Graph500 in the public cloud

Master project Systems and Network Engineering

Harm Dermois

Supervisor: Ana Lucia Varbanescu

What is Graph 500

- List of the best top 500 best graph processing machines
- Benchmark tailored to graph processing
- Other metrics

What is Graph 500

The Graph 500 List

November 2014

No.	Rank ▲	Machine	Installation Site	Number of nodes	Number of cores	Problem scale	GTEPS
1	1	DOE/NNSA/LLNL Sequoia (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)	Lawrence Livermore National Laboratory	98304	1572864	41	23751
2	2	K computer (Fujitsu - Custom supercomputer)	RIKEN Advanced Institute for Computational Science (AICS)	82944	663552	40	19585.2

What is Graph 500

93	93	DAS-4/VU (SuperMicro)	VU University	128	0	30	7.0867
172	172	Okorok (Dell - PowerEdge C1100)	Home	1	8	17	0.228
182	182	Scott Beamer's iPad (Apple - iPad 3)	UC Berkeley	1	2	14	0.0304

Getting on the list

Input : scale and edge factor

Create edge list

Make graph (timed)

For **64** random search **keys** do:

    Breadth First Search (timed)

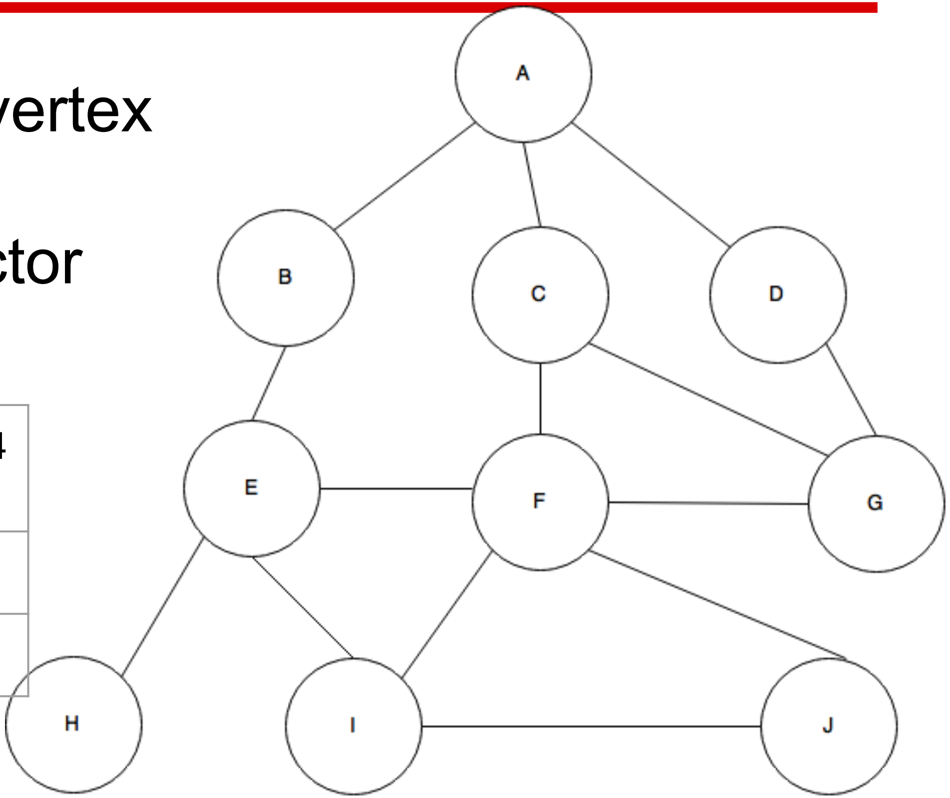
    Validate (Skipped)

Report time

Edge list generation

- Tuple of start vertex to end vertex and a label
- Uses the scale and edge factor
- Randomize edge list

1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	A	A	B	C	C	D	E	E	E	F	F	F	I
B	C	D	E	F	G	G	H	F	I	I	J	G	K

