Towards Performance and Scalability Analysis of Distributed Memory Programs on Large-Scale Clusters

Sourav Medya[1,2], Lucy Cherkasova[2], Guilherme Magalhaes[3], Kivanc Ozonat[2], Chaitra Padmanabha[3], Jiban Sarma[3], Imran Sheikh[3]

[1]University of California, Santa Barbara,[2]Hewlett Packard Labs, and [3]Hewlett Packard Enterprise

Motivation

– To improve completion time of many distributed memory programs (HPC and Big Data) they are executed on large-scale clusters.

– However, in the initial implementation phase, programmers are often bound to a limited size cluster

– **Challenges:** understand and assess scalability of the designed applications

- In a larger cluster, each node processes a smaller data portion
- However, increased communication volume might be detrimental for overall application performance

– **Goal:** Extrapolate performance of an application on a large system using measurements and data analysis on a small cluster

Use Case: Graph500 Benchmark

Graph500 is a new benchmark for measuring computer's performance in memory retrieval (introduced in 2010):

- It performs breadth-first searches (BFS) in undirected graphs;
- Find all the vertices "one-hop" away, "two-hops" away, etc.

– Two distinct kernels:

- **Kernel_1** (graph generator, arbitrarily large size): Kronecker Graph
- **Kernel_2** performs BFS from a randomly chosen vertex ***(timed)***

– The ranking is determined by:

- **Problem Scale** (defined by the Graph Size)
- Achieved throughput in **TEPS (Traversed Edges Per Second).**

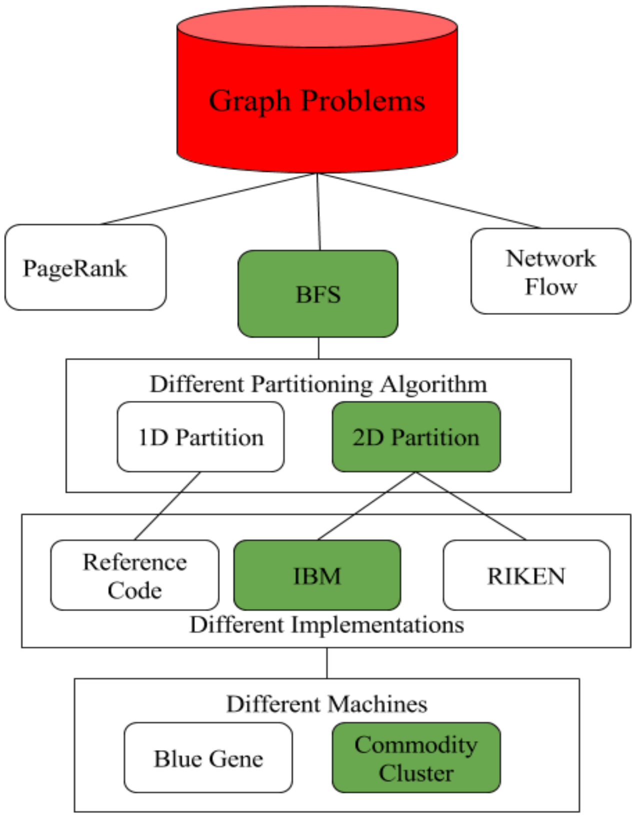
– Our Testbed:

32 nodes	28cores/node	256GB DRAM/node

Problem Definition

- Build a scalability model for application *Completion Time* as a function of "important" factors
- Assess the effect of available bandwidth on the increased communication volume in the increased size cluster

What Matters in Scalability Analysis?



