Introducing a new HPC service to non-traditional users.

Guy Robinson, *HPC Service Manager*

LSE Information Management and Technology

lse.ac.uk/imt

Overview.

- Background and history.
  - Why HPC at LSE, expectations.
- Experiences so far.
  - What has worked/not worked.
- Going forwards.
  - Some plans and ideas.

What is LSE?

- London School of Economics and Political Science.
  - ‘One of foremost Social Science Universities in the world’.
    - Includes economics, politics, law, sociology, anthropology, accounting, finance….
  - Provides both teaching and research.
  - Plan to support ‘pathbreaking research’.

State of Computing at LSE.

- Each to his own.
  - Local desktops/laptops, some servers, CLOUD.
    - Independently owned and operated.
      - Self taught community.
    - ABACUS Windows service alongside LSE SAN.
    - Applications: R, STATA, MATLAB, PYTHON, databases,
      - Own or campus licences.
    - CLOUD to supply peak load.

Why Interested in HPC?

- What this is expected to achieve?
  - Research Outcomes.
    - Grants, Prestige Research(ers).
    - Consideration of peers activities.
    - Growth strategy for future research.
- An understanding of what it will take?
  - Developing key skills.
  - Establish significant research programs.
  - Support and development required.

What is Fabian?

- **Collection of Computing Resources.**
  - ***Including applications, data and support.***
- Compute.
  - 15 compute nodes.
    - 24 cores ~~and 128GByte each~~.
      - 2x256GByte, 9x128GByte, 4x64GByte.
    - Soon to Add
      - 2 nodes, 24 cores, 1TByte memory.
      - Database as a Service(DaaS): 2 nodes with software stack.
  - Login node.
    - Your interface with the system.
      - Access nodes, storage, queues etc.
      - Don’t run big/intensive tasks here.

What is Fabian?

- Storage. (*96TByte in total*.)
  - **STORAGE ON FABIAN IS NOT BACKED UP.**
  - /users, ~16TBytes.
    - Space for small files/scripts/small results+data.
  - Other spaces.
    - Add specific data spaces with specific purposes.
      - Workspace for projects to share. /projects
      - Common resources. /databases
    - /tmp spaces for workfiles.
      - Offering higher performance for IO tasks.
  - Other file systems.
    - Mounted filesystems from other places.
      - LSE SAN, sharing services.

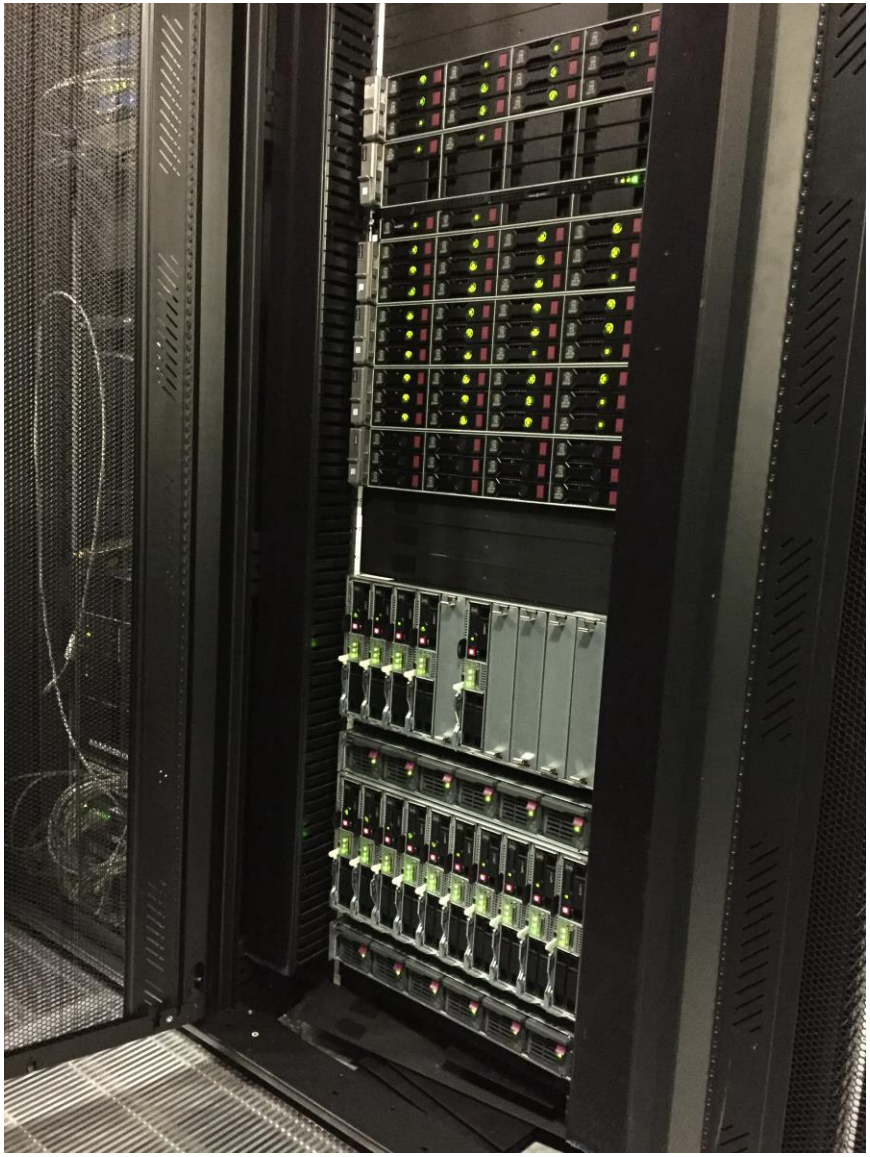
What is Fabian?