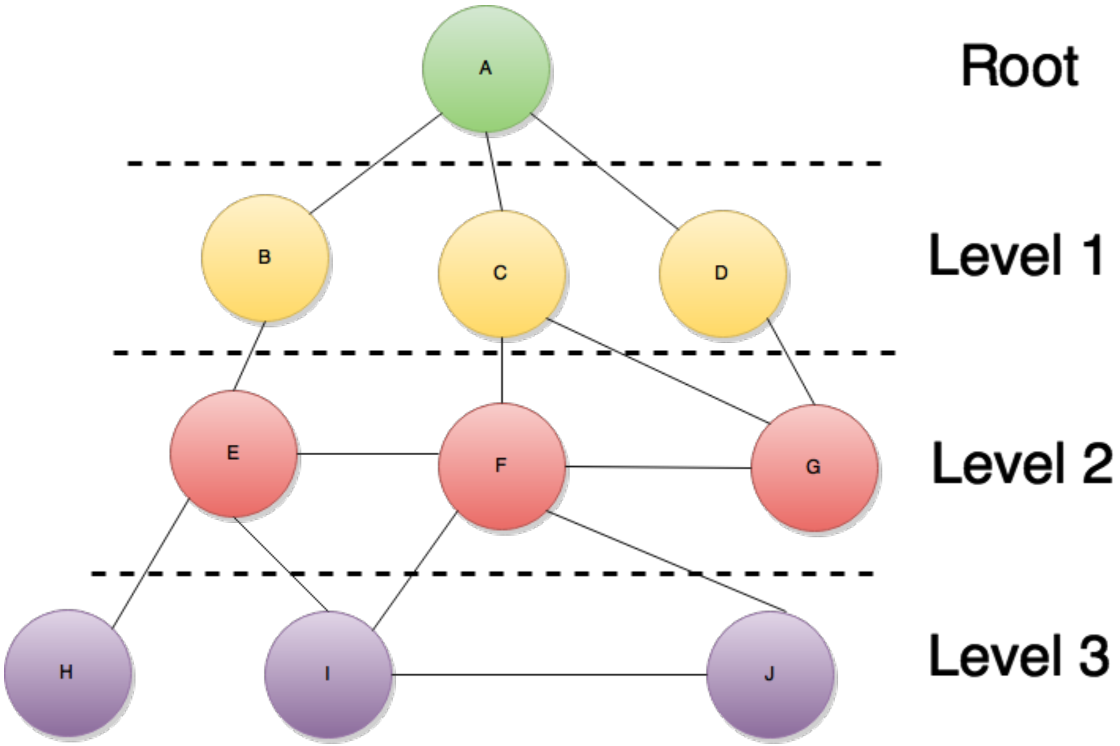


Graph construction

- Change edge list to other data structure with more locality
- Compressed Row Storage

Edge label	1	2	3	4	5	6	7	8
col_index	2	3	4	1	5	1	6	7
row_pointer	1	4	6	9				

Breadth First Search



Why run Graph 500 on the cloud?

How good is the cloud at graph processing?

- Advantage:
  - No need to own equipment.
  - Elastic for larger and larger graphs.
- Disadvantage:
  - Performance might be really bad …

… and it is cool to have your name in the list!

Research questions

Is it possible to **model** the **performance** of the Graph500 benchmark on a **public cloud** as a function of the used resources?

- What is the performance?
- What scale fits?
- What is the model?

Methodology & Scope

One implementation: *graph500_mpi_simple*

Hardware:

- *DAS-4 (With and without InfiniBand)*
- *OpenNebula (On the DAS-4)*
- *Amazon Webservices EC2*

Metric: *TEPS*

- BFS performance = number of traversed edges per second (TEPS)

Hardware specifications

Where	# Nodes	Processor	CPUs	RAM	Price
DAS-4 VU	46(all)	2.40GHz	2 * 8	24 GB	
DAS-4 LU	16	2.40GHz	2 * 8	48 GB	
OpenNebula	8	2.00 GHz	24 (8 VCPU)	66 GB	
c3.large	“Unlimited”	2.80GHz	2 VCPU	4 GB	$0.105 per Hour
r3.large	“Unlimited”	2.40GHz	2 VCPU	16 GB	$0.175 per Hour

