Improving Resources Usage in HPC Clouds

I. Petrov [1, a], A. Chupakhin[1, b], V. Antonenko [1, c], R. Smeliansky [1, d]

[1] *Lomonosov Moscow State University*

E-mail: [a] ipetrov@cs.msu.ru, [b] andrewchup@lvk.cs.msu.su, [c] anvial@lvk.cs.msu.su, [d] smel@cs.msu.ru

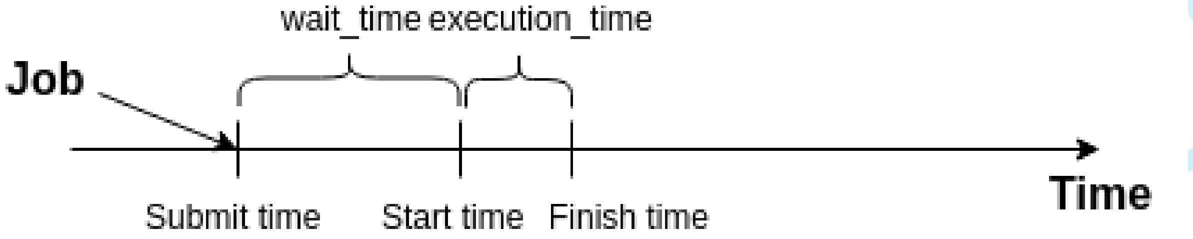
This work is supported by Russian Ministry of Science and Higher Education, grant #05.613.21.0088, unique ID RFMEFI61318X0088

NEC'2019

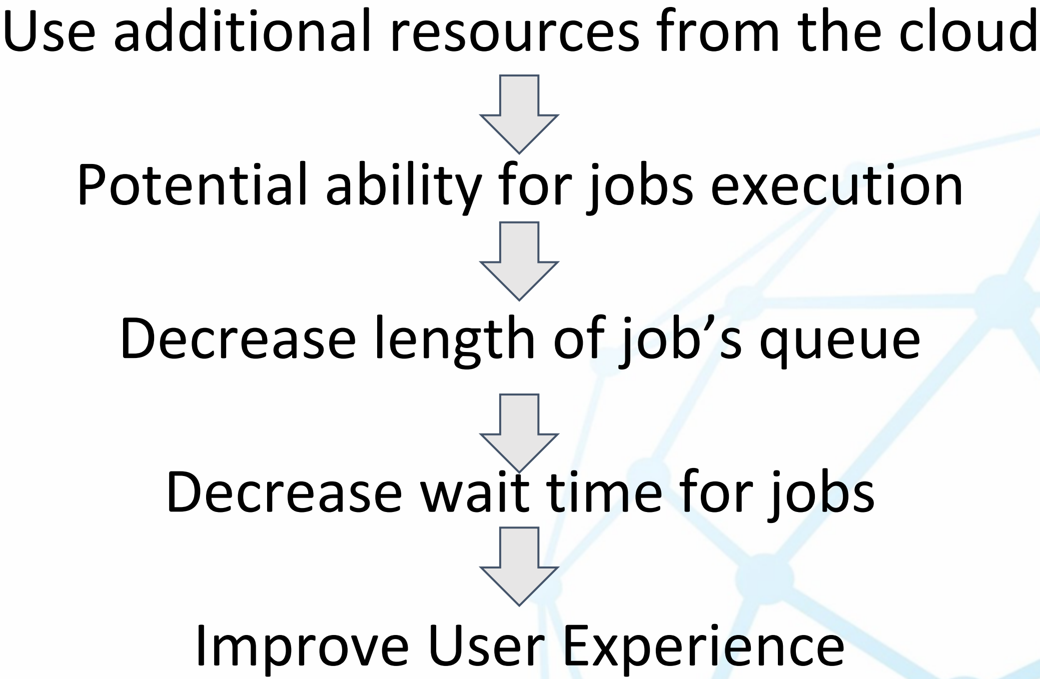
Problem Description

Current situation with HPC resources:

- Low User Experience for supercomputer users: problem with big (wait_time + execution_time)
- Supercomputer scheduler considers computing unit (not separated cores)
- Resources fragmentation => resources underutilization