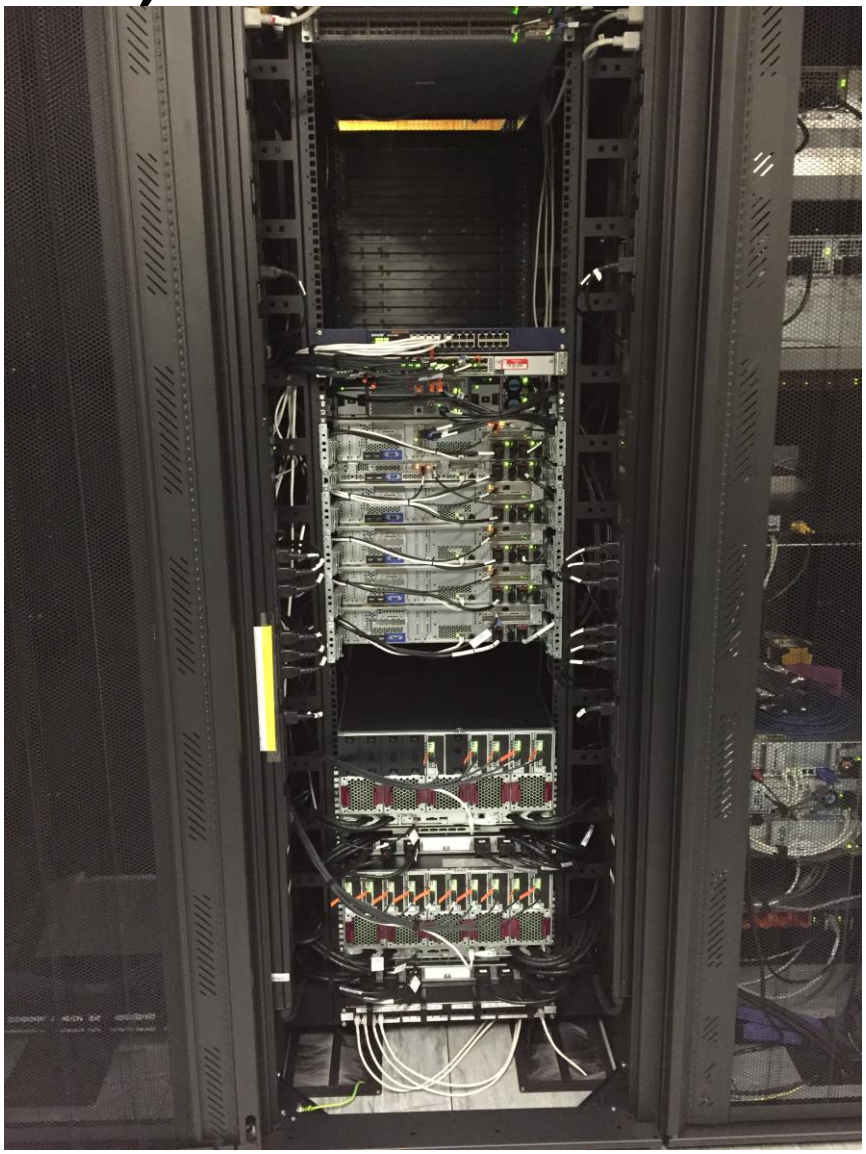
- Software.
  - Runs unix/linux operating system.
  - Openstack.
  - Applications.
    - Installed and tuned.
    - Licences,
      - often campus wide/specific to HPC systems.
  - Tools.
    - Data management/debugging/tuning/libraries.