graph500_mpi_simple

Distributes the vertices evenly over the nodes

Works top-down, per level

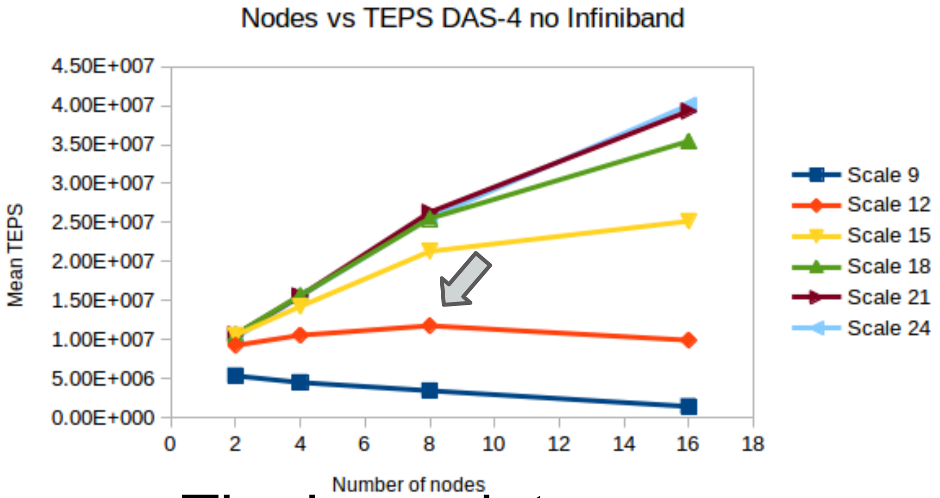
Each level => task queue

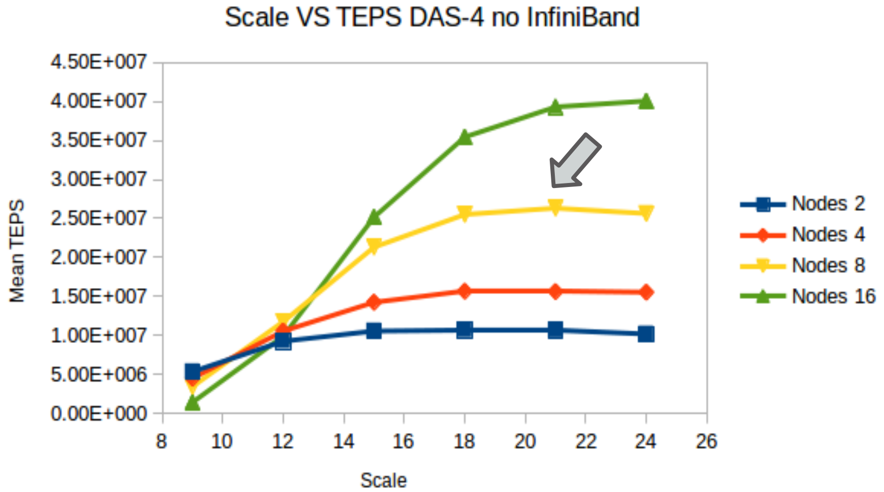
Uses Non blocking communication

Limitations

- Needs the number of nodes to be a power of 2
- Uses only 1 CPU for BFS

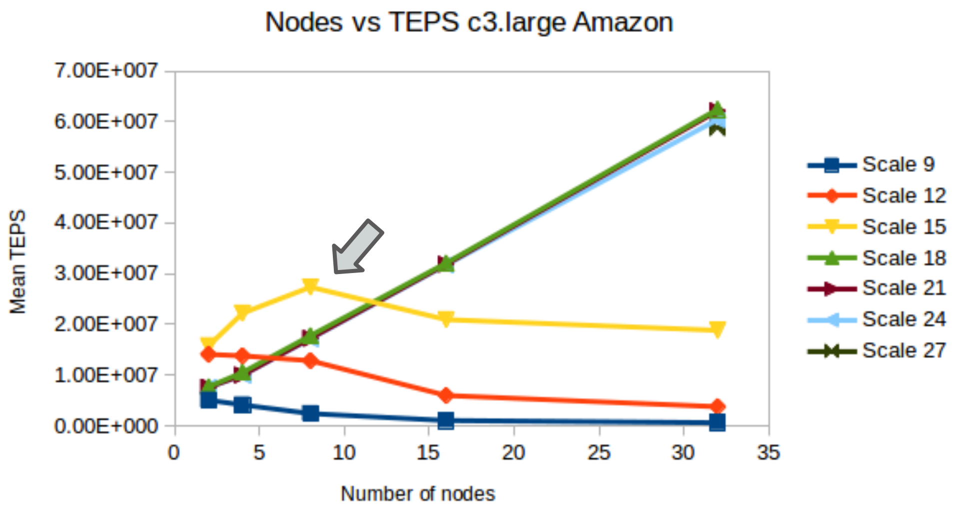
Results DAS-4 no InfiniBand