Possible solution



Use additional resources from the cloud

Potential ability for jobs execution

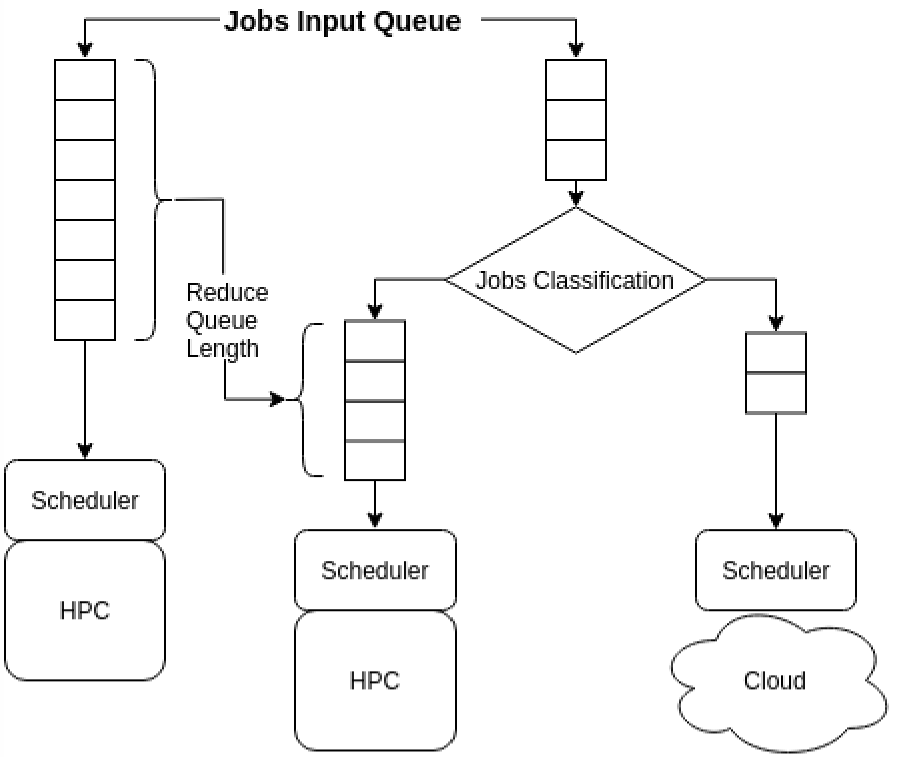
Decrease length of job’s queue

Decrease wait time for jobs

Improve User Experience

HPC or Cloud

Our goal -> reduce (wait_time + execution_time)



Our Hypothesis

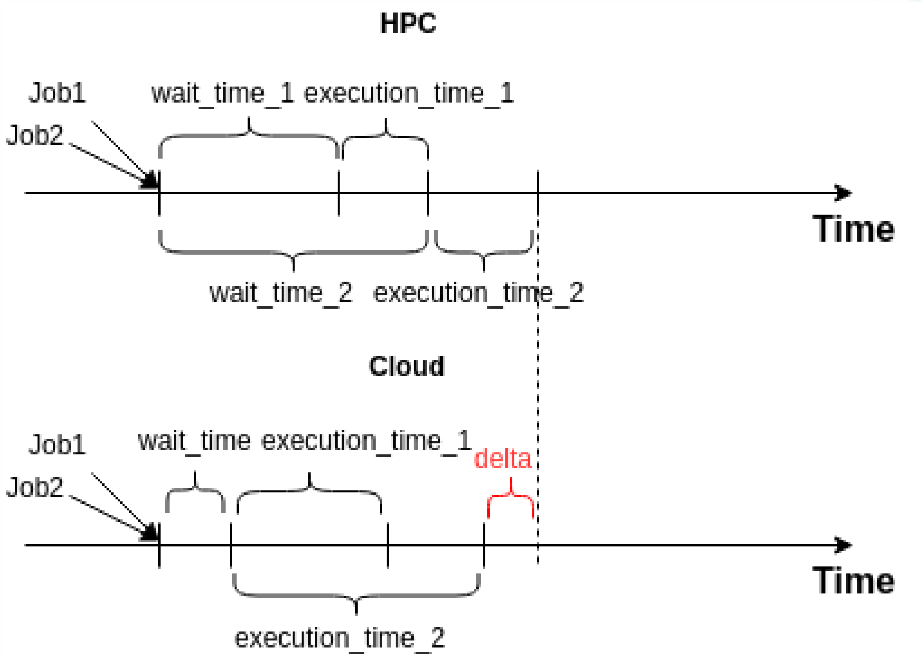
“MPI programs that don’t require a lot of computing resources can effectively share the same set of resources”