What is Fabian?

- Connectivity.
  - Networking
    - Local to LSE systems across campus.
    - Academic networking.
      - JANET
      - National/International.
  - Data Cloud.

Fabian Hardware, 1H2016.





Preparation/Early Days.

Operations.

- Operations.
  - Managed service.
    - Setup initially and then maintained.
  - Integrated with existing systems.
    - Shared filesystems, SAN.
    - Common IDs as per other LSE services.
  - Multiple environments.
  - No full time support team.

User.

- Expected users to move across.
  - Installed common software items.
  - Generic HPC training.
  - User expected to take advantage of system themselves.

Some of the Changes.

User point of view.

- Unix working environment.
  - Queue/sharing.
  - Specify resources(know them).
- Policies and procedures.
- Multiple filesystems.
  - Each with a purpose/feature.
  - Workspaces NOT backed up.
- Best practices.
  - Still evolving/realising benefit.
- Testing/Validation.
  - Differences in results.
  - Working timescales for projects.

IT point of view.

- Working alongside users, not just answering questions.
  - A complex and varied environment.
  - It isn't just the app, it is how it is being applied.
    - Science app v productivity apps.
    - Need to steer users to a solution.
- Poorly defined and evolving requirements. (It's RESEARCH!)
  - Gathering performance/activity metrics.
    - Benchmarking, testing, change management.
- Policies and procedures.

Achieved So Far. (4 months)

- Build up a HPC team to support.
  - 3 people, manager, systems, support.
    - Engage with the users.
    - Manage expectations of the sponsors/managements.
- Develop a realistic timeline of adoption.
  - Workshops to engage.
  - Promotion on campus.
  - Peer driven user cases and success stories.

Current status.

- Inhouse HPC.
  - Cluster, support team, early users.
    - An environment and key applications.
    - Integration with the LSE SAN fabric.
  - Workshops, introduction and migration.
    - Web site with simple getting started.
    - Surgery where we 121.
- Growing gradually to reflect users needs.

Understanding the Users.

