COMP28112 – Lecture 20

Grid and Cloud Computing
(and more…)

An Early Experiment

The advent of the Internet made scientists think about the possibility of exploiting interconnected machines for time-consuming applications. E.g.:

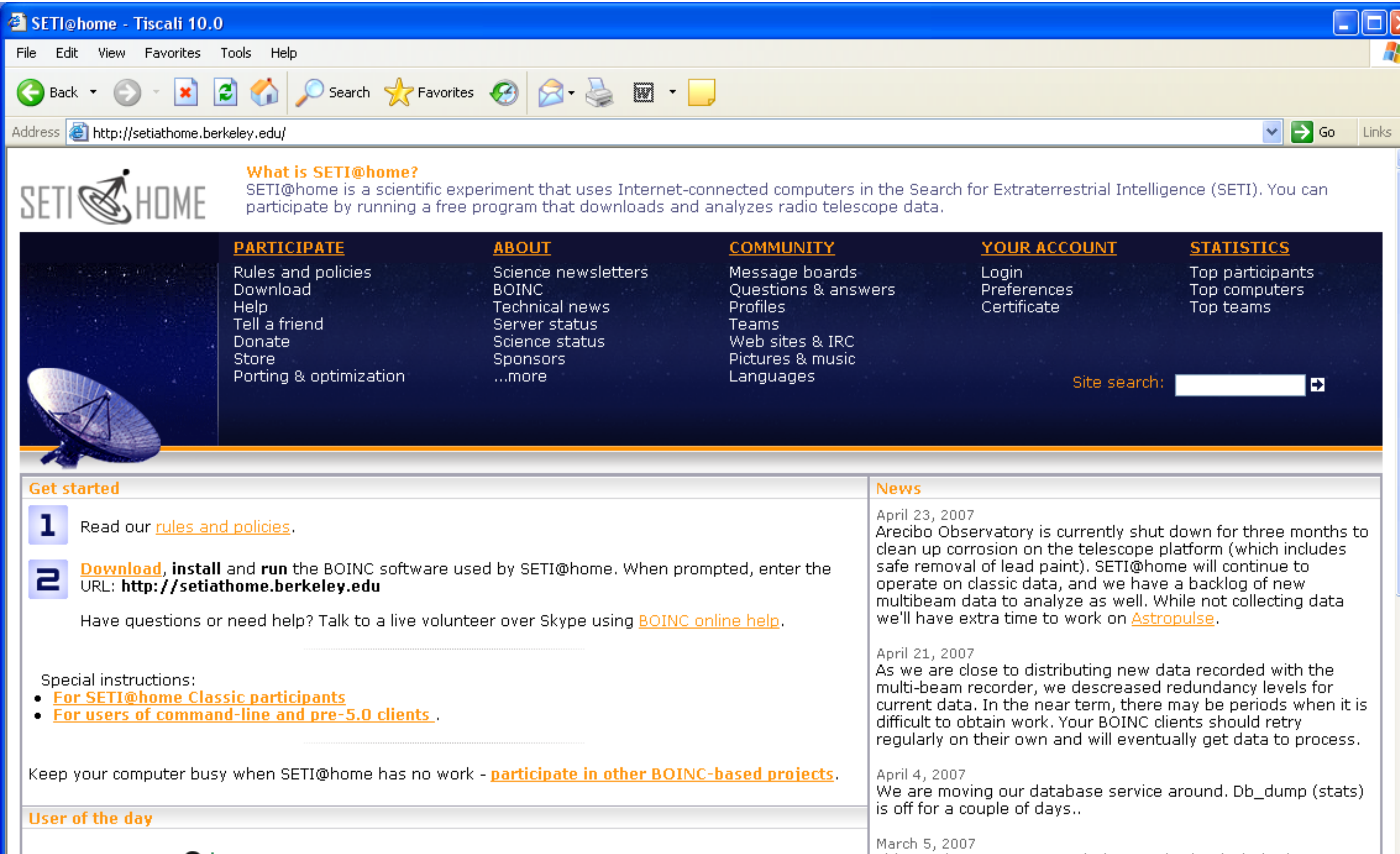
- Long Integer Factorisation (find the prime factors of an integer – remember: public-key cryptography relies upon the difficulty of finding the prime factors of long integers):
  - Algorithms for integer factorisation might be time consuming: for integers whose factors are two primes of about the same size, no polynomial time algorithm (to find the factors) is known.
  - http://en.wikipedia.org/wiki/Integer_factorization

An Early Experiment (cont.)

- Around 1990, the Internet mostly consists of Unix machines.
- A C program (implementing an integer factorisation algorithm) was developed that would run on a machine when it was idle; it would use email to communicate with a server, to email results, to request data:
  - At a time when factoring 100-digit-long integers would take one month using expensive machines, it was realized that **with a good implementation** such integers could be factored within a few days and for free!
  - Read “factoring by electronic mail”, EUROCRYPT1989

SETI @ HOME

(download and analyse radio telescope data)



Is all this (number crunching) useful?

- Science (and human progress) rely on curiosity…
- Frank Nelson Cole found in 1903 the prime factors of $2^{67}-1$:
  - $2^{67}-1 = 193707721 \times 761838257287$
  - How long did it take him? "Three years on Sundays". How long would it take today?

(read: "In Praise of Science: Curiosity, Understanding, and Progress")

The origins of the Grid

- The term ‘Grid’ was coined around 1996.
  - It was used to describe a hardware and software infrastructure that was needed by the rapidly growing and highly advanced community of High Performance Computing (HPC):
    - HPC refers to the use of (parallel) supercomputers and computer clusters linked together in a way that provides high-end capabilities; the latter are needed for a number of scientific applications (‘big science’).

WARNING: This was a vision!

What is the vision of the Grid?

- "*A computational Grid is a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities*"

  ("The Grid: Blueprint for a New Computing Infrastructure",1998)

- Why 'Grid'?
  - Electricity Grid: "A network of high-voltage transmission lines and connections that supply electricity from a number of generating stations to various distribution centres, so that no consumer is dependent on a single station".
  - As early as 1969, it was suggested that: "We will probably see the spread of computer utilities, which, like present electric and telephone utilities, will service individual homes and offices across the country".

So, what is the Grid?

- The original term was ‘catchy’
  - Soon, researchers started talking about:
    - Data Grids
    - Knowledge Grids
    - Access Grids
    - Science Grids
    - Bio Grids
    - Sensor Grids
    - Campus Grids
    - Tera Grids
    - Commodity Grids, and so on…

The sceptic would wonder if there was more to the Grid than a ‘funding concept’.