Details:

- We considered MPI programs in the cloud
- We want find MPI programs don’t require a lot of computational resources:
  - According to MPI program nature
  - MPI programs wait for data transmission due to the slow network

Our Hypothesis

“MPI programs that do not require a lot of computing resources can effectively share the same set of resources”



MPI programs

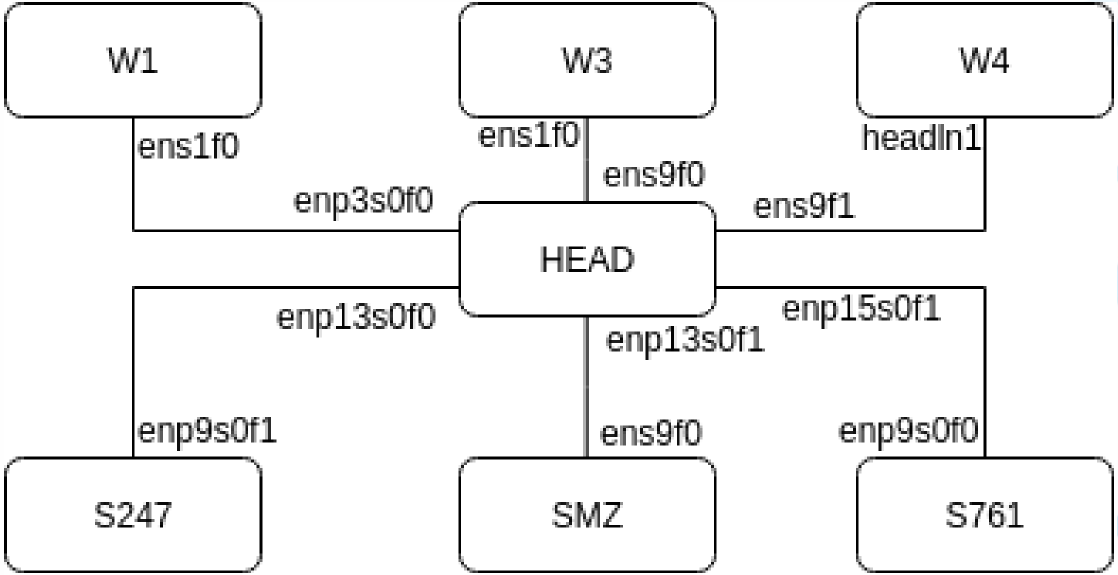
NASA Parallel benchmark:

- CG - Conjugate Gradient
- EP - Embarrassingly Parallel
- FT - Discrete 3D fast Fourier Transform
- IS - Integer Sort
- LU - Lower-Upper Gauss-Seidel solver