Selected Algorithm

Implementation Code

Underlying System Hardware and Software

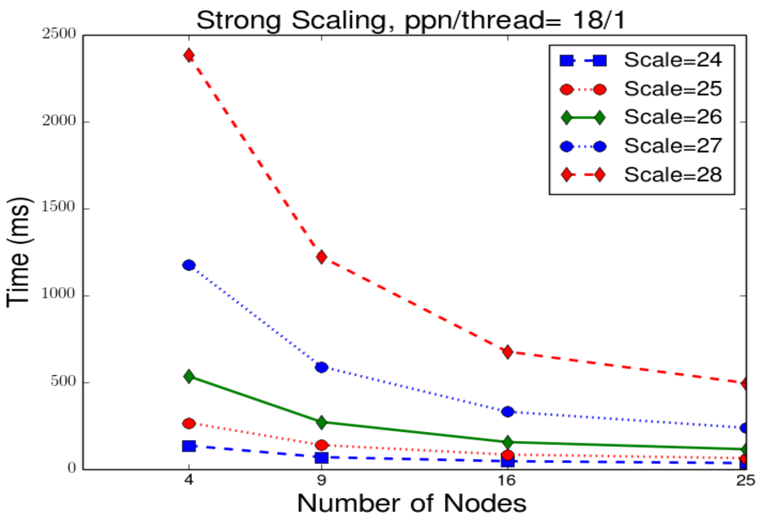
Hewlett Packard Enterprise

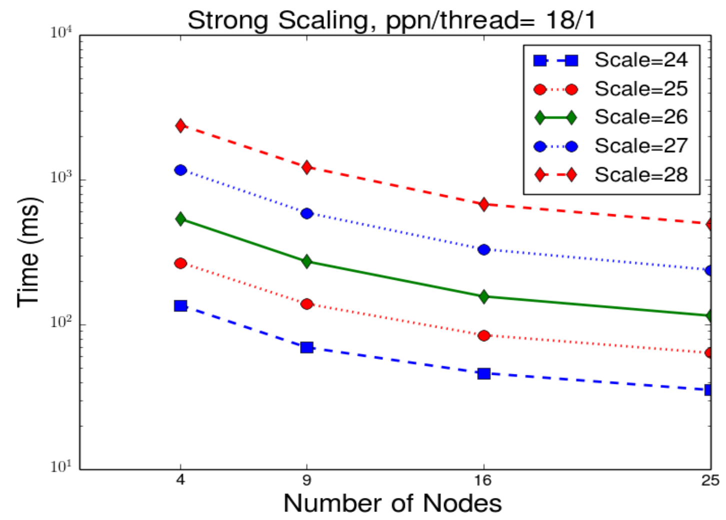
Scalability Analysis of Graph500

Strong scaling:

– Increased number of nodes in the cluster

– Fixed problem size





- Scale (s) denotes the size of the graph with 2^s nodes and 16.2^s edges
  - Graph of scale 27 has 134 Million vertices and 2.1 Billion edges;
  - Graph of scale 28 has 268 Million vertices and 4.2 Billion edges; etc.

- *Modeling: How to capture processing and communication time?*

Hewlett Packard Enterprise

Base Linear Regression Model

Simple Equation (p= number of processes/cores)

Completion Time = Processing Time + Communication Time

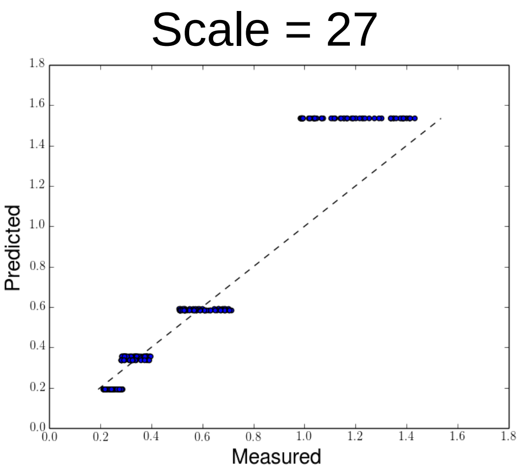
$$= O\left(\frac{1}{p}\right) + O\left(\frac{1}{\sqrt{p}}\right)$$

$$= c_1 * \frac{1}{p} + c_2 * \frac{1}{\sqrt{p}}$$

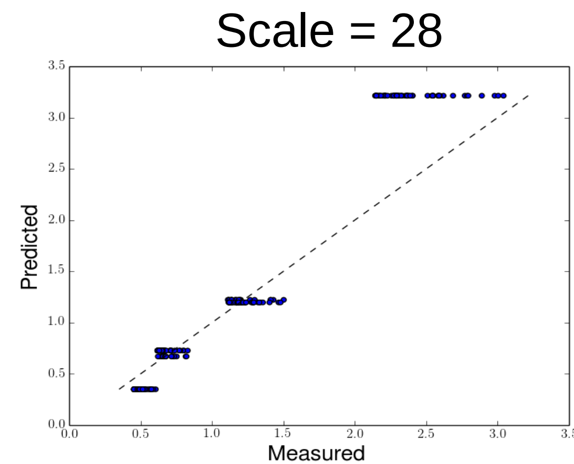
- Use properties of 2D-data partition algorithm (where, a number of messages per process is O($\sqrt{p}$))
- We apply linear regression to find c_1 and c_2 coefficients

Hewlett Packard Enterprise

Base Linear Regression Model



	Scale =27	Scale =28
R^2 error	0.95	0.97
MSE	0.04	0.19
c_1	57.67	102.1
c_2	3.8	10.3



Prediction of completion times (in seconds) along with Table of errors and coefficients

