

# Science domains in NeSI cluster job submissions - first results

Gene Soudlenkov  
[g.soudlenkov@auckland.ac.nz](mailto:g.soudlenkov@auckland.ac.nz)



# Outline

- ① NeSI platforms
- ② Pan cluster
- ③ Users
- ④ Jobs
- ⑤ Science Domains and Applications
- ⑥ Conclusion

# NeSI platforms

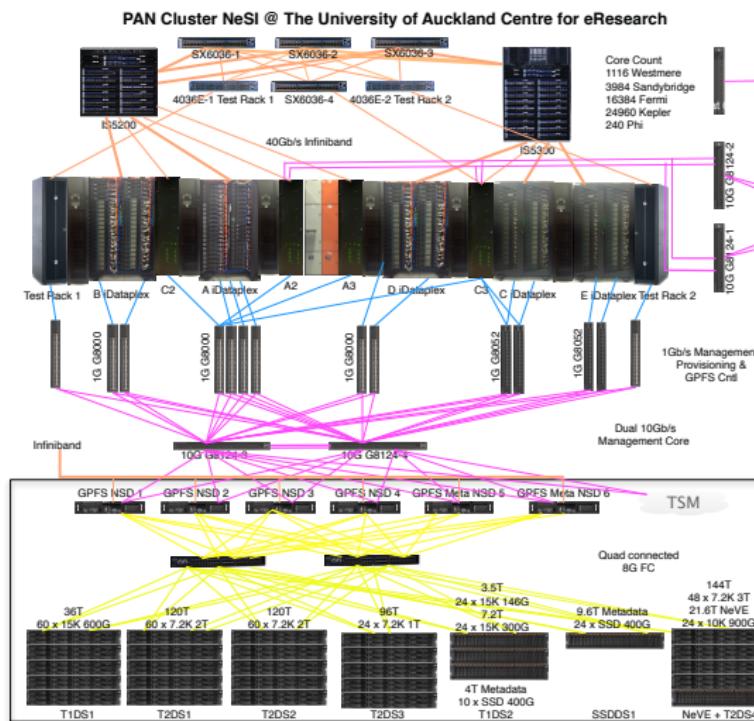
NeSI platforms available for the researchers:

- **BlueFern**: BlueGene/P {8192 cores, 8TB total memory}
- **Fitzroy**: P575/POWER6 {3456 cores, 8TB total memory}
- **BlueFern**: P755/POWER7 {416 cores, 1.5TB total memory}
- **Pan**: x86 cluster {5000 cores, 16TB total memory}

# NeSI platforms

- over **700** registered users
- **700** registered projects
- Universities traditionally have varied job patterns
- Audit databases did not cater for meta information
- Central projects database was not wired to carry quantitative jobs information
- University of Auckland Pan cluster was used for carrying out database test

# Pan cluster



# Pan cluster

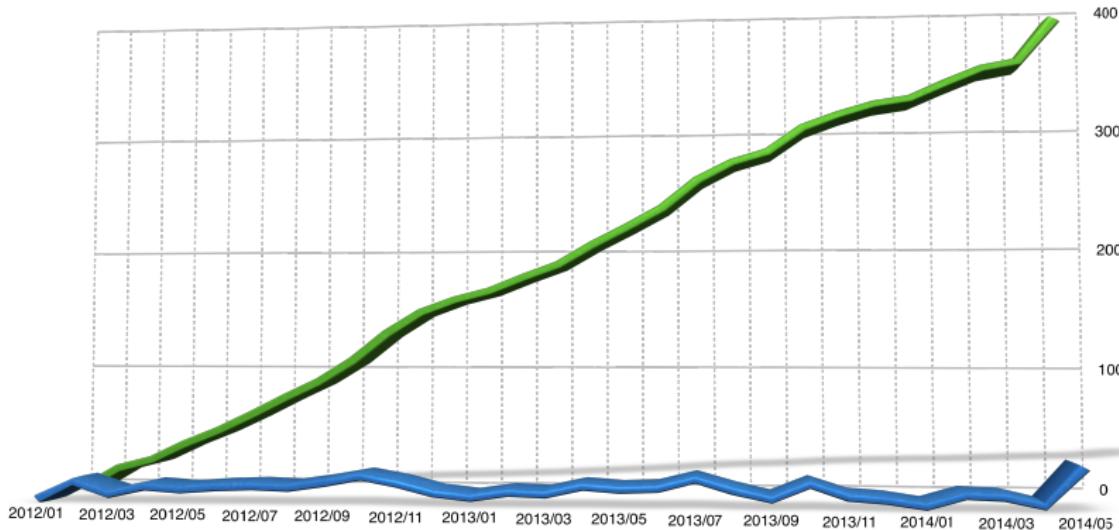
- Conceived in 2010 and launched in January, 2011
- **5,000** x86 cores, **40,000** GPU cores
- **400TB** shared file system, **GPFS®**
- **InfiniBand®**interconnect
- **LoadLeveler®**batch job scheduler, being phased out in favour of **Slurm®**