A Grid checklist

- The Grid coordinates resources that are not subject to centralized control … (i.e., resources of different companies, or different administrative domains)
- … using <u>standard</u>, <u>open</u>, general-purpose protocols and interfaces…
- … to deliver non-trivial qualities of service (for example, related to response time, throughput, availability, security, …)

Defining the Grid…

- There are competing definitions, defining it as:
  - An implementation of distributed computing
  - A common set of interfaces, tools, APIs, …
  - The ability to coordinate resources across different administrative domains (creating Virtual Organisations)
  - A means to provide an abstraction (virtualisation) of resources, services, …
  - Resource sharing and coordinated problem solving in dynamic virtual organisations

Grid computing must provide…

- Resource discovery and information collection and publishing
- Data Management on and between resources
- Process Management on and between resources
- Common Security Mechanisms
- Process and Session Recording/Accounting

Some middleware available…

- Globus (http://www.globus.org)
  - GridFTP (extensions to FTP protocol to cope with HPC and Grid Security)
  - OGSA-DAI (http://www.ogsadai.org.uk/): standard approach for data access on the Grid
  - WS-Resource Framework: allows the use of established Web Services standards (slow, complex, …).
- Condor, Condor-G, HTCondor (http://research.cs.wisc.edu/htcondor/)
  - Workload Management for compute-intensive jobs

Some Grid Infrastructures (and beyond the Grid…)

- Teragrid, now Extreme Science and Engineering Digital Environment (http://www.teragrid.org/)
  - http://access.ncsa.uiuc.edu/witg/
- Datagrid (http://eu-datagrid.web.cern.ch/eu-datagrid/)
  - http://real1.rm.cnr.it:8081/ramgen/Grid/Grid.rm
- UK National Grid Service (http://www.ngs.ac.uk/)
- Many more national initiatives…
- Learn about the grid: http://www.gridcafe.org/
- Beyond the Grid:
  - GLORIAD: http://www.gloriad.org/
  - PlanetLab: http://www.planet-lab.org/

Grid Computing Summary

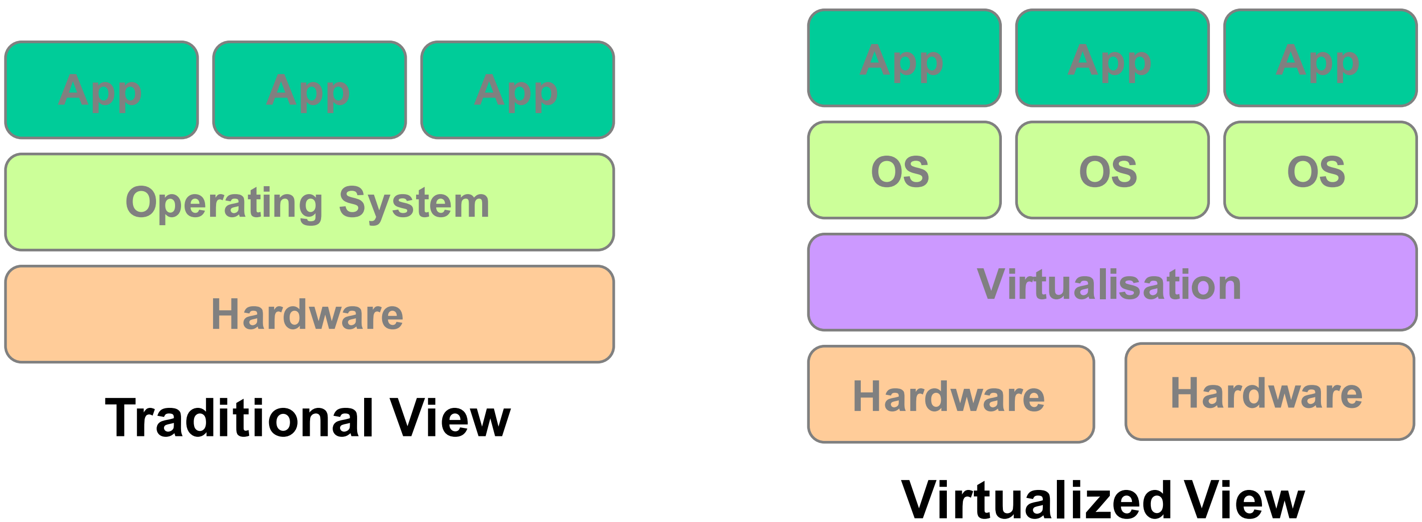
- There is a lot of hype around grid computing…
- …but there is real-world value in e-science, e-business…
- …through virtualisation of the underlying distributed resources!
- The Large Hadron Collider Grid has been set up to support the Large Hadron Collider experiment:
  - http://wlcg.web.cern.ch/
- There has been related research in the School

Cloud Computing

- Largely evolved from the Grid.
- The idea:
  - On-demand resource provisioning. Users of the Cloud are consumers who don't have to own the resources they use; these resources are provided by others (providers) as a service!
- Associated concepts:
  - Software as a service (SaaS)
  - Platform as a service (PaaS)
  - Infrastructure as a service (IaaS)

Virtualisation: the key concept

- In the same way that resources of a single computer are virtualised by traditional OSs, a cloud computing layer can virtualise multiple computers.

