Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

Benchmarking of public and local cloud resources

Davide Grassano

Final Term Review

22 September 2015

Supervisor

Gabriele Garzoglio

Outline

- Introduction
  - Cloud computing
  - Benchmarks
- Benchmarks used
- Automation of the process
- Obtained results
- Conclusions

What is Cloud computing?

- A cloud is a collection of services and infrastructure accessible from anywhere in the network
- It improves on the concept of Grid computing by adding an on-demand resource provisioning and a layer of virtualization
- Thanks to this, resources are not static, but dynamically reallocated, maximizing their efficiency.
- Allows the sharing of resources for the users and the achievement of an economy of scale for the host
- Allows for enterprises the on-demand expansion of their computing capabilities, with no upfront commitment of money

Models of cloud servicies

- Software as a service (SaaS)
  - Allows user to gain access to software and databases on a pay-per-use basis
- Platform as a service (PaaS)
  - Computing platform are provided to the user, allowing for more simple and cost effective development of software and applications
- Infrastructure as a service (IaaS)
  - Provides the user with access to virtualized or bare metal machine within the cloud
  - Can be used to expand current computing power through Cloud bursting

The cloud project

- Aims to set up a hybrid IaaS cloud
- Allows a dynamic and more flexible allocation of resources on a on demand basis without requiring the intervention of system administrator
- A local cloud can be used to share resources not in use with partner institutions
- By switching to a hybrid cloud, extra computing power can be harnessed when needed through cloud bursting, from platforms like AWS

What is benchmarking?

- A series of tests to assess the capabilities of the hardware and software of the machine of interest
- Gives a metric to compare the performance of machines with different architecture
- Both standard benchmarks and specific ones (that mimics part of the workload of a full job) can be used
- A relative value of benchmarks can be obtained by comparing a known system such as local machines, with an unknown one such as AWS

Why is it needed?

- Help to determine which machines are most suited to run your jobs on
- Allows the optimization of the allocation of resources
- Gives an estimation of how long the job will need to run
- Fundamental for determining a system specs will satisfy the requirement of the project

Benchmarks used

- $t\bar{t}$bar_gensim
- hepspec06
- Bandwidth throughput tests
  - Amazon S3 up/download
  - FermiGrid
  - cmseos

Benchmarks used

- $t\bar{t}$bar_gensim
  - Simulates the generation of a small (150) number of ttbar events.
  - Give metrics specific for a full CMS job based on the ttbar/s result
- hepspec06
  - Smaller package of the more notorious collection of benchmarks SPEC2006
  - Stress the CPU and compiler of the system
  - Gives a more standard metric for comparison of resources