# Cluster operations

- over **400** registered users
- **300** users submitting jobs on regular basis
- **350** registered projects
- Average daily job throughput: **6000** jobs
- Average daily compute time: **120,000** core hours

# Users intake

— Number of new users      — Accumulated number of users



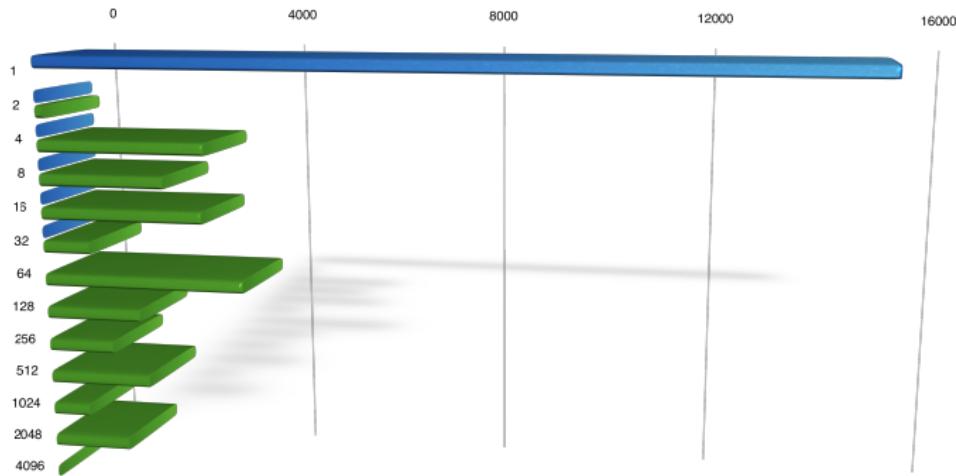
# Users intake

- Users are counted from the time they submit their first job
- Users are associated with affiliation metadata - we can follow institutional use
- Each user participates in one or more projects
- We did **not** know what science domains their work related to

# Jobs structure

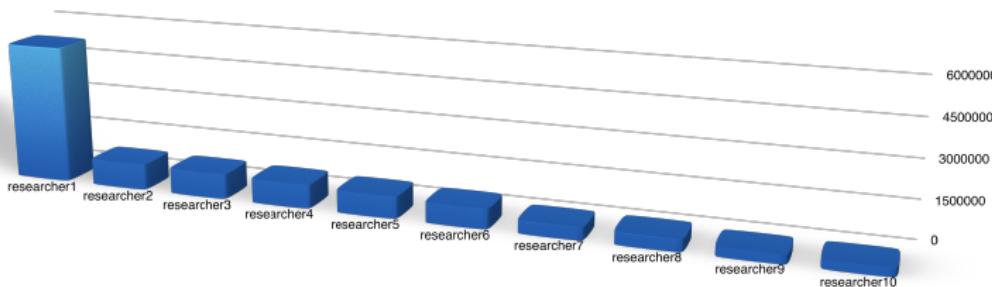
Job distribution by the number of cores, thousand core hours