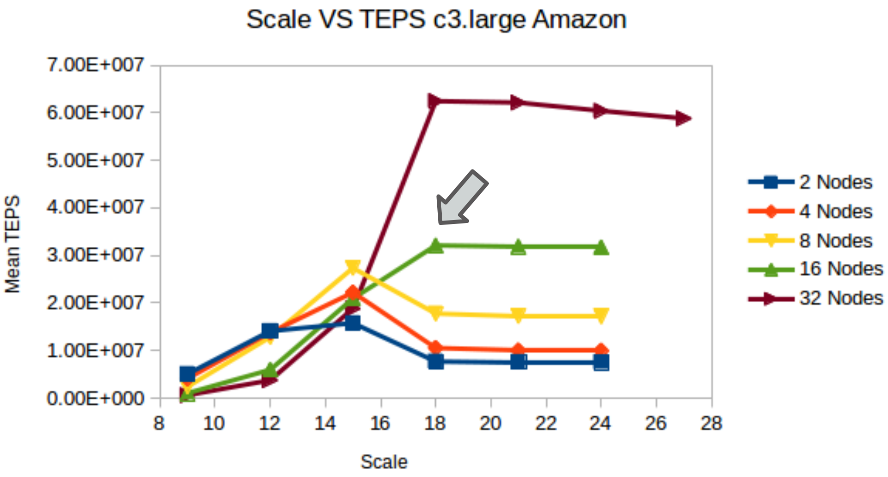




- Tipping points
- More nodes => more TEPS for scales 15 and larger
- TEPS is a linear function of the number of nodes

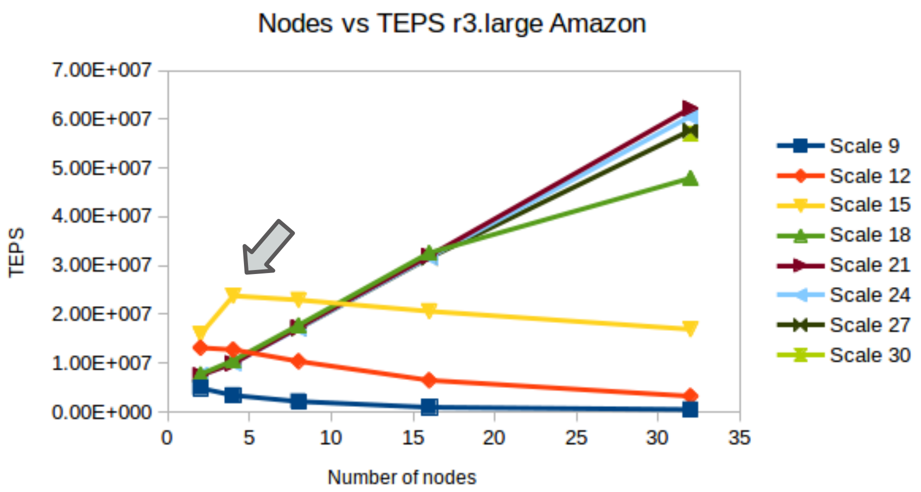
Results Amazon c3.large

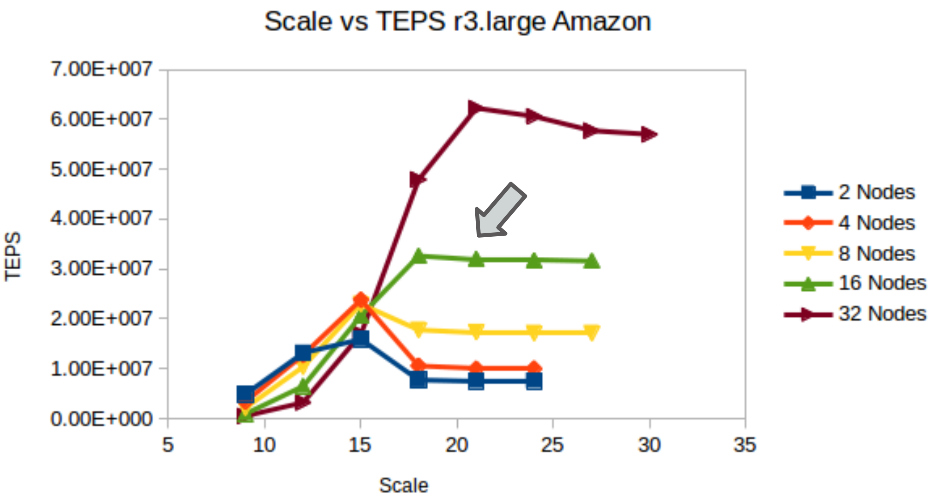




- Same behavior as DAS-4 no InfiniBand at higher scales.
- Scale 15 and lower a different behavior
- Even less of a decline than the DAS-4 at higher scale.