Benchmarks used

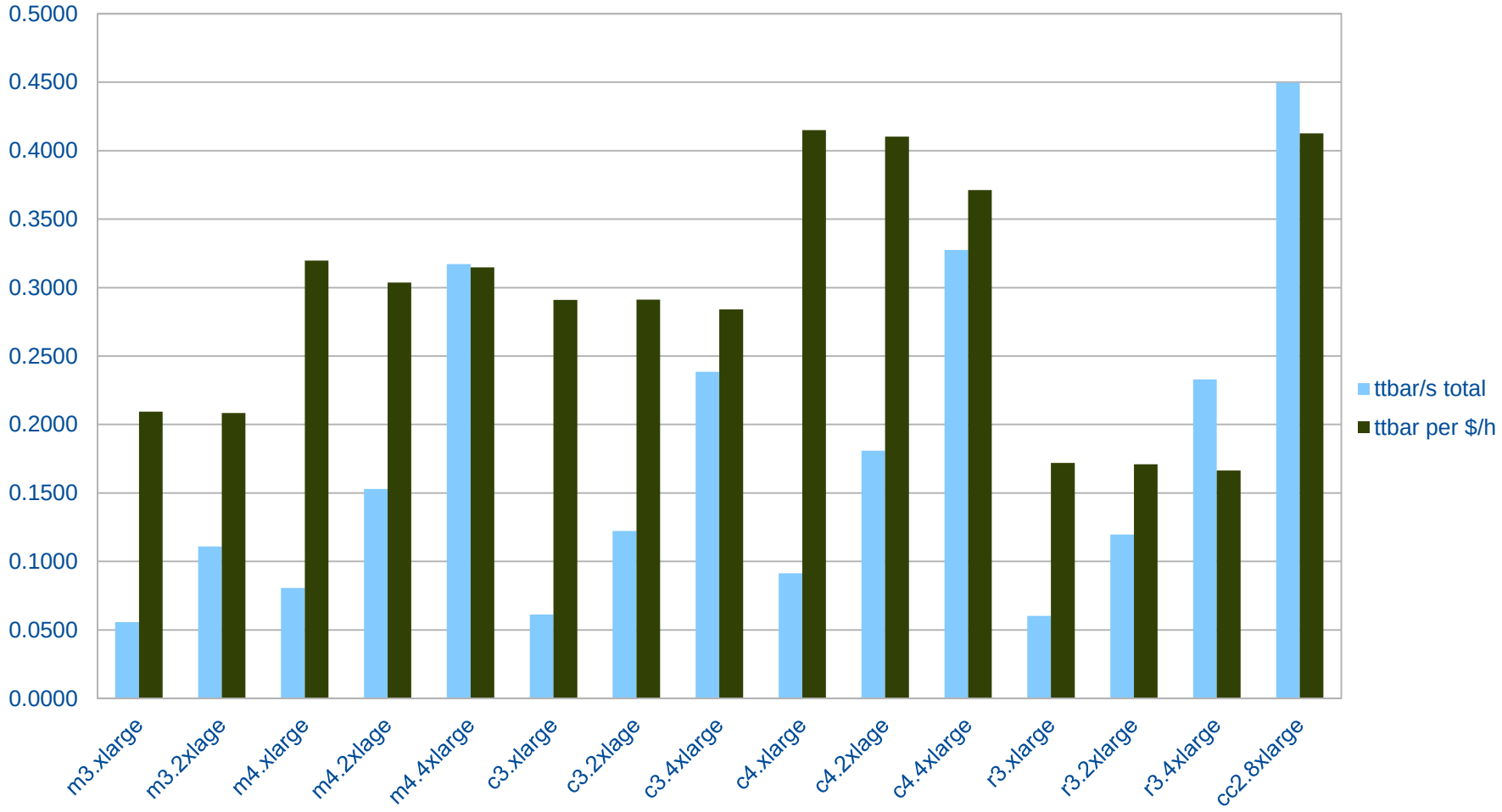
- Bandwidth throughput tests
  - Amazon S3: Simultaneous up/download of 1, 10 and 100 1GB files, from up to 25VMs contemporarily
  - FermiGrid: Simulation of a CMS job upload of data to the FermiGrid using parallel streams and multiple VMs contemporarily
  - Cmseos: Simulation of a CMS job upload of data to the FermiGrid using parallel streams and multiple VMs contemporarily (different dCache)
  - Parallel streams are required because of long latency between AWS and Fermilab

Automation of the process

- Script to automatically launch benchmarks on multiple machines:
  - aws_launch_benchmark.sh + instructions
- Script to automatically crop the results of the benchmarks:
  - crop_results.sh
- Script to automatically log in into aws machines:
  - aws_login.sh