■ Core time serial, x1000      ■ Core time parallel, x1000

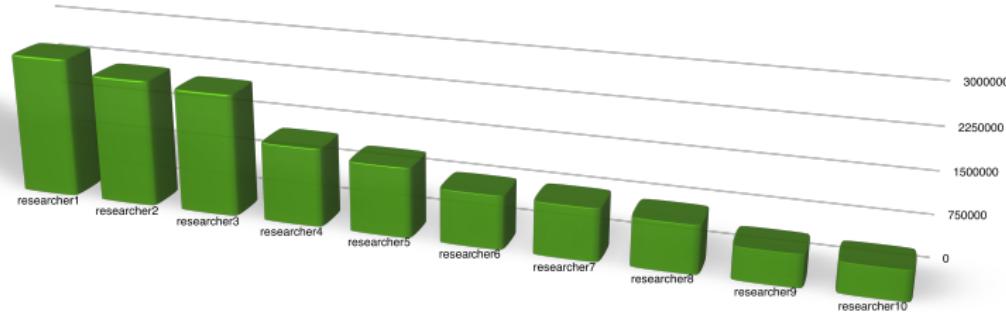


# Jobs structure

Serial core hours, top 10 users



Parallel core hours, top 10 users

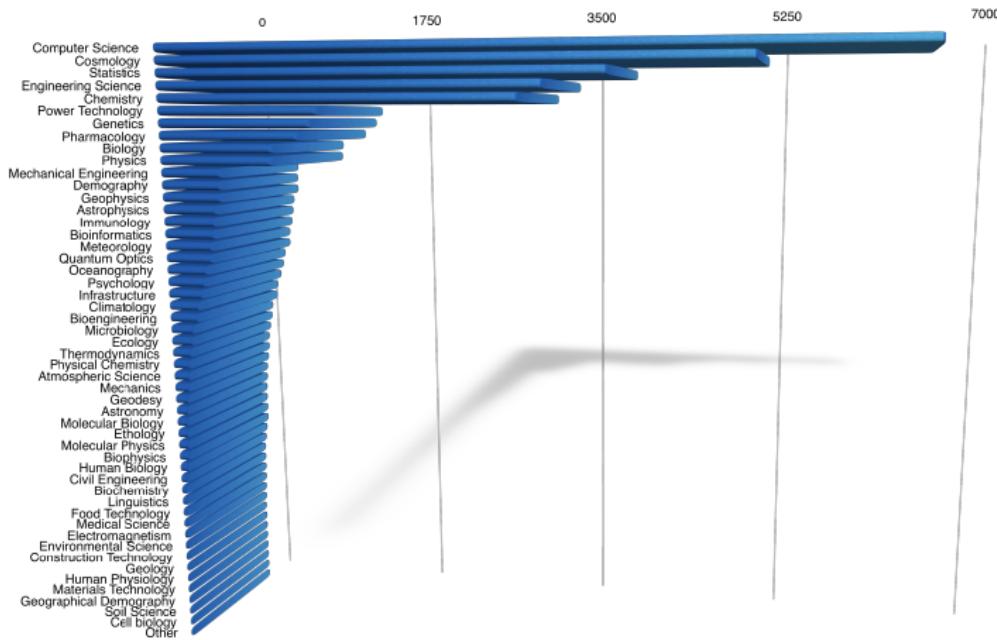


# Jobs structure

- The majority of the jobs were serial jobs
- There are users capable of utilising multiple nodes in a single job
- We knew who were our largest users
- We did **not** know what science domains their work related to