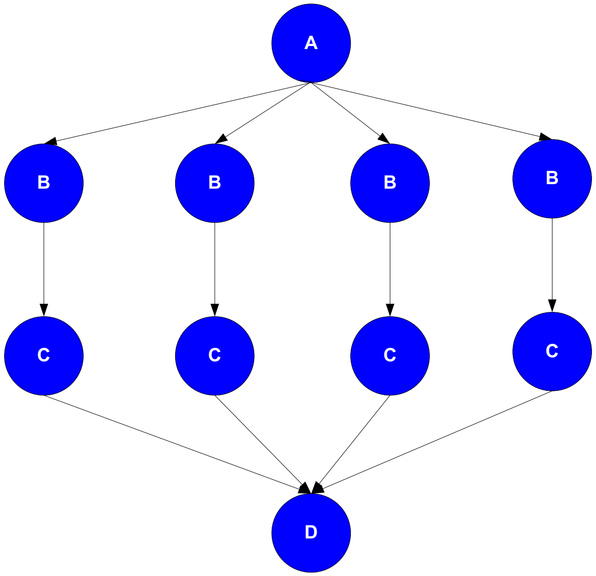


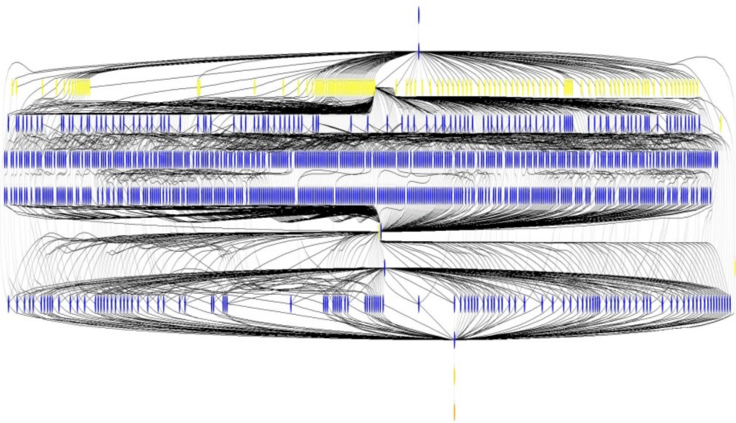
Grid & Cloud

- Similarities: Scalability, Divide and Conquer, Lots of Data, SLAs, …
- Differences: The Grid has a tighter relation between users and administrators; the execution model on the Grid is more restrictive.
- A subject for debate:
  - http://www.ibm.com/developerworks/web/library/wa-cloudgrid/
  - http://ianfoster.typepad.com/blog/2008/08/cloud-grid-what.html
- The Cloud has raised privacy concerns.
- For an example of a Cloud Computing service: Amazon's EC2: http://aws.amazon.com/ec2/
- Grids and (even more) Clouds are here to stay!

The Applications…

- Often, these are scientific workflows consisting of sequences of tasks. Tasks may be partitioned into parallel tasks each of which operates on different data (a technique known as divide-and-conquer)
- Tools are available to enable the automatic composition of the workflows and their mapping onto resources
  - http://pegasus.isi.edu/
- Applications in astronomy, earth sciences, bioinformatics, …





https://pegasus.isi.edu/application-showcase/ligo/

Advanced LIGO pyCBC Workflows

The LIGO Laboratory started the first Observing Run 'O1' with the Advanced LIGO detectors in September 2015 at a sensitivity roughly 4 times greater than Initial LIGO for some classes of sources (e.g., neutron-star binaries), and a much greater sensitivity for larger systems with their peak radiation at lower audio frequencies. The pyCBC analysis workflows are modeled as a single stage pipeline that can analyze data from both LIGO and VIRGO detectors.

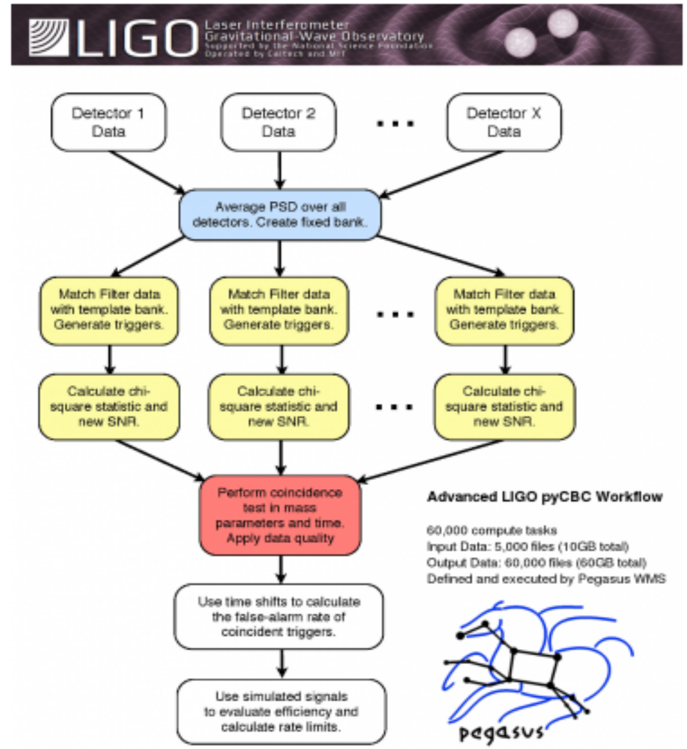
These workflows are executed using Pegasus on a variety of resources such as local campus clusters, LIGO Data Grid, OSG, XSEDE in combination with Pegasus MPI Cluster and VIRGO Clusters.

Each pyCBC workflow has

- 60,000 compute tasks
- Input Data: 5000 files (10GB total)
- Output Data: 60,000 files (60GB total)

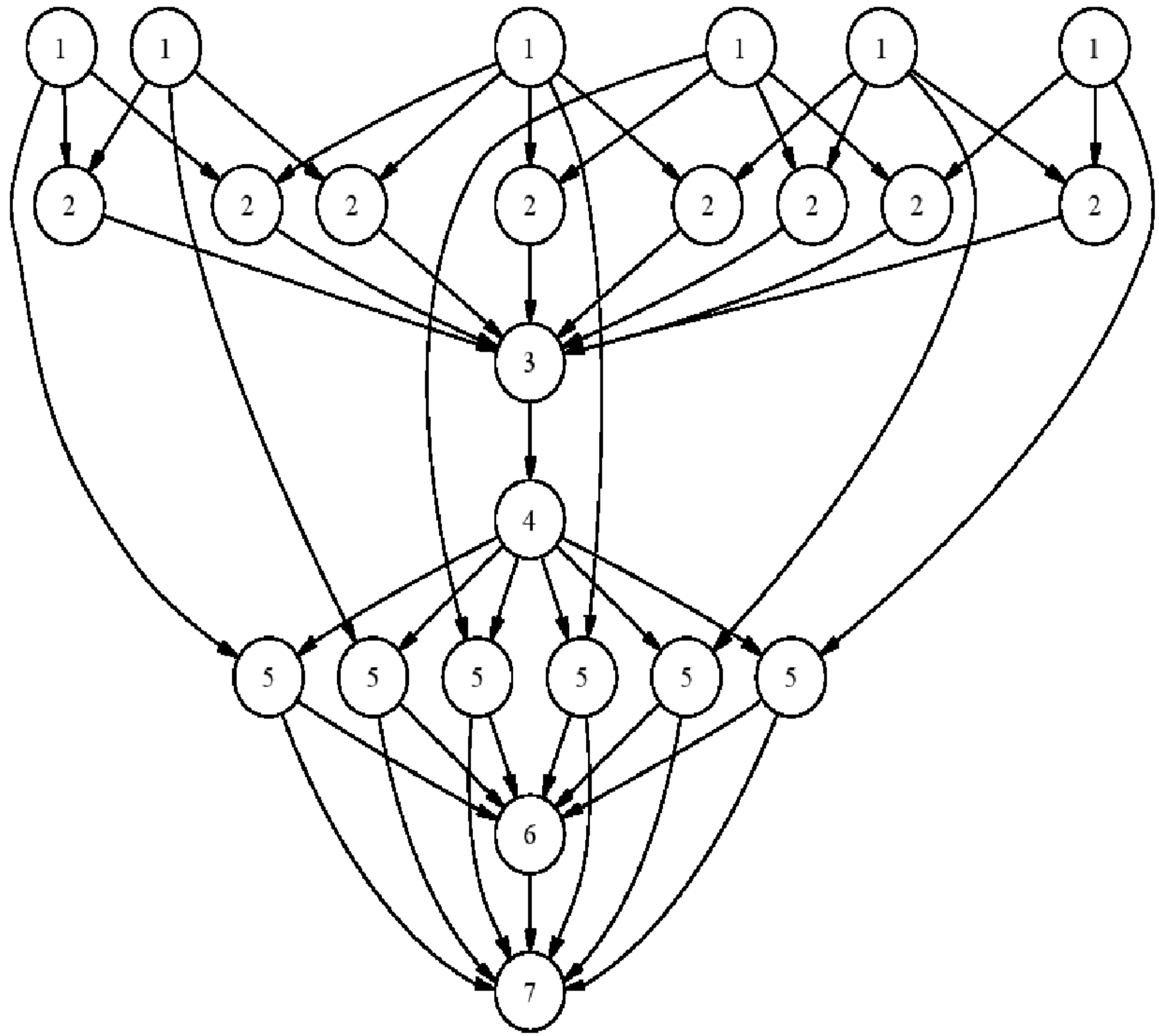
The incorporation of the Pegasus into the production pipeline has enabled LIGO users to