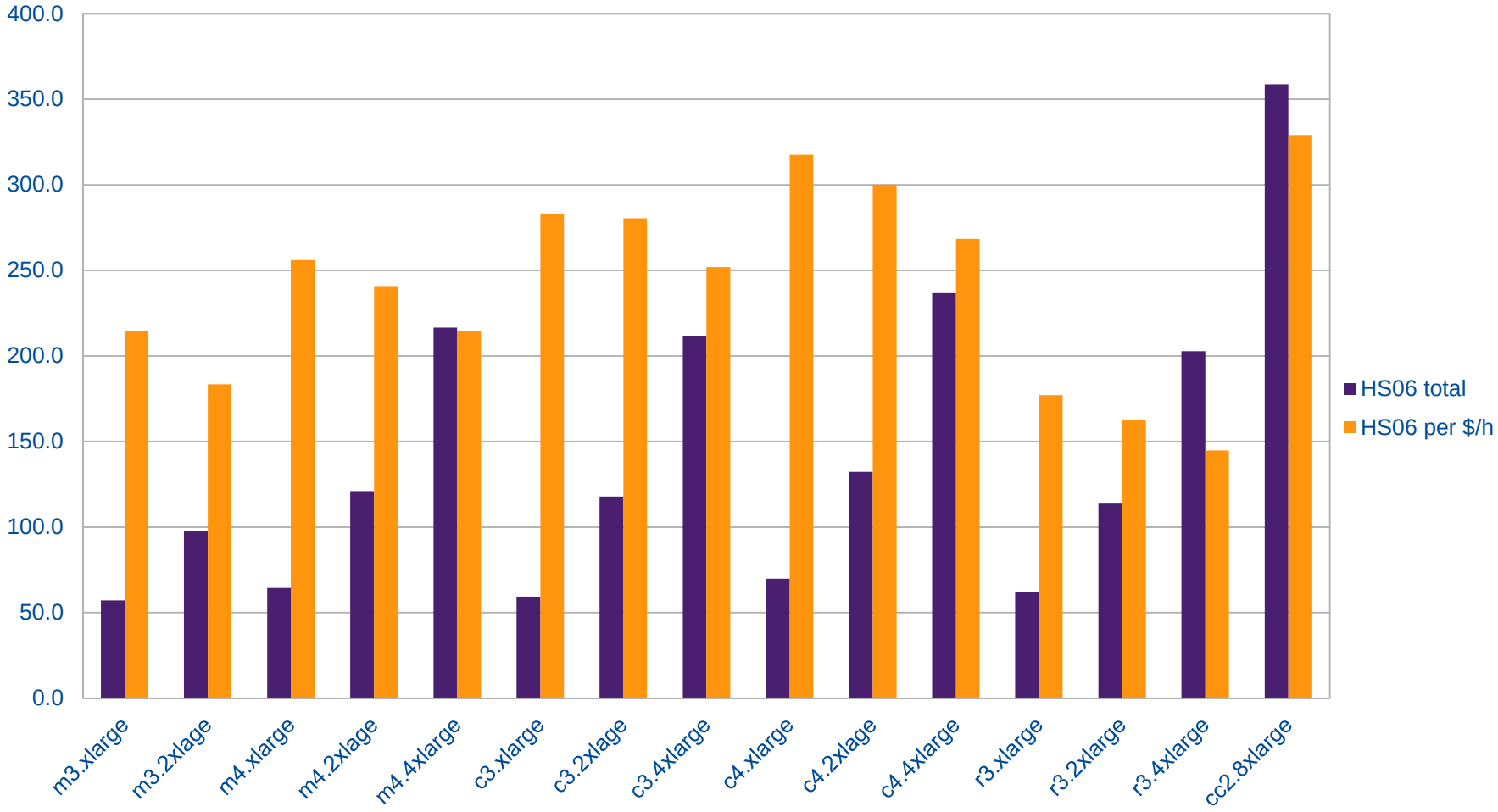
Obtained results

- $t\bar{t}$bar_gensim



