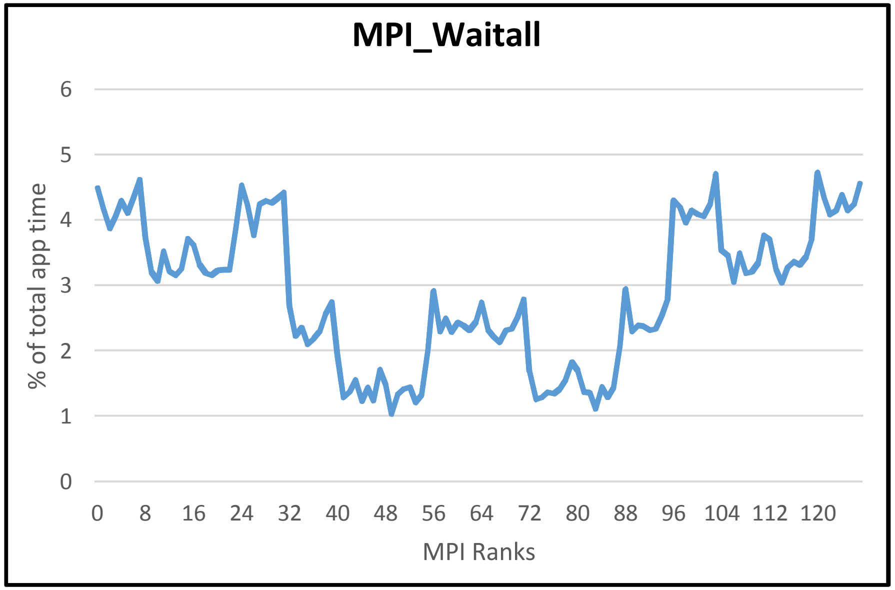


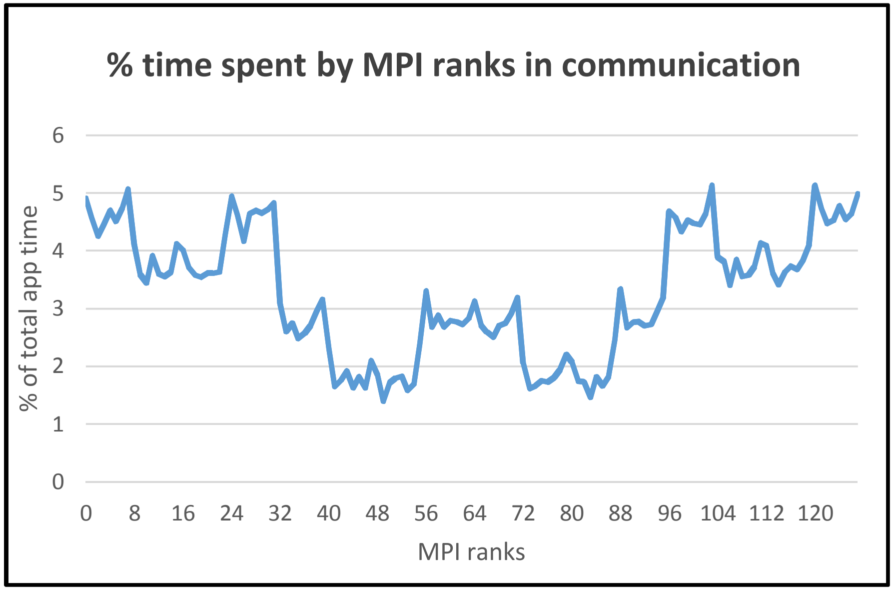


Data for Estimation of Transfer Times

These experiments were run on Intel Sandy Bridge based ASC testbed at Sandia National Laboratories, Albuquerque, NM.

Overall Communication Time Estimation





These experiments were run on Intel Sandy Bridge based ASC testbed at Sandia National Laboratories, Albuquerque, NM.

- Most of the time is spent in MPI_Waitall
  - Need timed simulations to look at these effects
  - It may still be possible to use coarse models for actual transfer time estimations

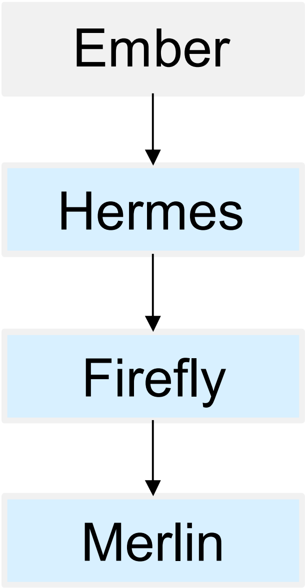
Sandia National Laboratories

CCMT

Scalable Network Simulation using SST

- Develop abstract end-point models '**motifs**' for various communication routines used in CMT-Nek
  - Identified routines: Nearest-neighbor communication using pairwise exchange, all-to-all using crystal routing, allreduce, bcast etc.

- Ember is an end-point model for network communications
  - Motifs are condensed, efficient models of communication which are able to correctly represent the target, size and data type of messages in larger applications, libraries and mini-apps
  - Events generated by motifs are interpreted by the Ember engine and then handed off to the Hermes middleware emulation layer
  - Hermes provides timing for basic middleware operations such as MPI message matching
  - Currently supports SHMEM/MPI-3 one-sided communications



Scaling & Speeding up SST Simulations

- Currently working on evaluating the sensitivity of simulations to different model parameters
  - Run simulations across a sweep of different parameters such as MPI match latency, packet size, buffer sizes etc.
  - Quantify the effect of these parameters on simulated time
- Final goal is to speedup the simulations by reducing
  - Number of components being simulated,
  - Number of parameters that are needed to describe a system, and
  - Number of events being generated by each component
- It has to be good enough to provide a first-order approximation of performance which can enable application developers to do some early design space exploration