- Run an analysis workflows across sites. Analysis workflows are launched to execute on XSEDE and OSG resources with post processing steps running on LIGO Data Grid.
- Monitor and share workflows using the Pegasus Workflow Dashboard.
- Easier debugging of their workflows.
- Separate their workflow logs directories from the execution directories. Their earlier pipeline required the logs to be the shared filesystem of the clusters. This resulted in scalability issues as the load on the NFS increased drastically when large workflows were launched.
- Ability to re-run analysis later on without running all the sub workflows from start. This leverages the data reuse capabilities of Pegasus. LIGO data may need to be analyzed several times due to changed in e.g. detector calibration or data-quality flags. Complete re-analysis of the data is a very computationally intensive task. By using the workflow reduction capabilities of Pegasus, the LSC and Virgo have been able to re-use existing data products from previous runs, when those data products are suitable.



LIGO pyCBC Workflow. Image Credit: Samantha Usman, Duncan Brown et al

Remember in Lecture 2?



Dealing with large data sets

- Speed-up processing by parallelizing the operations!
- Map-Reduce Framework
  - Apache Hadoop: open source implementation
- The ‘Big Data’ fashion/obsession/buzzword…
  - Data + Framework + Algorithms = Knowledge?
  - http://lsst.org/lsst/google: 30TB of data every night
  - LHC@CERN: 1 petabyte of data per second

A challenge for the future…

- The huge amounts of data we can collect and process will offer us great opportunities to improve quality of life and shape progress in a way that benefits everybody but to achieve this we need a better understanding of the challenges in handling and analyzing large-scale data including new technologies…

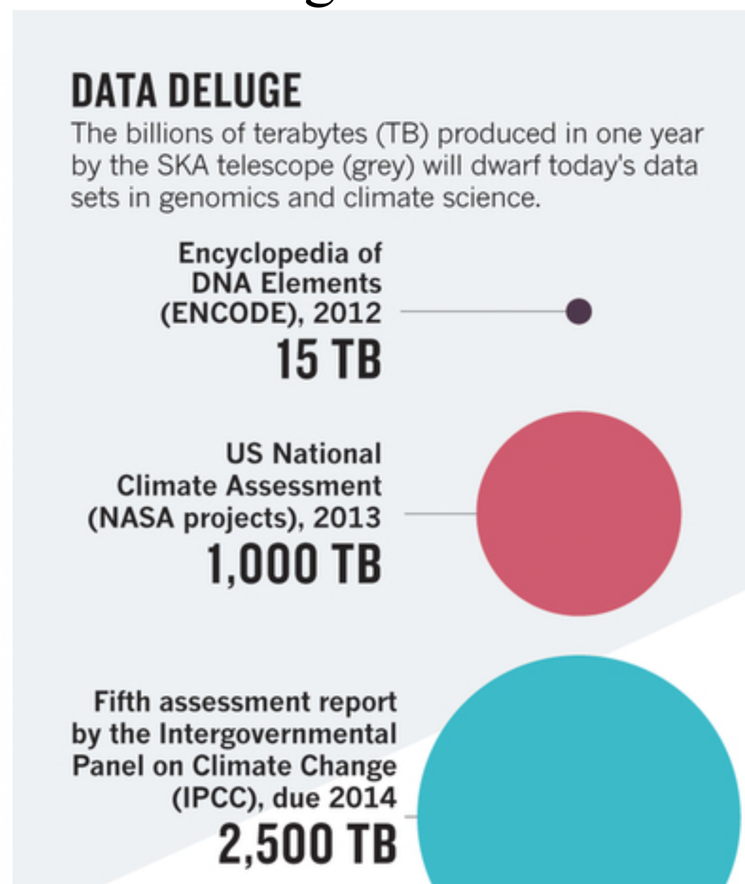
Oct 3, 2012 - 12:51PM PT

NASA tries to free creativity with Big Data Challenge

BY Derrick Harris

5 Comments f +1

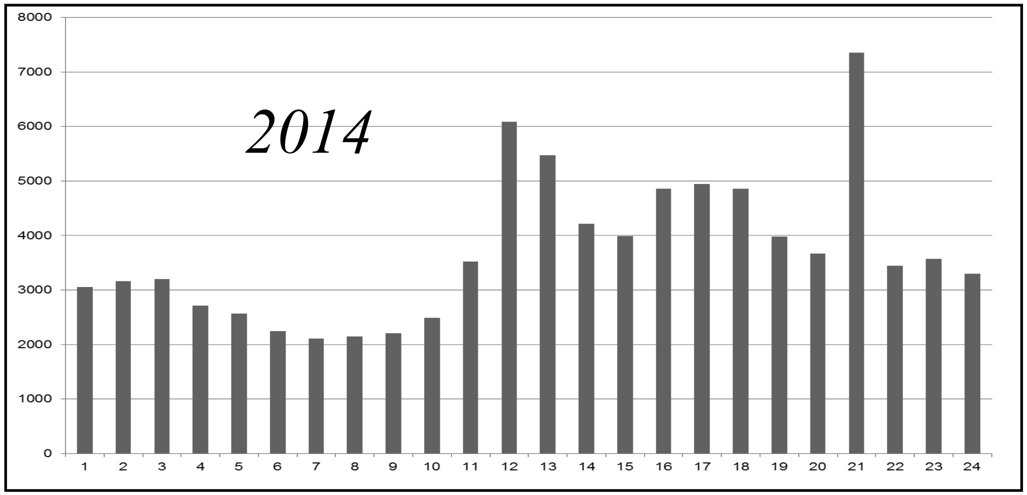
NASA and a couple other government agencies have kicked off a series of TopCoder challenges designed to find innovative solutions to the government's big data problems. The first contest is all about making disparate, incompatible data sets usable and actually valuable across agencies.