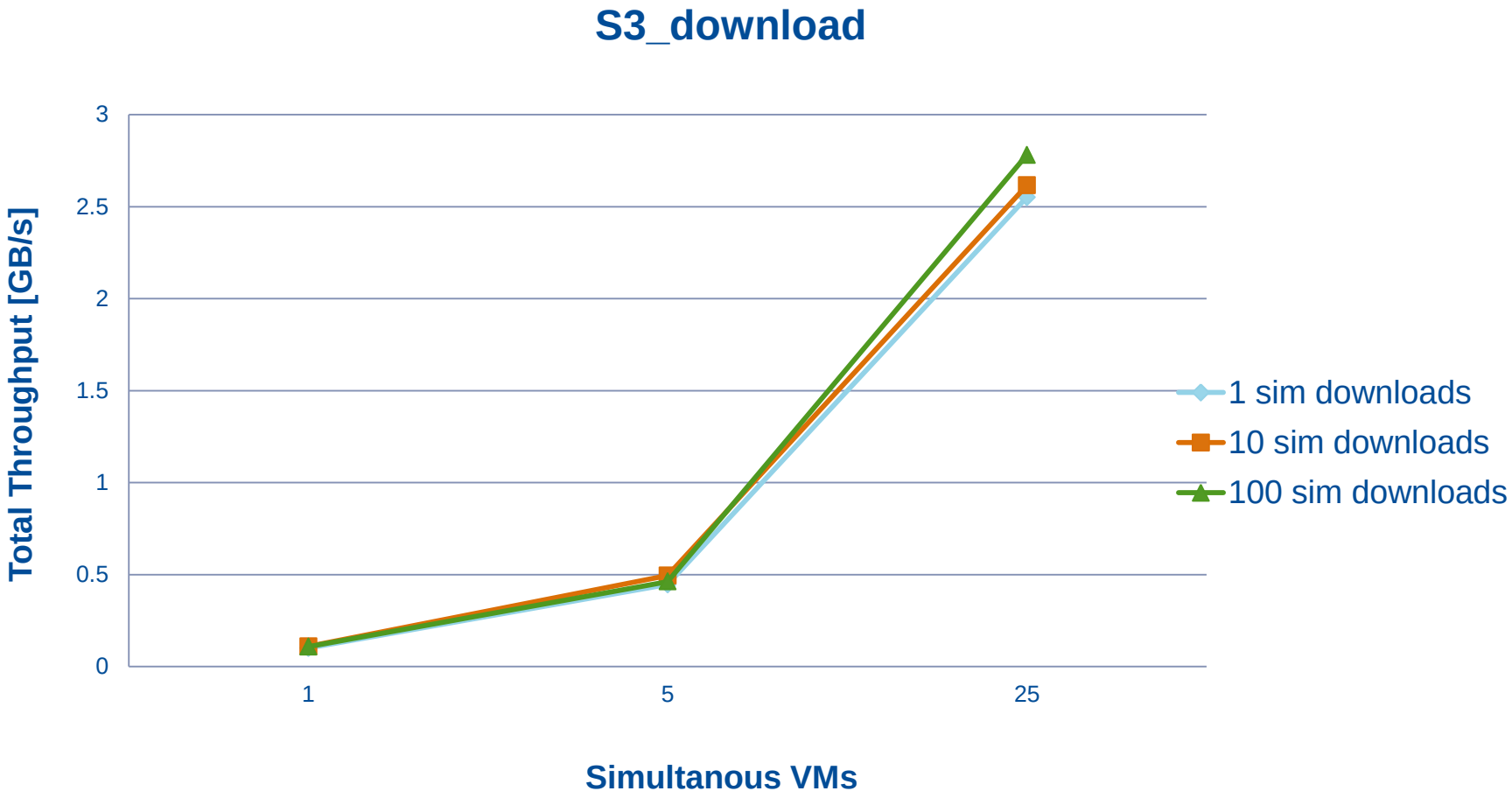
Obtained results

- hepspec06