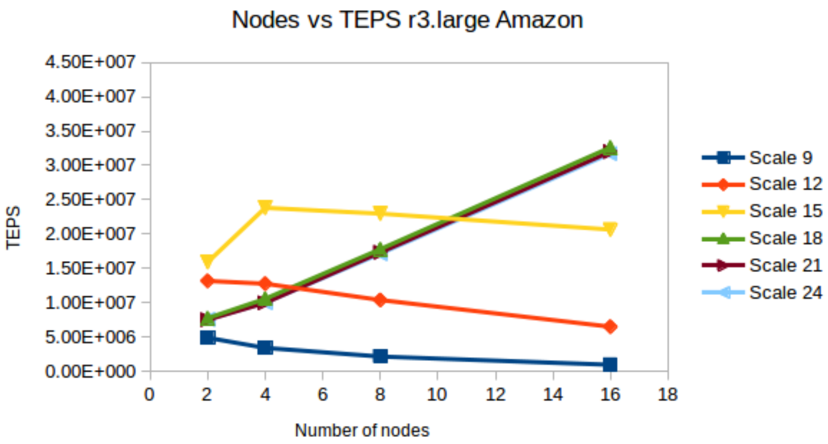
Results Amazon r3.large

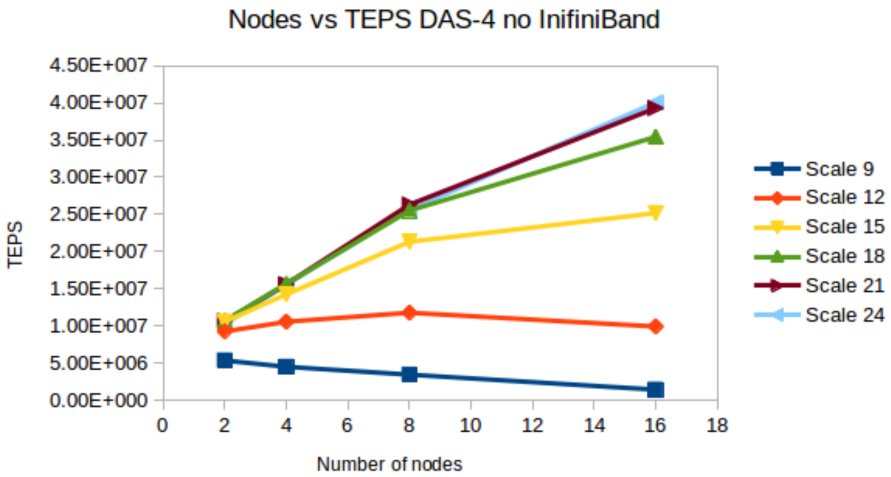




- Results almost identical to the c3.large
- Can handle larger scales because it has more **RAM**

Comparison Amazon and DAS-4





- **10%-50%** difference for large **scale** and number of **nodes**

Research questions

Is it possible to **model** the **performance** of the Graph500 benchmark on a **public cloud** as a function of the used resources?

- What is the model?
- What is the performance?
- What scale fits?

Conclusion

A model can be made:

$$\text{TEPS(scale)} = \begin{cases} a*\#\text{nodes}+b, & \#\text{nodes} <= T \\ \text{slow decrease}, & \#\text{nodes} > T \end{cases}$$

where Tipping point = T = f(scale, architecture)

a,b=f(scale?, architecture)

- Scale 30 is doable with 32 nodes r3.large
- Overall competitive, performance-wise, with the ranks 5-10 supercomputers.

Future work

- More nodes and larger scales.
- Multiple processes per node.
- Different cloud instances.
- Optimizations.

Prediction*

# Nodes	2048	8192	2097152
GTEPS	1.9891	7.9565	2036.8654
Cost per hour	$245.76	$983.04	$251,316.48

With 8192 nodes => above the DAS-4.
With 2097152 nodes => 6th place can be achieved

**Disclaimer: this is just a prediction*

Questions?