- ***Good accuracy*** – the model is a good fit to observed data
- High R^2 (close to 1 is better)
- Low MSE (close to 0 is better)
- *Processing* time dominates *Communication* time in a small cluster
- c_1 / c_2 decreases as data scale increases

Hewlett Packard Enterprise

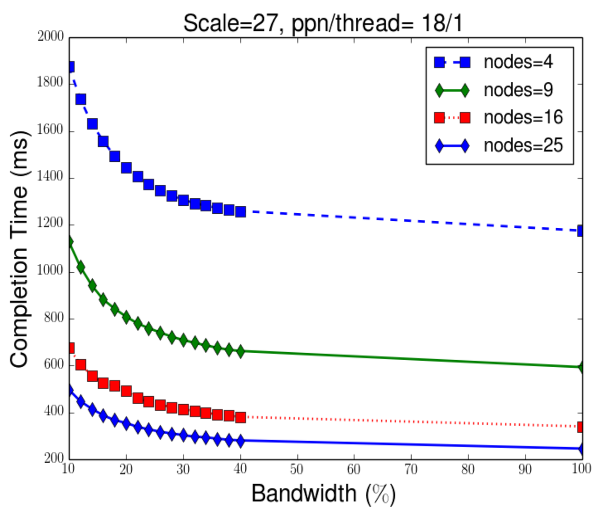
Effect of Interconnect Bandwidth

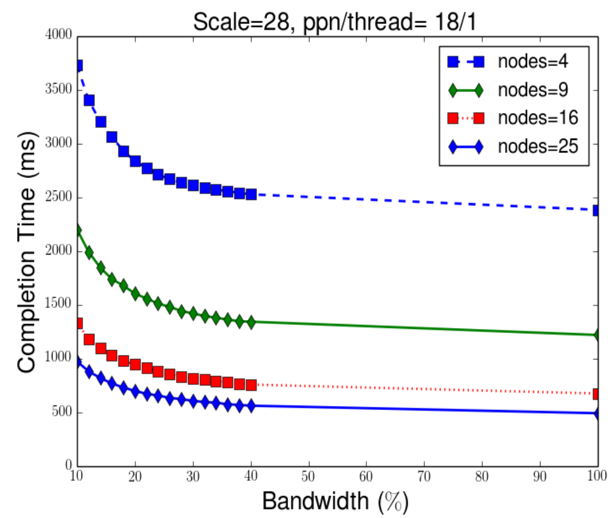
– Communication becomes a dominant component with an increased number of nodes

–Challenges of bandwidth measurements: a variety of MPI collectives and calls

– Our approach: apply a bandwidth throttling tool *InterSense* [1]

–It uses a message padding to reduce the effective bandwidth



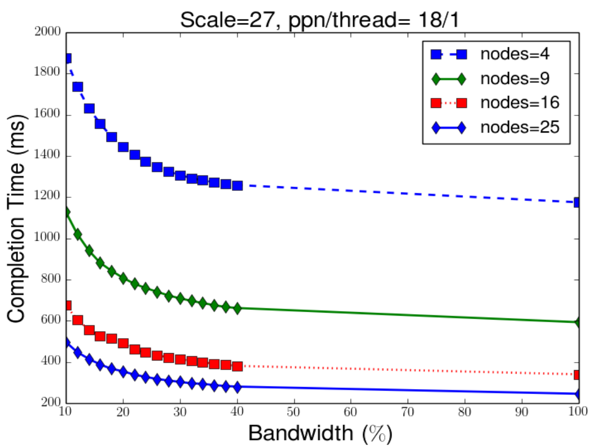


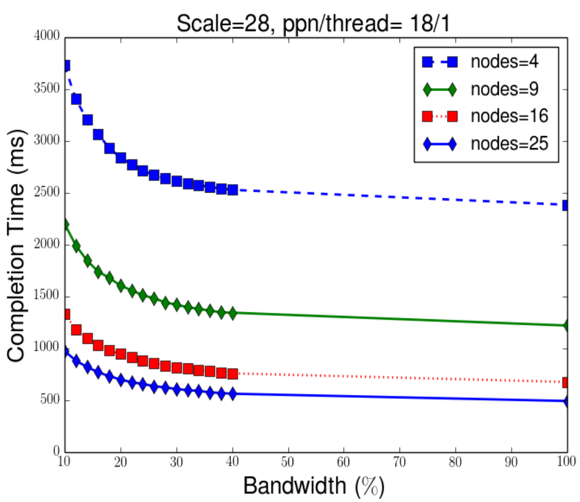
- **Need** to assess the increased bandwidth demands in the increased size cluster.

