

# IaaS: Cloud computing, for research

ChemIT currently does not have plans nor resources to invest in R&D or consulting services to help CCB researchers evaluate or utilize the power of cloud computing for research. However, this page can raise awareness of these new services' potential to advance CCB computing, cost-effectively.

- See also
- Chemistry "case study" of a web server, hosted using Amazon's Web Service (AWS)
- Cornell "case study", using Amazon's Web Service (AWS)
- Google Compute's calculator and comparing to ChemIT's clusters
  - Comparing Google Compute to a ChemIT's systems.
- Articles
- Vendor links
- Other
  - Roger's thoughts, from 7/30/14:

## See also

- Restricted: Two CU researchers' Cloud case-studies (Cornell NetID required; cited below, too)
- Ramping up and down cloud services to save money
  - <http://www.docker.com/>
  - <http://agaveapi.co/reproducible-science-with-agave-and-docker/>
- Cornell's own CAC's RedCloud:
  - [www.cac.cornell.edu/redcloud](http://www.cac.cornell.edu/redcloud)
- <http://cloudcomputing.cornell.edu/Research.php> (added 3/20/15)
  - This is sponsored by the Computer Science department. (Not tied to CIT or CAC.)
- <http://itnews.cornell.edu/cloudification-services-will-offer-ramp-cloud-infrastructure> (3/13/15 CIT article)
- A Cornell IT manager peer shared this nice digestible interview/talk with one of his favorite experts talking about cloud and IT for life sciences:
  - <http://info.averesystems.com/blog-0/the-state-of-it-infrastructure-in-the-life-sciences>
  - Note: Cross-posted in Yammer, <<http://www.it.cornell.edu/services/yammer/>>.

## Chemistry "case study" of a web server, hosted using Amazon's Web Service (AWS)

- September 2016: CIT successfully packaged old NMR Scheduler's Python code and text files and moved it to contemporary OS and web server software running on AWS. They had been running on an old, not patched or maintained, Linux and Apache public web server.
  - <http://nmrscheduler.chem.cornell.edu/>
    - <http://nmrscheduler.chem.cornell.edu/cgi-bin/login.pl>
  - Cloud Support
- Automatic monthly billing via internal Cornell KFS. Over the past week (9/9/16), the NMR Scheduler AWS resources cost about \$0.20/day Expect to see this rise a little when we release it. Still CIT folks would be very surprised if it was more than \$0.50/day all told.
  - For 365 days/ year, that comes out to ~\$73-183/year for a patched and otherwise maintained Linux and Apache public web server running custom, old Perl scripts. And one easily replicated, if necessary.
  - [AWS Standard Tagging](#)

## Cornell "case study", using Amazon's Web Service (AWS)

The "case study" cited here contains examples of innovative use of cloud-based infrastructure for provisioning research-based computing power similar to what CCB researchers get by investing in high-performance computers and clusters.

N.B. Other academic departments in A&S which could benefit from cloud computing, per Frank Strickland (3/20/15), include Linguistics, Anthropology, Psychology, and Econ. Some of these are listed in the [DNSDB entry for Cornell's cloud computing research hub](#).

- Restricted: Two CU researchers' Cloud case-studies (Cornell NetID required)

In addition to Amazon, services and providers in this space include [Cornell's own CAC's RedCloud](#), as well as Google Compute Engine and Microsoft Azure.

## Google Compute's calculator and comparing to ChemIT's clusters

- <https://cloud.google.com/products/calculator/>

## Comparing Google Compute to a ChemIT's systems.

The Google Compute service offering we compared was 4 of their "CP-COMPUTEENGINE-VMIMAGE-N1-STANDARD-8" (8 core, 30GB RAM) systems.

Using that service, expect to pay an Effective Hourly Rate of \$0.392. And Monthly total of \$1,132.10 (24/7, all month)

- And that does not include upload/ download or data storage costs.
- \$1,423.21 for 16 cores, the next jump up.
- Question: Quality of processes

ChemIT buys a 4-computer, 2U system that costs ~\$10,000 (2.5K\*4) (10 core,

- And that does not include ChemIT's services (CCB invests ~\$100,000 per year for this) or data storage costs.
- There are no upload/ download costs.

| Offering  | Core count compared (performance, though?)                                  | RAM (FWIW)          | Cost  | Cost comparison  |
|---|---|---------------------|---|--|
| ChemIT  | <b>48 cores</b><br>(6 cores/ proc. *<br>2 procs/ computer *<br>4 computers) | 32-64GB,<br>usually | <b>\$10,000 total hardware</b><br><br>(~\$2,600/computer *<br>4 computers/2U unit)                    | <b>\$2,500/ yr</b><br><br>(Assumes last 4 years, 3 of which are under warranty)<br><br>Lots of local IT labor costs. (Maybe \$10K/ yr, at least for first set of 4?) |
| Google Compute  | <b>32 cores</b><br>(8 cores/ computer *<br>4 computers)                     | 30GB                | <b>\$1,132.10 per month.</b><br><br>(Used Google Compute's<br>calculator,<br>running 24/7, all month) | <b>\$13,585.20/ yr</b><br><br>No local IT labor costs.   |
| Google Compute:<br>More cores and<br>RAM<br>( n1-standard-16) | <b>64 cores</b><br>(16 cores/ computer *<br>4 computers)                    | 60GB                | <b>\$2,264.19 per month.</b><br><br>(Used Google Compute's<br>calculator,<br>running 24/7, all month) | <b>\$27,170.28/ yr</b><br><br>No local IT labor costs.   |
| Google Compute:<br>More cores, less<br>RAM<br>(n1-highcpu-16) | <b>64 cores</b><br>(16 cores/ computer *<br>4 computers)                    | 14.4GB              | <b>\$1,423.21 per month.</b><br><br>(Used Google Compute's<br>calculator,<br>running 24/7, all month) | <b>\$17,078.52/ yr</b><br><br>No local IT labor costs.   |