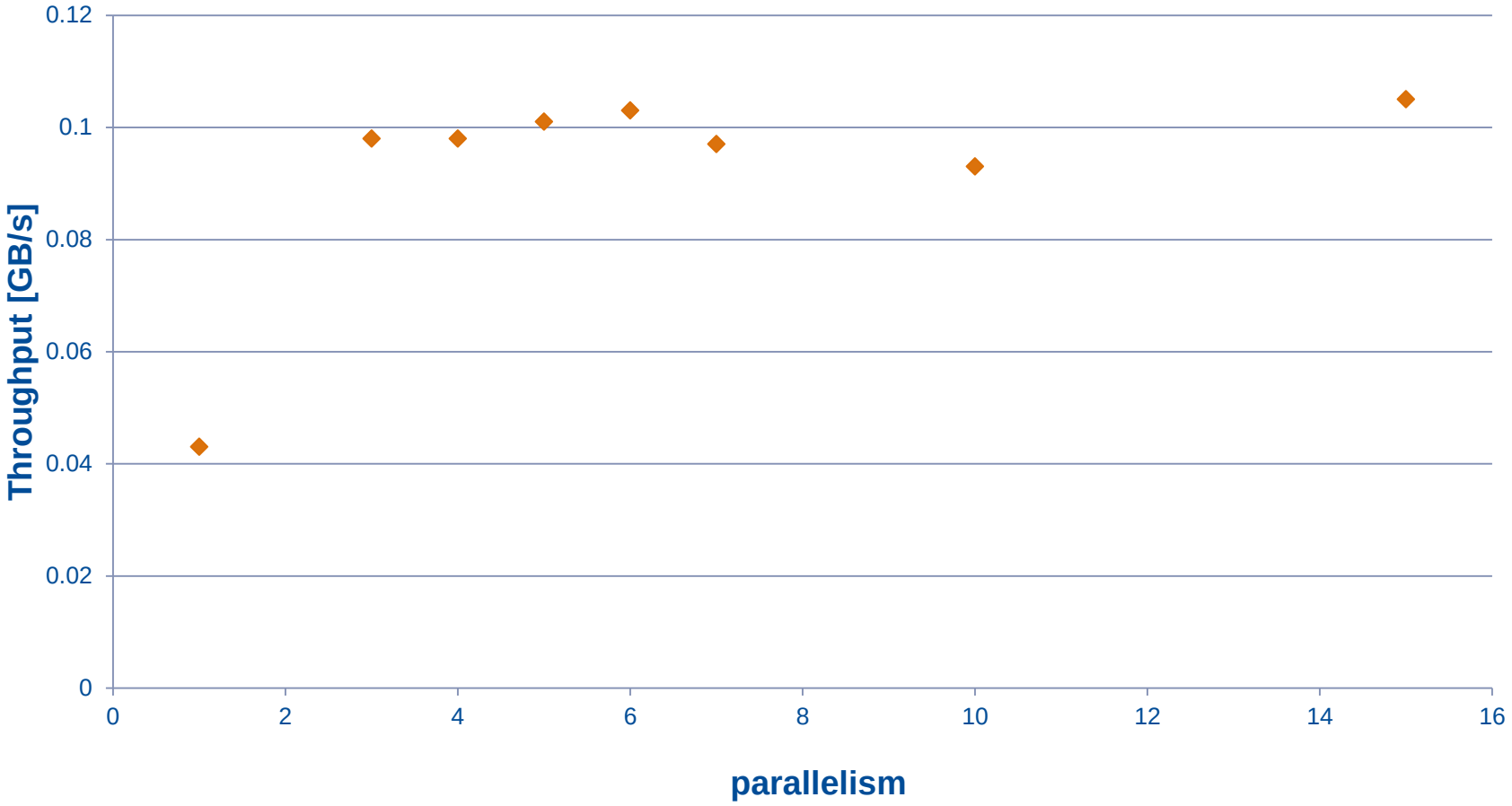
Obtained results

- Amazon S3 (Amazon storage device)