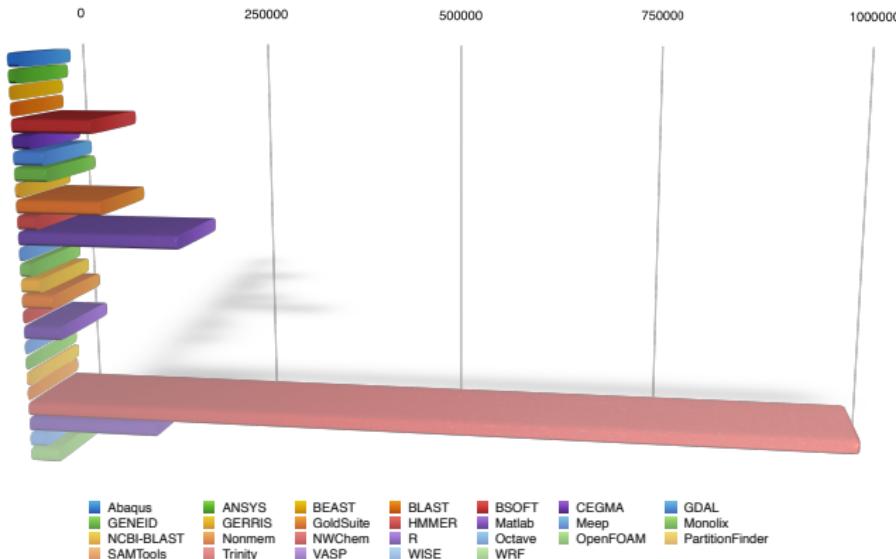
# Science domains spread, Pan cluster

Job distribution by the field of study, thousand core hours



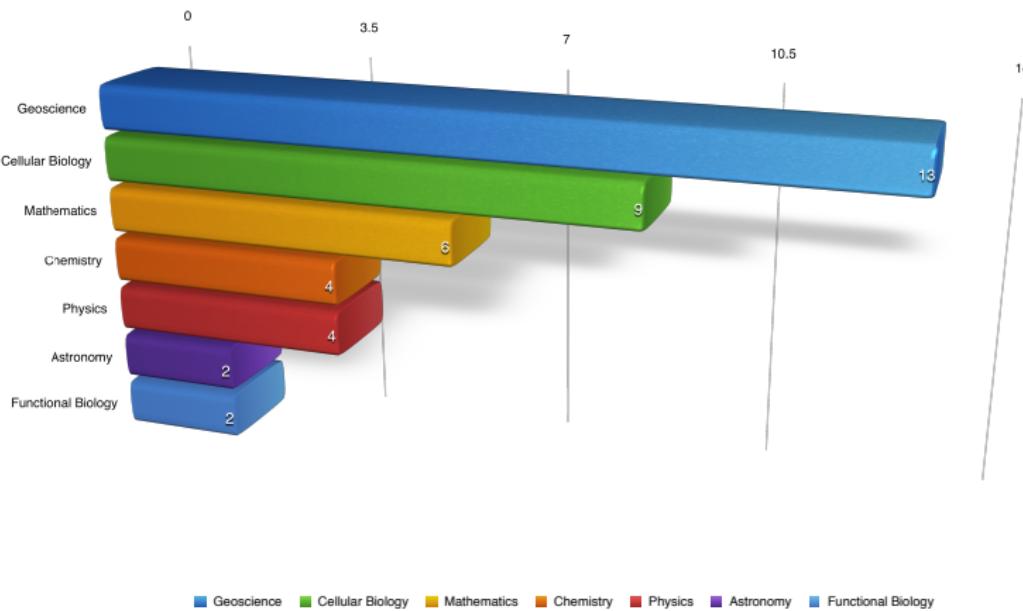
# Applications spread, Pan cluster

Number of jobs per application



# Science domains spread, BlueFern

Number of projects in science domains, BlueFern



# Conclusion

So, why do we need this science domain information? There is a multitude of reasons and many of them have significant practical benefit:

- We can better understand what sciences we enable
- We can prioritize job scheduling based on the science needs
- We can predict applications usage and optimize packages placement
- We can plan for the future with regard to computational resources and support component



# Questions & Answers