## Articles

[http://www.theregister.co.uk/2016/02/15/nice\\_catch\\_amazon\\_bezos\\_buys\\_hpc\\_toolkit\\_from\\_italy/](http://www.theregister.co.uk/2016/02/15/nice_catch_amazon_bezos_buys_hpc_toolkit_from_italy/)

- AWS already offers HPC-clusters-as-a-service. Once it beds in NICE, it will also have a fine way to make those clusters attack certain workloads. And once it is in that position, it will be in a position to take more of the HPC market and give yet another corner of on-premises kit market a kicking.
  - <https://aws.amazon.com/blogs/aws/amazon-web-services-to-acquire-nice/>
  - <http://www.nice-software.com/news/-/blogs/nice-to-join-forces-with-amazon-web-services>

<http://www.admin-magazine.com/Archive/2014/21/Building-Big-Iron-in-the-Cloud-with-Google-Compute-Engine/>

- (...) **Google Compute Engine** allows you quickly and easily to create anything from a simple single-node VM to a large-scale compute cluster on Google's world class infrastructure. As of this writing, it supports several stellar open source Linux distributions (and one closed-source option), including Debian and **CentOS**; CoreOS, FreeBSD, and SELinux [2]; and Red Hat Enterprise Linux, SUSE, and Windows. Instances are available with many options and are completely customizable from a hardware perspective. You can choose the number of cores, RAM, and other machine properties, and you can scale them as you grow [4]. Virtual instances start at a micro instance (f1-micro), with one core and 0.60GB of memory, and go up to **16 cores** and 104GB of RAM. For the sake of the demo here, I will be using a shared core micro instance (g1-small; Table 2). Competition from Amazon, Microsoft, Rackspace, and others in the cloud marketplace has put increasing downward pressure on the price of many cloud offerings.(...)

<http://www.admin-magazine.com/HPC/Articles/Building-big-iron-in-the-cloud-with-Google-Compute-Engine/> (7/2014)

- (...) **Google Compute Engine** was opened to the public in June 2012, a bit later than most other players in the cloud marketplace. Arrival time aside, it is a powerful, scalable, and performant IaaS solution.

Compute Engine allows you quickly and easily to create anything from a simple single-node VM to a **large-scale compute cluster** on Google's world class infrastructure. As of this writing, it supports several stellar open source Linux distributions (and one closed-source option), including Debian and CentOS; CoreOS, FreeBSD, and SELinux [2]; and Red Hat Enterprise Linux, SUSE, and Windows [3].

Instances are available with many options and are completely customizable from a hardware perspective. You can choose the number of cores, RAM, and other machine properties, and you can scale them as you grow [4]. Virtual instances start at a micro instance (f1-micro), with one core and 0.60GB of memory, and go up to 16 cores and 104GB of RAM. For the sake of the demo here, I will be using a shared core micro instance (g1-small; Table 2). Competition from **Amazon, Microsoft, Rackspace, and others** in the cloud marketplace has put increasing downward pressure on the price of many cloud offerings.(...)

<http://www.infoworld.com/d/cloud-computing/ultimate-cloud-speed-tests-amazon-vs-google-vs-windows-azure-237169>

- February 26, 2014: Ultimate cloud speed tests: Amazon vs. Google vs. Windows Azure

A diverse set of real-world Java benchmarks shows **Google** is fastest, **Azure** is slowest, and **Amazon** is priciest.

## Vendor links

Links to vendors not linked in any of the above analysis sections.

<http://research.microsoft.com/en-us/projects/azure/>

The Microsoft Azure for Research project facilitates and accelerates scholarly and scientific research by enabling researchers to use the power of Microsoft Azure to perform big data computations in the cloud. Take full advantage of the power and scalability of cloud computing for collaboration, computation, and data-intensive processing. Microsoft Azure is an open platform that supports languages, tools, or frameworks, such as Linux [...].

<http://research.microsoft.com/en-us/projects/azure/technical-papers.aspx>

The information in these papers can be used by Windows, Linux, and Mac users. If you have attended the Microsoft Azure for Research training, have received an award through the RFP program, or are just curious about Microsoft Azure, we believe you will find this content useful. The papers do assume some prior technical computer programming skills, such as Python, Matlab, and basic scripting.

## Other

### Roger's thoughts, from 7/30/14:

- I think cloud clustering has lots of potential for getting computations done quickly, or without permanent systems. (surge)
- I believe it will take fair amount of development & learning time (which we currently don't have much of). A working example would be useful.
- The above cited Cornell researcher's "case study" is somewhat informative to IT-type folks such as us, but on the technical side. It is non-specific regarding costs, effort, results, or comparing to physical system costs.
- Suggestions for making this "case study" more useful to our managers and researchers:
  1. Create a summary of the case study.
  2. Put together some numbers to allow comparison of this example and Chemistry's research computing needs and current practices.