Hypothesis

Performance =

max(CPU Time, Comm time) / Traversed edges

- CPU time => function of number of nodes
- Comm time => function of scale, number of nodes, and message buffering

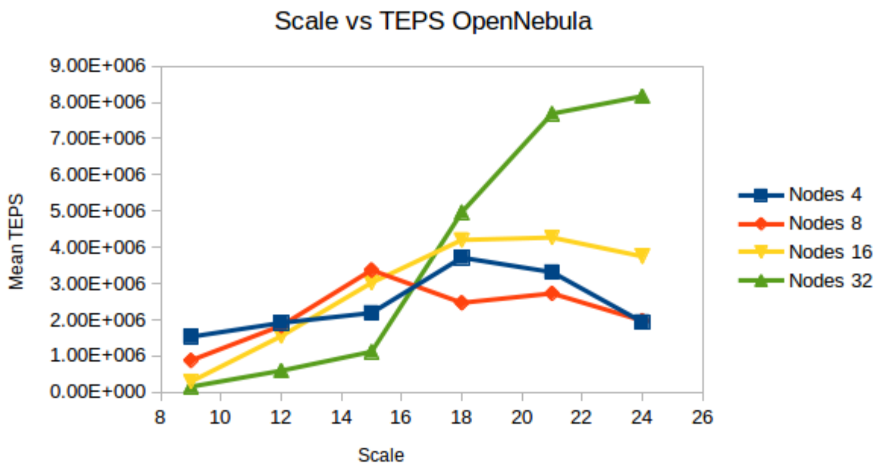
Technical difficulties

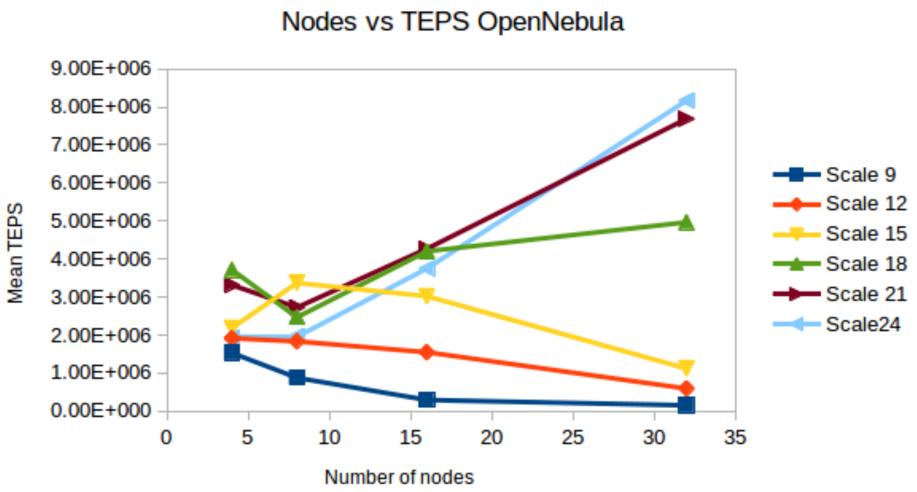
Does not work properly with MPI 1.4

OpenNebula cloud shutdown the day I started

On demand instances limit

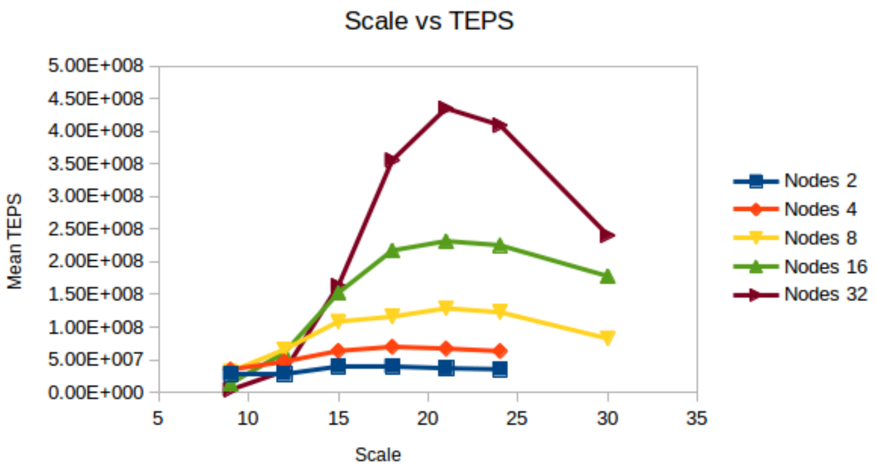
Results OpenNebula

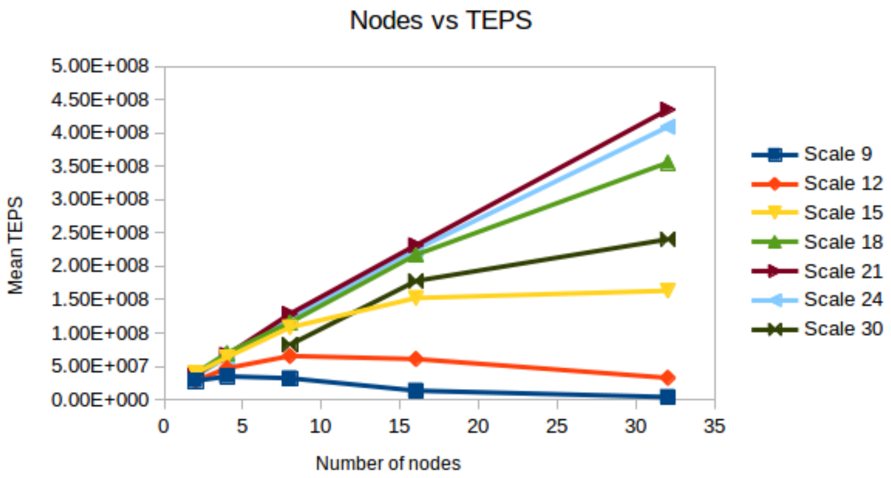




- Lines cross more often.
- 8 times **less** TEPS compared to InfiniBand.

Results DAS-4 with InfiniBand





- After tipping point a harsh decline.
- Scales above 15 double in TEPS as Nodes double.

Intel MPI Benchmark

Size (bytes)	DAS-4 µsec	DAS-4 InfiniBand µsec	OpenNebula µsec	Amazon µsec
0	3.81	46.55	112.75	81.82
1024	4.93	56.97	130.76	91.40
2048	5.96	68.36	269.74	102.96

Output

```
SCALE:                        21
edgefactor:                   16
NBFS:                         64
graph_generation:             3.52925
num_mpi_processes:            8
construction_time:            2.32684
min_time:                     1.63176
firstquartile_time:           1.87955
median_time:                  1.96341
thirdquartile_time:           2.01269
max_time:                     2.1227
mean_time:                    1.94503
stddev_time:                  0.097596
min_nedge:                    33554432
firstquartile_nedge:          33554432
median_nedge:                 33554432