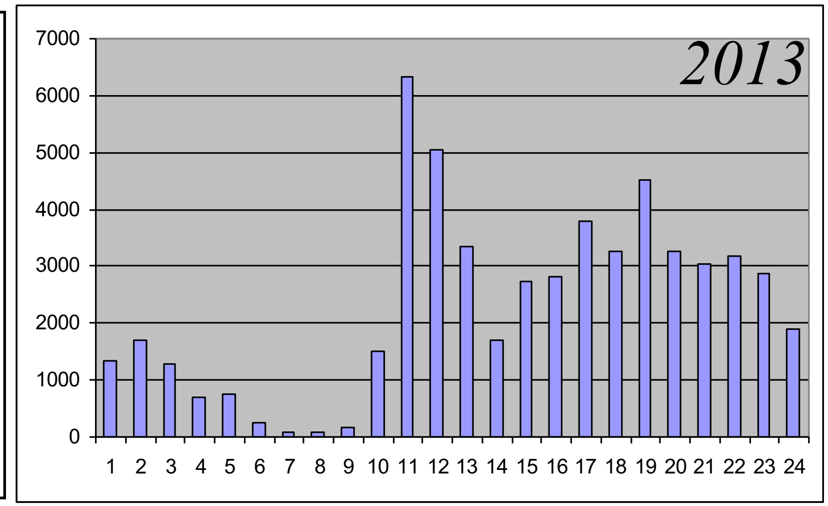


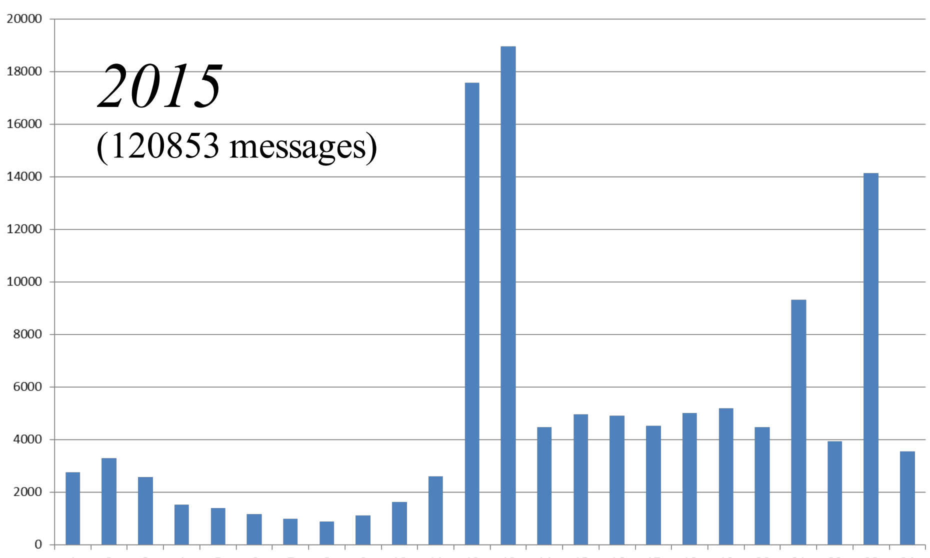
Finally…

Motivating some advanced problems
(but first some statistics about lab 2)

Messages to the server based on hour of the day (0-1, 1-2, …, 23-24)