NPB has different sizes. We check S, A, B, C, D

Experimental stand

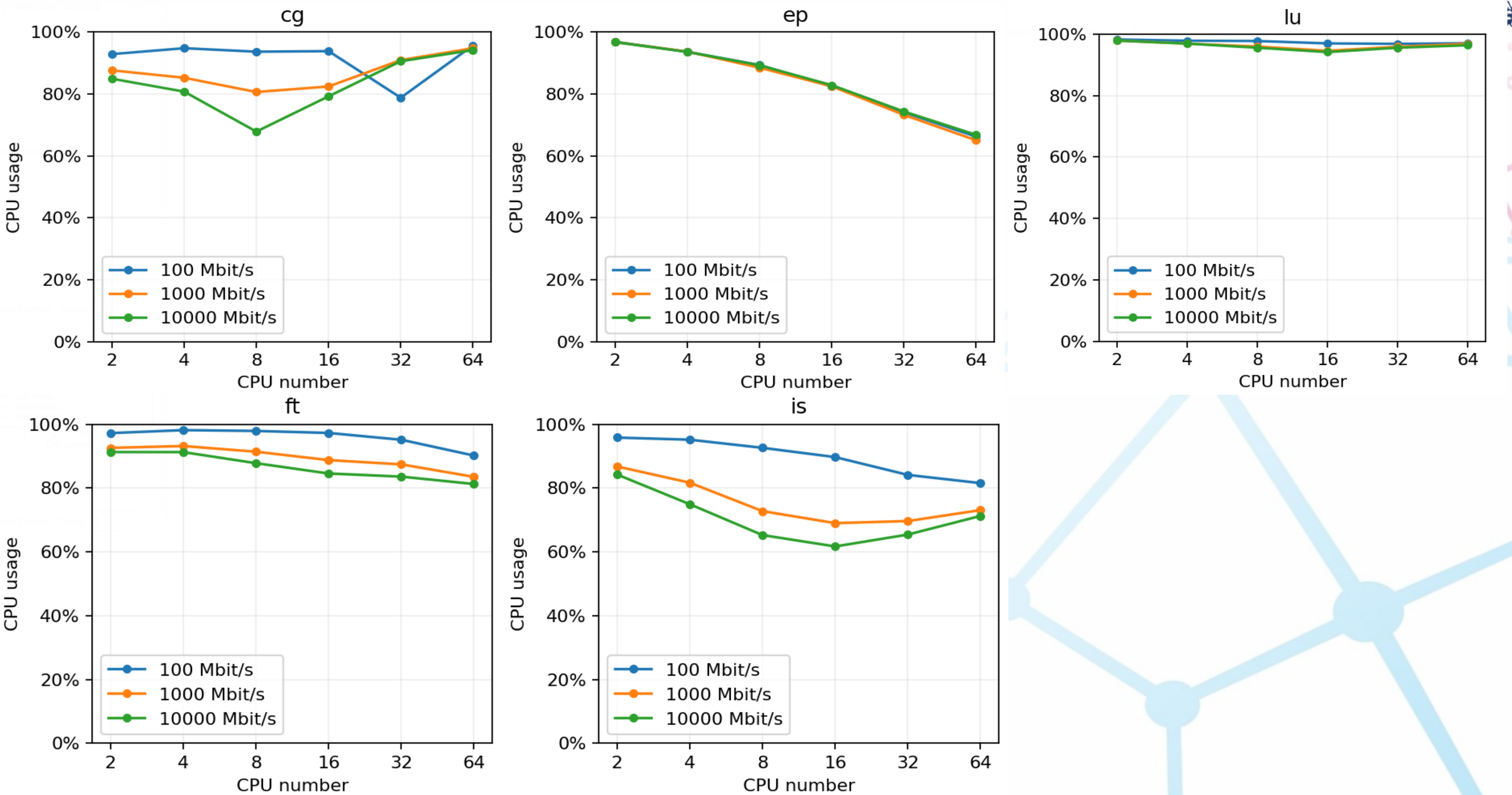
7 servers, 64 virtual machines, optical fibers between servers



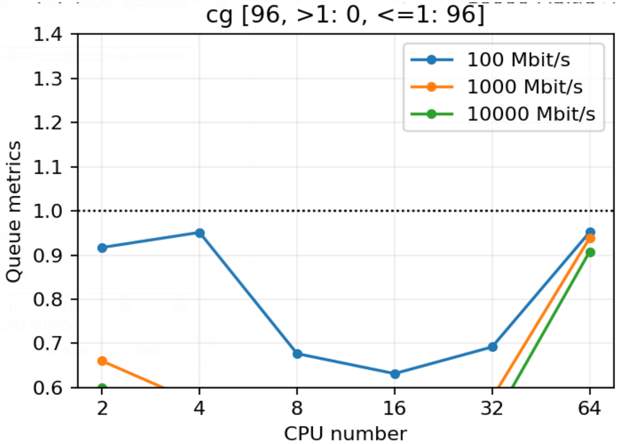
Measuring MPI program parameters

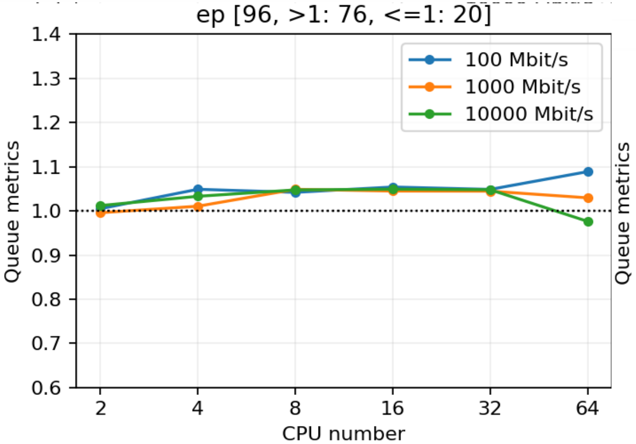
- CPU
  - Perf utility
- Network
  - /sys/class/net/<iface_name>/statistics/{rx_packets, tx_packets,rx_bytes, tx_bytes}
- Bandwidth, delay
  - traffic control utility