thirdquartile_nedge:          33554432
max_nedge:                    33554432
mean_nedge:                   33554432
stddev_nedge:                 0
min_TEPS:                     1.58074e+07
firstquartile_TEPS:           1.66714e+07
median_TEPS:                  1.70898e+07
thirdquartile_TEPS:           1.78524e+07
max_TEPS:                     2.05633e+07
harmonic_mean_TEPS:           1.72513e+07
harmonic_stddev_TEPS:         109058
Program time:                  130.366
```

Related work

Suzumura, Toyotaro, et al. "Performance characteristics of Graph500 on large-scale distributed environment." *Workload Characterization (IISWC), 2011 IEEE International Symposium on*. IEEE, 2011.

Angel, Jordan B., et al. *Graph 500 performance on a distributed-memory cluster*. Tech. Rep. HPCF–2012–11, UMBC High Performance Computing Facility, University of Maryland, Baltimore County, 2012.

Edge list and graph creation

	A	B	C	D	E	F	G	H	I	J
A	0	1	1	1	0	0	0	0	0	0
B	1	0	0	0	1	0	0	0	0	0
C	1	0	0	0	0	1	1	0	0	0
D	1	0	1	1	0	1	0	0	0	0
E	0	1	0	0	0	1	0	1	1	0
F	0	0	1	0	1	0	1	0	1	1
G	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	1	0	0	0	0	0
I	0	0	0	0	1	1	0	0	0	1
J	0	0	0	0	0	1	0	0	1	0

# of non zeros	1	2	3	4	5	6	7	8
col_index	2	3	4	1	5	1	6	7
row_pointer	1	4	6	9				

Future work

- More nodes and larger scales.
- Multiple processes per node.
- Further investigate effect of the network on the performance for the DAS-4.
- Different cloud instances.
- Optimizations.