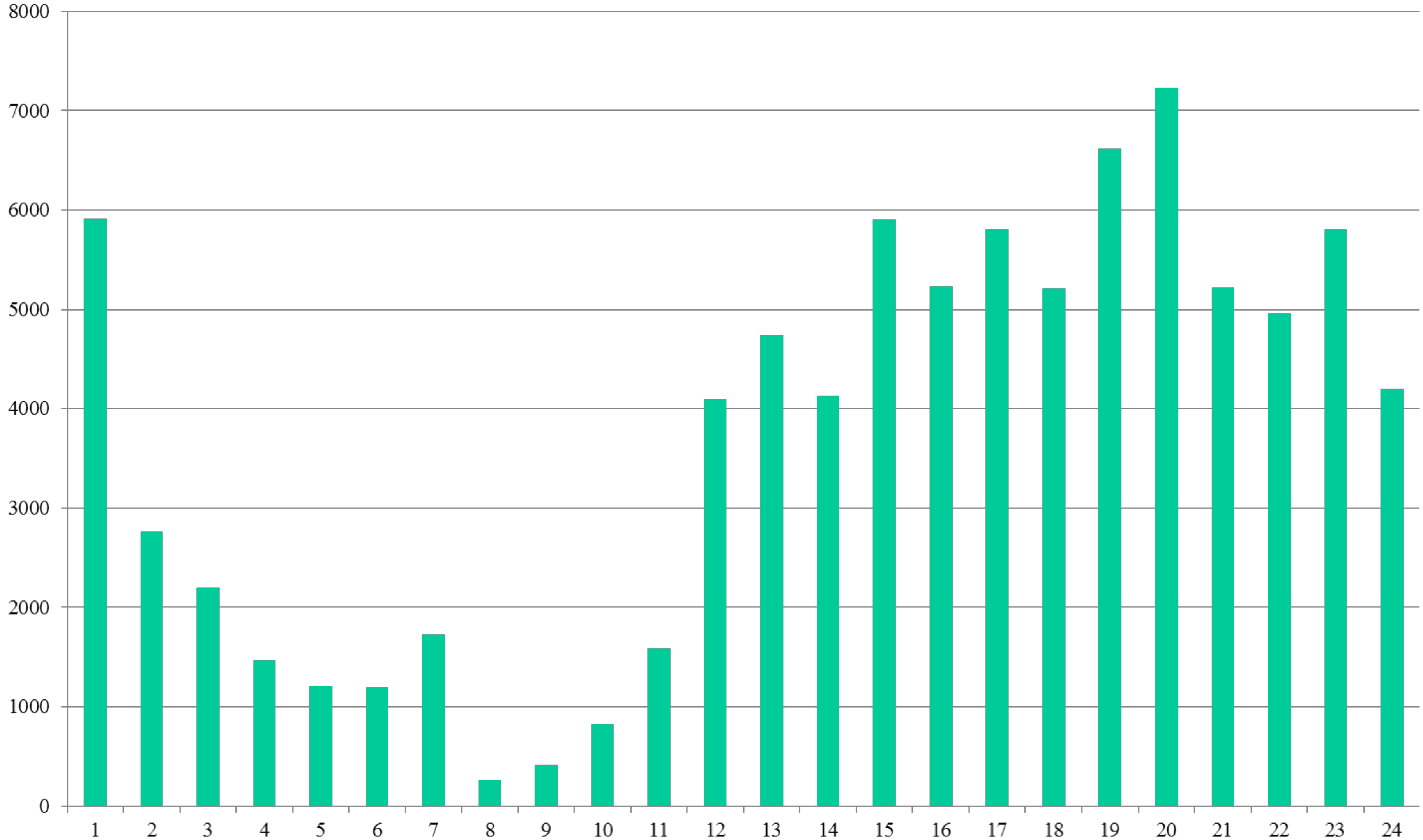




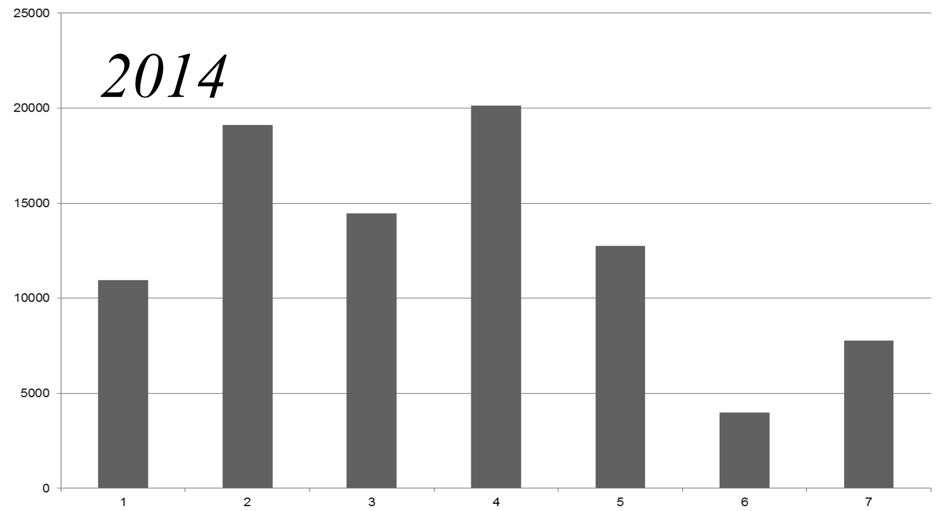
2016

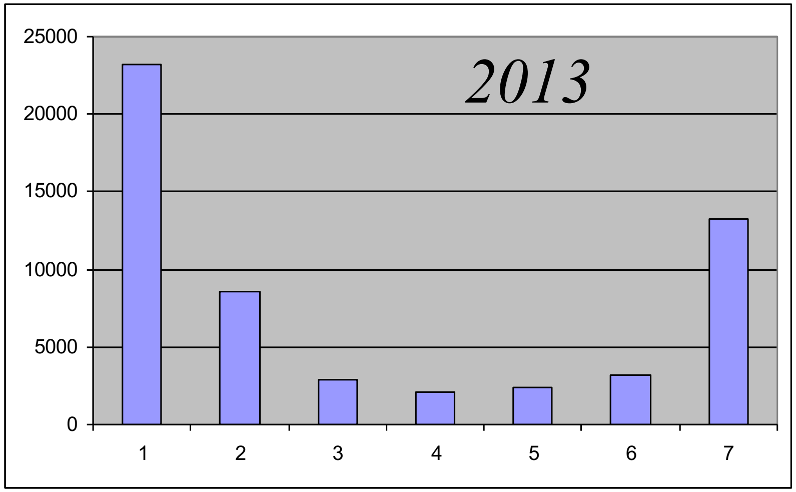
?

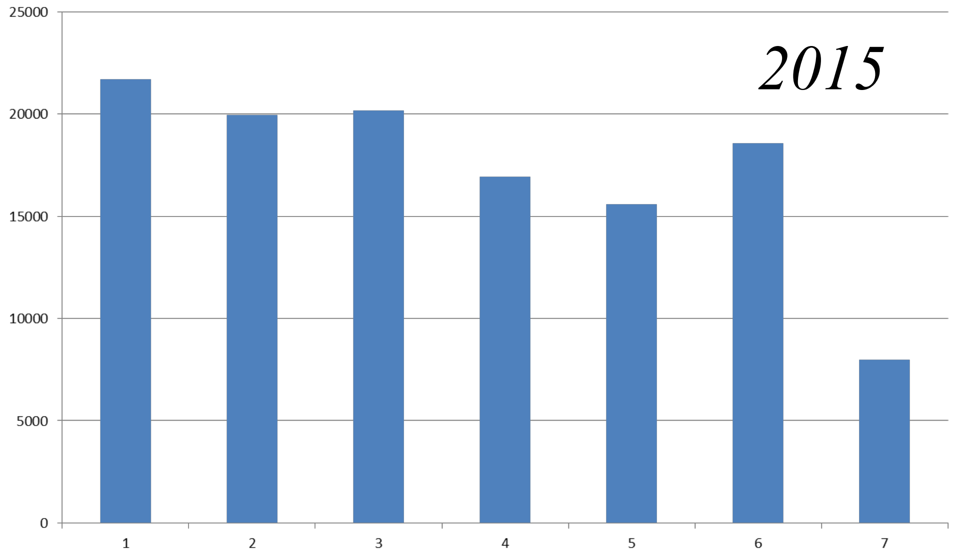
(88718) Messages to the server based on hour of the day (0-1, 1-2, …, 23-24)



What day of the week was busiest? (1: Mon, 7: Sun)