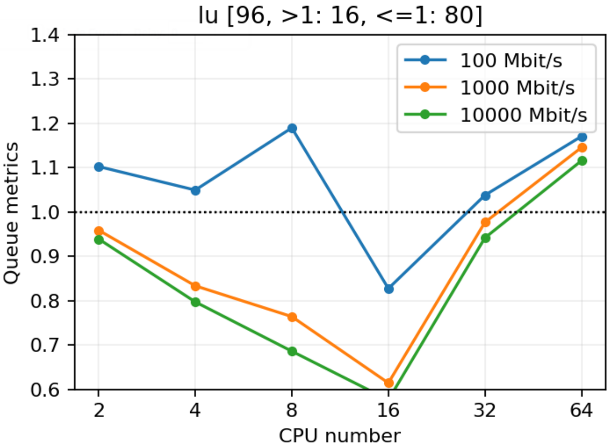
Experiment. NPB. CPU Usage

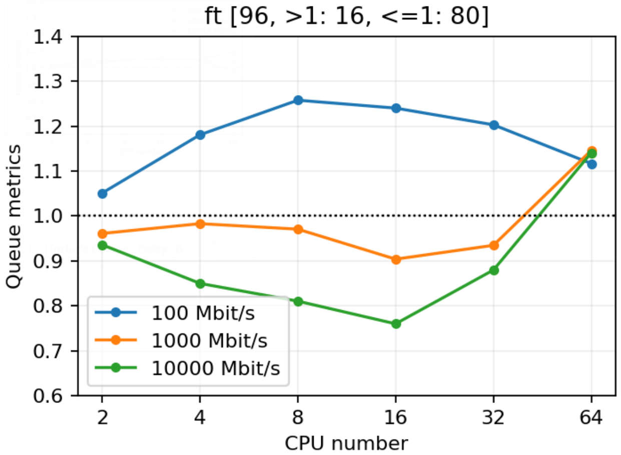


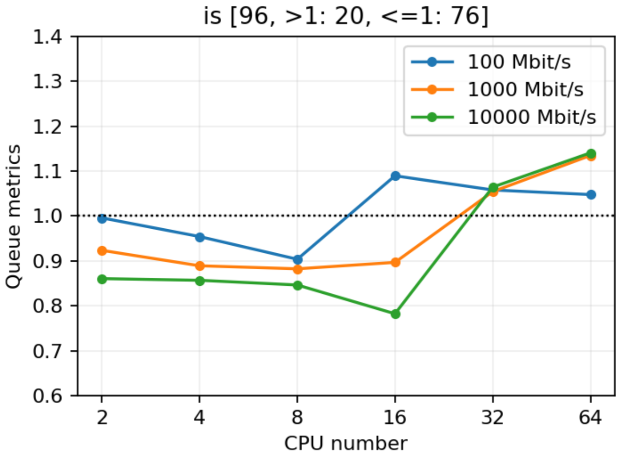
Experiment. NPB. Sharing


cg [96, >1: 0, <=1: 96]


ep [96, >1: 76, <=1: 20]


lu [96, >1: 16, <=1: 80]


ft [96, >1: 16, <=1: 80]


is [96, >1: 20, <=1: 76]

$$Queue\_metric = \frac{T^1_{pure} + T^2_{pure}}{max(T^1_{sharing}, T^2_{sharing})}$$

Queue metric is speed-up coefficient

Conclusion and further works

Conclusions:

- Experiments have shown, that you can do resources sharing in the cloud with slow network, but not for all programs. Our next goal is to write a scheduler that can do this.
- We need suitable criteria for evaluation of sharing opportunity

Future research:

- Develop scheduler for the cloud which can share resources
- MPI program execution time prediction
  - Extrapolation time for the same job
  - Time prediction using supercomputer log file