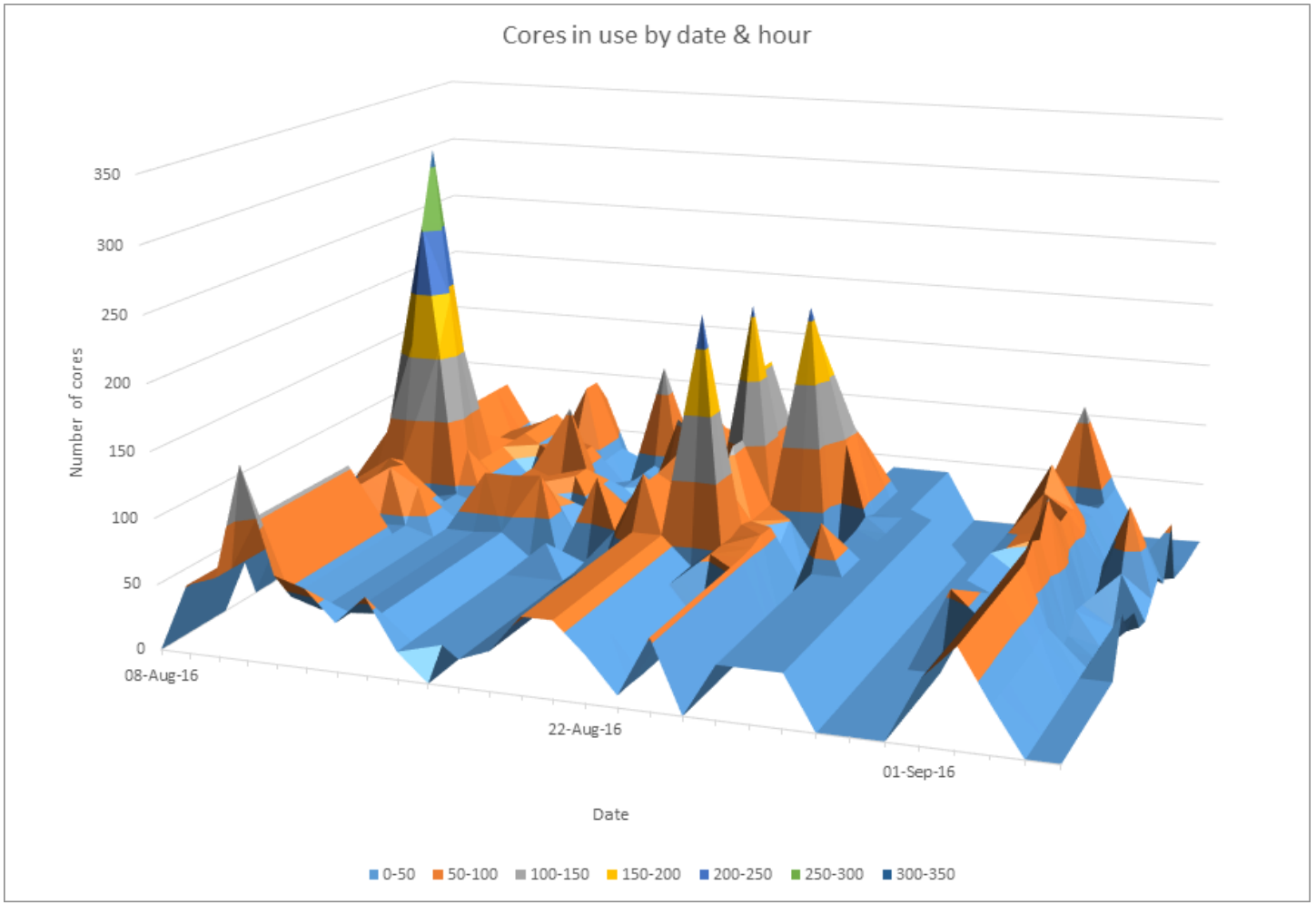
Use Cases.

- How are systems used.
  - Single jobs/single core.
    - Bigger datasets, more memory.
    - Running a lot of these.
    - Manually, simple script, task farm.
  - Parallelism within application.
    - Automatic, directive driven.
    - Simple modifications to programs.
  - Significant fraction of work interactive.
  - Only some of which can be scripted.

What Applications.

- R
- STATA MP2 campus licence.
- STATA MP24, HPC licence
- MATLAB
- PYTHON
- Databases
- SAS,
- Not much code in C.

Gathered Metrics.



Experience So Far.

User Side.

- Building a community.
  - Traditionally working on own.
  - Working together on a problem.
    - ‘Early Career’ quicker to adopt.
    - ‘troubleshoot’ problems.
  - Lack of standards/common practices.
  - Big ‘humps’ to get over.

IT side.

- Initially HPC was a managed service.
  - Now taken in house.
  - Working together on a problem.
- Integration with existing services.
  - Plans to close current service.
  - Large scale shared SAN single point of storage.
  - Many see this support level and engagement as way forwards.

IT and HPC.

- Bit of a culture divide.
  - Many keen to engage.
    - IT see this as being way industry is going.
      - Looking after users and services rather than providing systems.
  - Balance can be different across campus.
  - Convince management at all levels.

HPC better understood now.

Is NOT!

- Putting all servers/services in one management system.
  - This is the CLOUD.
- Creating a bespoke environment for each individual user.
  - Infinite number of modules, VMs, module, filespaces...
- Not working with the user to help them improve things.

Is!

- A jigsaw of many different aspects working together to be greater than the simple sum of the parts.
  - Continual Advancement.
  - Adopting/developing new techniques and working practices.
- Working with the user to help them improve things.
  - Sometimes requires an understanding of their methods.

Other Challenges.

Engagement.