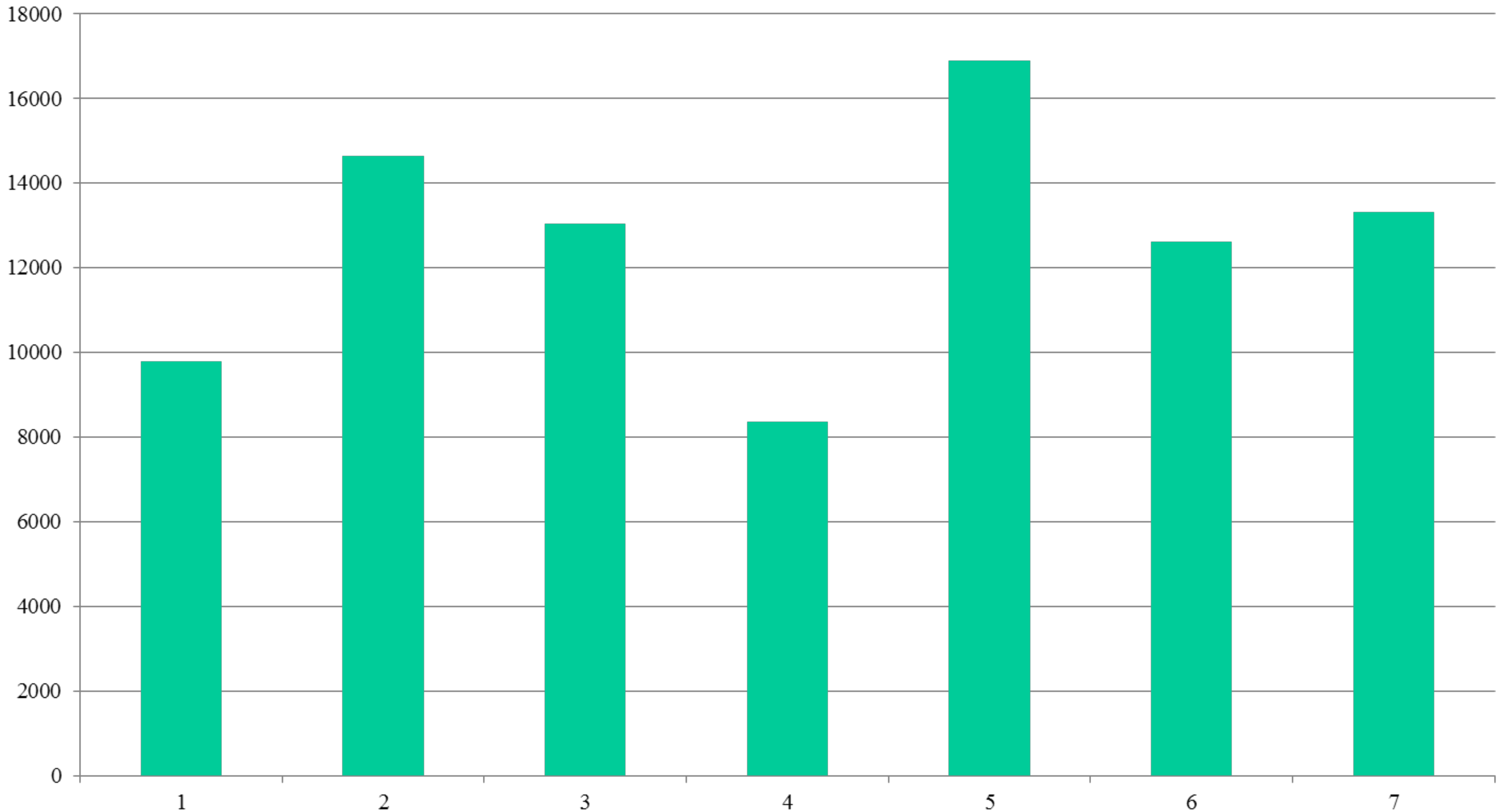




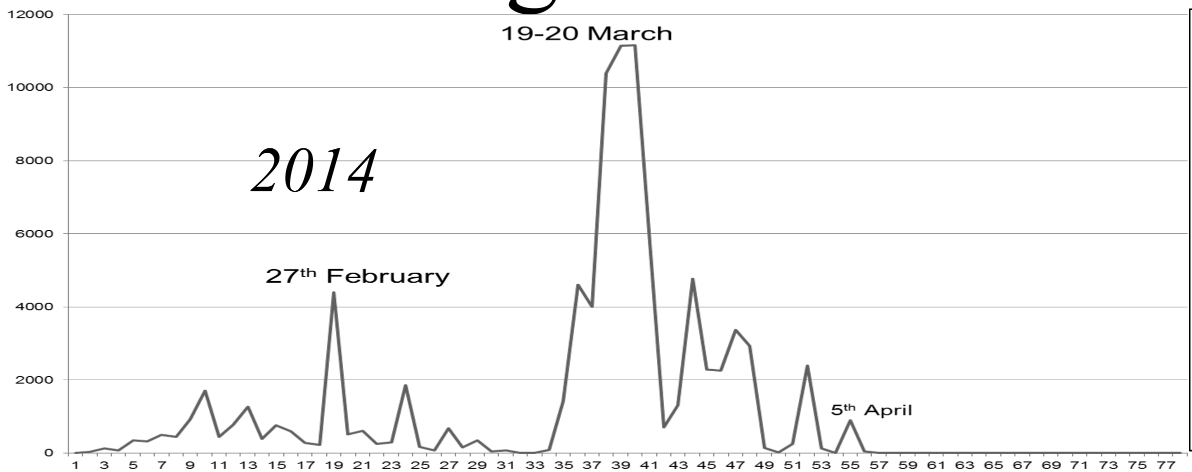
2016

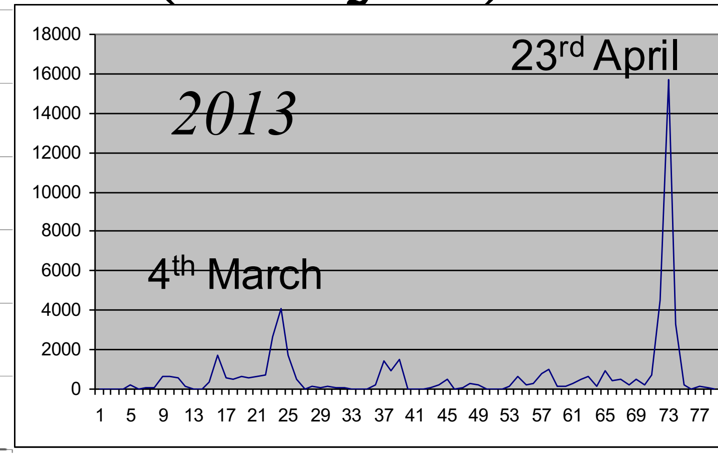
?