Hewlett Packard Enterprise

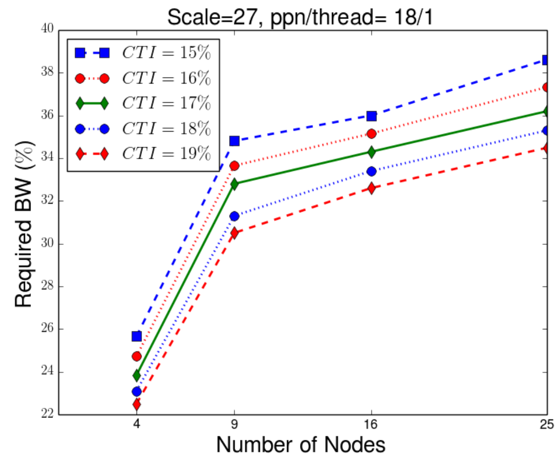
*[1] Q. Wang, L. Cherkasova, J. Li, and H. Volos. **InterSense:** Interconnect Performance Emulator for Future Scale-out Distributed Memory Applications. MASCOTS, 2015*

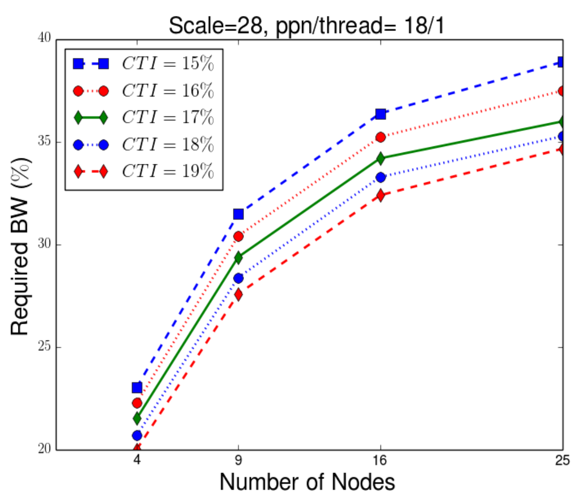
Summary



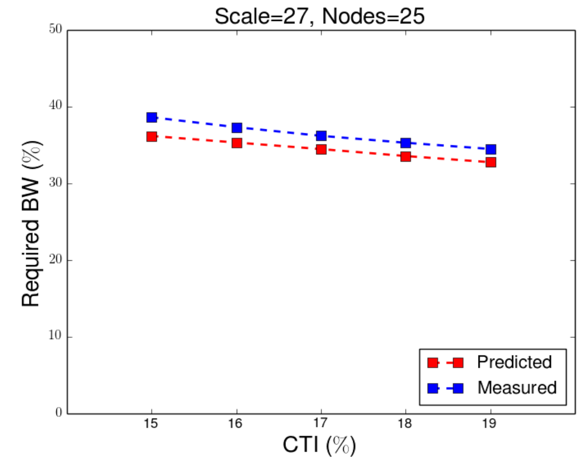


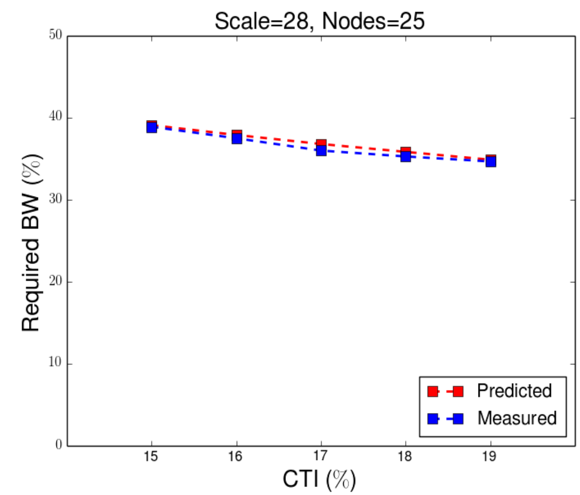
Effect





Modeling





Prediction

The **error is under 2%**

Hewlett Packard Enterprise

Conclusion

- Scalability analysis of a distributed memory program is challenging
- Two critical factors: communication volume and the available interconnect bandwidth
- We propose a novel approach for estimating the required interconnect bandwidth in a **larger cluster** using the experiments in a **small/medium** cluster performed with **"bandwidth throttling tool"**

Future Work:

- Build a general model as a function of data scale, number of nodes, and available bandwidth
- Produce a performance curve for CT and TEPS metrics:
  - Add a "correction" factor reflecting the impact of available bandwidth on a completion time
  - Estimate a cluster size, where a communication cost becomes a dominant component and cripples scalability benefits

Hewlett Packard Enterprise

Thank you!

Questions?

Hewlett Packard Enterprise