- Challenging.
  - Not a history of academic support at the level of HPC has with STEM.
  - Self taught/pass on skills/experiences.
- Solutions(over time).
  - Projects/programs to work together.
    - Wider peers. HPC+research.
  - Funding/motivations/SRE role.
  - Moodle within the workshops/surgery
    - Get users talking/sharing.
    - Newsletter.

Security.

- A lot of the data is 'personal'.
  - ISO27001.
  - specific rules from provider.
    - Often different.
- Methods adopted aren't good for a shared/common resource.
  - Individual machines for each dataset/project.
  - Separate USB disks.
    - Limiting factor in transfers.
  - Data encrypted when ‘moved’.
    - HPC moves data more!!!!!!

Where are we going?

- Research Computing rather than HPC.
  - Broader than performance.
    - Methods of working, from algorithm to best practice.
    - Develop a solution, from system to applications.
  - Data management.
    - Engage with library over shared datasets.
  - Provide a robust, sustained research environment.
    - Across generations of research…

Compare with STEM.

- The non-traditionals aren't that different.
  - Earlier stage in the process.
    - Define HPC/Research Computing.
    - Ran own systems and developed code.
    - We are at leading edge (of user).
    - Lessons (re-)learnt, the hard way?
      - Established/ing (selected) best practices.
    - Building balanced collaborations.
      - Skills and logical processes of STEM.
      - Many who work in HPC come from STEM.

Technical Vision.

- Variety of Resources.
  - IN house HPC/Research Computing.
    - Systems, people, environment.
      - Or People, environment, systems.
    - A stepping stone to/from other services.
      - Laptop <> Fabian <> CLOUD (?)
  - User Communities.
    - Driven by SRE skillset.
      - Scripting, algorithms, data management, …
    - Apply experience not duplicate systems.

Bigger Picture.

- Build a community for this research area.
  - Adopt/define best/common practices.
    - Standards, driven by need to share data/results.
    - SRE and HPC support engagement.
  - Dream of bigger things.
    - HPC-STEM as an example.
      - Take the relevant experience/mistakes.
    - HPC-SIG, previous experiences…
  - Motivation.
    - Closing previous service.

Conclusion.

- Idea of the journey so far.
  - Expectations, achievements, issues...
- Where are we now(?)
  - Setting up an embryonic service.
- Where we are going?
  - Our plans to build service with users.
  - Build useful collaborations.

Overview.

- Background and history.
  - Why HPC at LSE, expectations.
- Experiences so far.
  - What has worked/not worked.
- Going forwards.
  - Some plans and ideas.
    - Comments welcome.

Overview.

- What does LSE do?
- Number of students/researchers.
- Mission statement, roles,
- Summary slide from jess.

From small

- BHPC(before HPC).
- Own computers, laptops, desksides, ‘high-end’ servers.
- Cloud
- Independently run.
- ABACUS, windows farm. Specs.

BHPC(Before HPC).

- Resources.
  - Own systems.
    - Laptops, desktops, dept. servers.
    - Independently sources/managed.
  - ABACUS Service.
    - Windows server farm.
    - SAN storage.
  - Cloud.
    - Experts/Independent takeup.
- Support.
  - Self/community taught.
- Research Undertaken.
  - Large scale data analysis.
  - Simulations.
  - Majority write scripts
    - A few write code, C

AHPC(After HPC)/Vision.

- Resources.
  - Inhouse systems.
  - Peers have significant resources.
  - Campus licences
  - Engagement with providers.
  - Coordinate cloud presence.
  - Transition work between systems without significant challenges.
- Support.
  - Experts in.
    - Application.
    - Use of Computers.
    - Library data.
      - Management and collections.
- Research Impacts.
  - Significant Grants.
  - Collaborative Research.
  - Attract Faculty.

Now.

- Resources.
  - Fabian.
- Support.
  - Small team of three.
    - HPC System Support.
    - HPC Applications Support.
    - HPC Service Manager.
- Research.

Fabian.

Strategy to grow users.

- HPC Users.
  - Early Adopters.
  - ABACUS migration.
  - Case studies, as motivation.
  - Simple environments to foster movement between systems.

Multiple Environments.

- Fabian supports.
  - HPC ‘linux’. New to many.
  - Openstack. Familiar to many.
- Users know.
  - Windows. Own systems and ABACUS.
  - Cloud.
    - AWS offerings, own configurations.

How do we steer users to better environments.