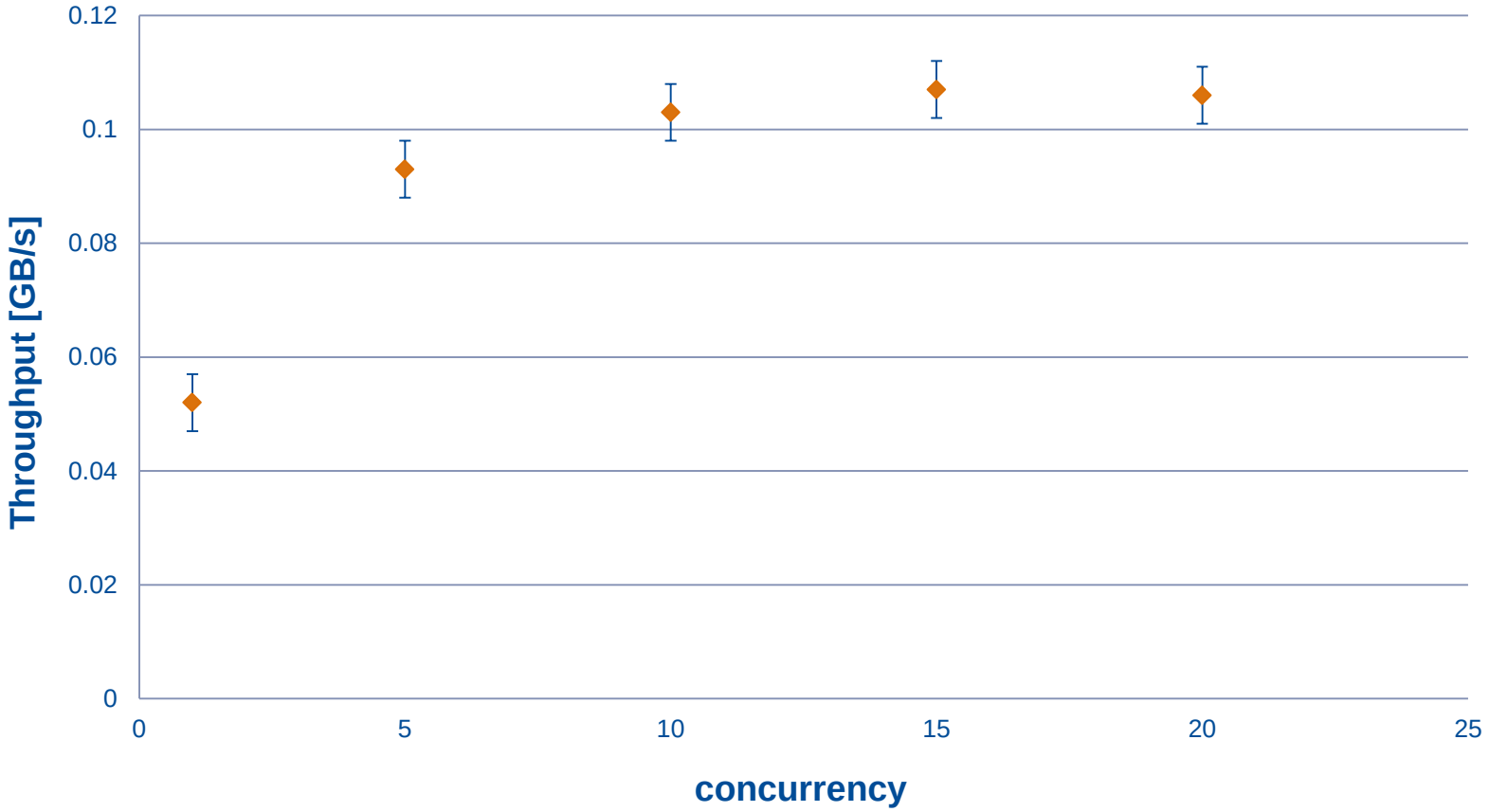
Obtained results

- Parallelism test for bandwidth



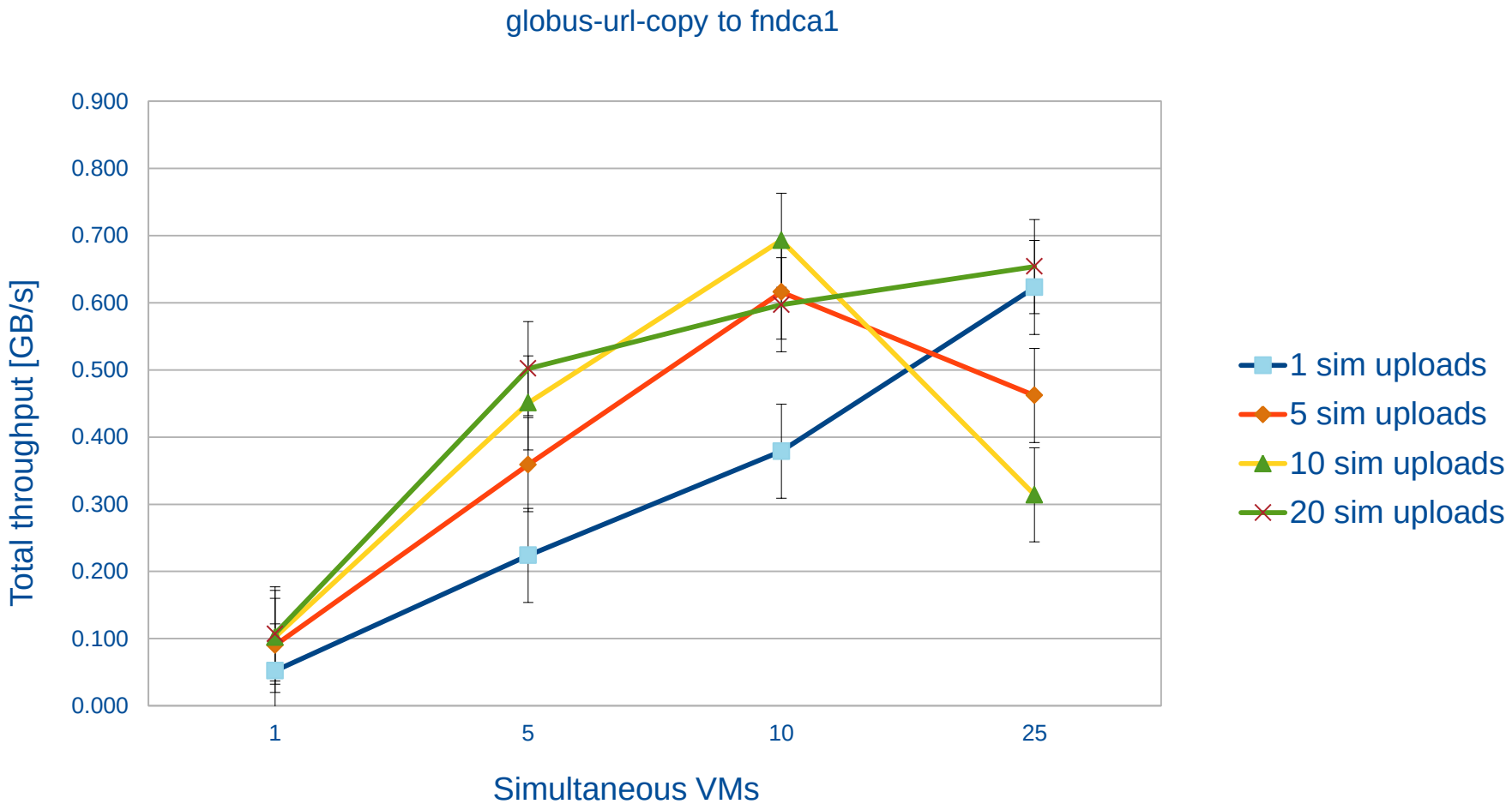
Obtained results

- Concurrency test for bandwidth



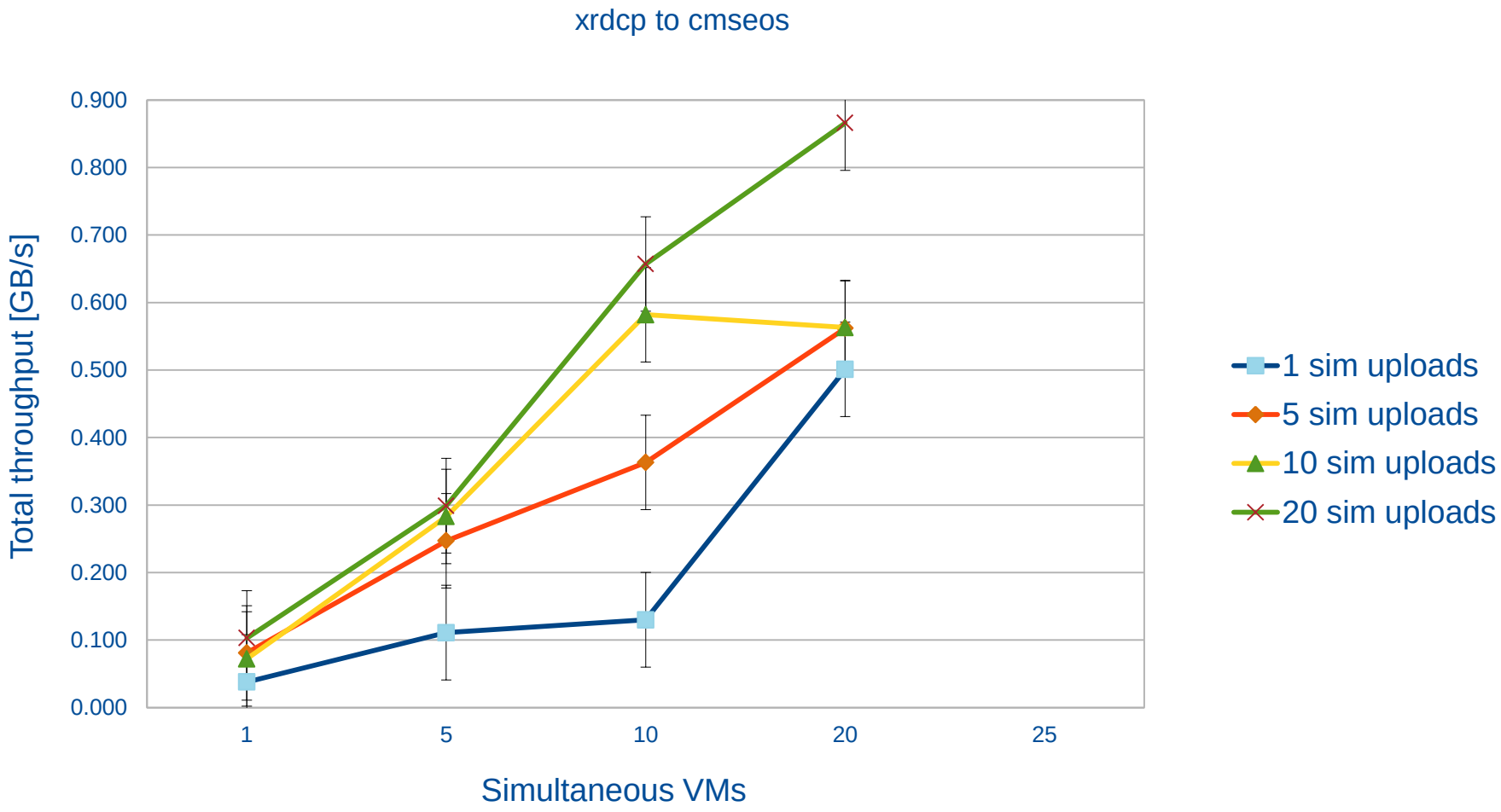
Obtained results

- Globus-url-copy to FermiGrid



Obtained results

- xrdcp to cmseos



Conclusions

- By comparing benchmarks of AWS resources with local ones, it could be seen that the values are very comparable.
- A CMS job should use 56000 cores at once, each sending a 1GB file on average every 8 hours.
- From this numbers and the bandwidth measured with the benchmarks, it was seen that we can reach 2.5x the amount of requested bandwidth for a CMS job

Thanks for your attention