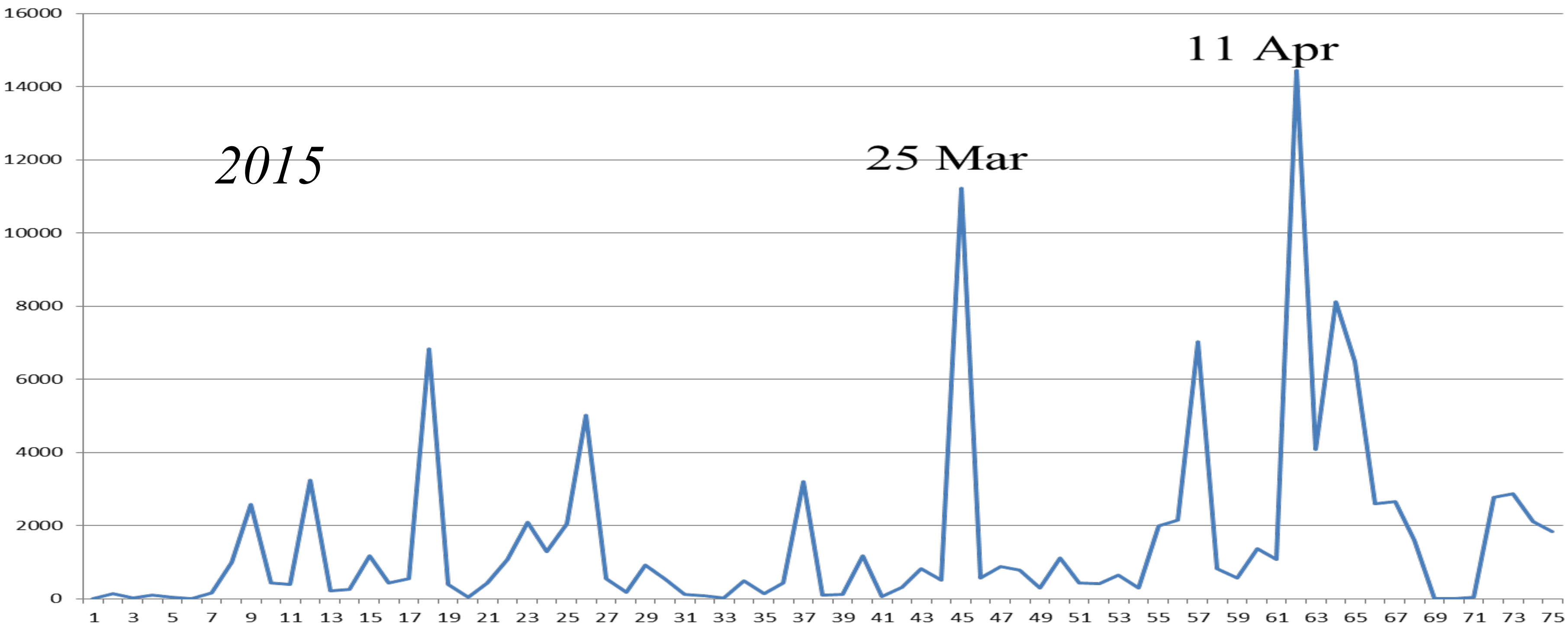
What day of the week was busiest? (1: Mon, 7: Sun)



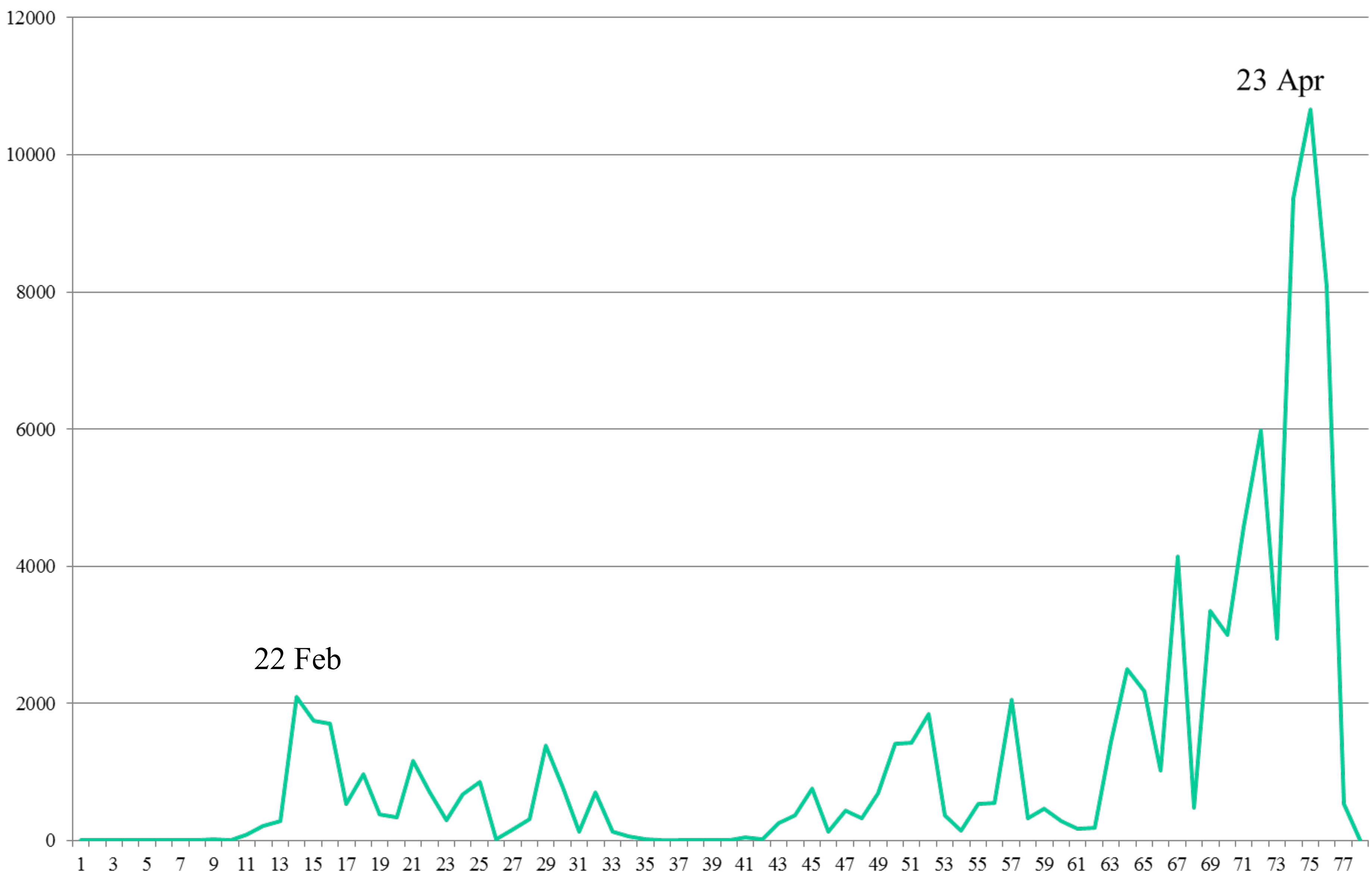
Messages since 9/2/2014 (=day 1)







Messages since 9/2/2016 (=day 1)



Some problems…

1. When is it best to execute two tasks on one machine as opposed to one task on one machine + communication (data transfer) and computation on another?
2. How to design random values, which are not uniform? E.g., two-digit integer where the probability is: 0-9 (90%) and 10-99 (10%)
3. How to compute the average of streamed data? (related to lab exercise 3, where you have to compute the average over large values)
4. Split the following loop, so that each of 4 threads takes the same amount of computation. Rewrite, in terms of t, where t is between 0 and 3.

```
for(i=0;i<1000;i++)
        { some_computation_on_i; }
```

Marking sessions for the labs: May 10, 11, 13
All your lab work should be marked by May 13!