- We want them to be able to move to/from LSE to other locations.
- Build up cases
- Build up community preference.
- Develop understanding of how much working environment plays in smooth research working.

Environments plus/minus.

Windows.

- Know it.
- Desktop and @HOME.
- Campus support.
- Established, especially workflows.
- Performance and scaling?
- Applications/Community codes.

Linux/cloud

- Strange
- Different desktop and not @HOME.
- eScience tools.
- Performance.
- Established HPC.
- Community codes.

Build the community.

Own training.

- Classical offering.
  - Introduction.
  - Migration.
- Surgery.
  - Building a culture of asking for help.
- Web/Moodle.
  - Campus presence and culture.
  - Need to ‘seed’.

Campus training.

- In departments. 121/surgery
  - In classes alongside methods or first use of applications.
- Summer school,
  - especially a methodology section.

Key Challenges.

- Multiple Environments
- Various skills.
- HPC culture.
- HPC as a service.

Is this HPC?

- Better called Research Computing?
  - Data centric/intensive.
  - Methods of computing above performance.
  - What metrics identify the difference?
  - Multiple component.
  - Collaborative/sharing.

Data focus.

- Storing the users data.
  - ‘eScience’ aspects.
    - Portals/environments to compute consistently.
    - Best practices and perceived values.
- Enabling sharing.
  - Data registration as part of publications.
  - Data sharing with collaborations.
  - Data security,

Are ‘successes’ really the role model?

- Select the lessons to learn.
- Let them make their own, small scale mistakes.
- How to present the lessons so they are taken up?
- What were our mistakes/lesson learnt?
- How could we have avoided them at the time?

Making the most of an opportunity!

- Very short windows to engage.
- The hump can be difficult!
- Change needs an observed reward.
- Long term view.

Added Value of HPC team.

- Apply experience.
- Configure system to promote best practices.
- Encourage and grow.
- Hide things from the users…..
- Simple scripts/wrappers/
- Solve problems

Grow a community.

- Develop expertise
  - Over time.
  - Right steps at the right time.
  - Prepare for changes.
    - Growing HPC
    - Trends in HPC.
  - Researchers/students retain a focus on research.
  - Enabler of a community.
    - Established standards, and standards needed.

Maintaining a Momentum.

- How to keep advancing HPC in this community.
  - Growth of the problems.
  - Bigger datasets, breath/detail/
  - Complexity of theories.
  - *Cost savings.*
- Who are the drivers?
  - Researchers/research councils/peers/'experts'

Use of the cloud.

- As a data resource.

Direct v Indirect.

- Direct view.
  - Users, researchers and students.
- Indirect view.
  - IT managers.
  - Departmental Managers/University Leadership.
- Middle Ground.
  - HPC support team.

Fabian HPC Service.

- Hardware

Changes we have made.

- Storage.
  - Initially one big filesystem.
  - Users, projects, datasets.
  - Clearly defined as NOT BACKED UP.
- Queues.
  - Coping with a mixture of workloads.
- Users resource requests.
  - Bigger memories. 1TByte nodes.
  - Specifically configured and integrated Database nodes.
  - Larger licences, 2 to 24 cores for STATA.

What has been useful.

- Applications having ‘peer written’ how-to guides. (For us and users☺.)
  - MATLAB, parfor
  - STATA, scaling of library routines.
  - R differences between parallelism for windows and linux.
  - PYTHON.A local user group.

    - *Issues with each having a centre/system configuration part.*

What is a non-traditional user.

- Use applications.
  - At LSE, STATA, R, PYTHON, MATLAB.
- Own resources.
  - Buy bits.
    - Laptop, desktop, deskside, server.
    - Storage, USB disks, more USB disks, YAUSBD….
    - Software, individual licences.

What comes as a shock!

- The working environment.
  - Linux, unix, ‘typing’.
  - Sharing.
  - Scripting.
  - Batch.

HPC Practices.

- We don’t backup working disks.
- We test things.
- We upgrade things.
  - That why we need to test.
- We keep expanding/growing.
- We need policies and procedures.

IT department and HPC.

- IT departments provide services.
- Is HPC a service.
  - Some aspects are.
  - Some aren’t?

IT and HPC

- What is?
  - Admin tasks
  - Basic skills training
- What isn't?
  - Help programming
  - Help debugging
  - Help optimising
  - Help with applications.
  - Help with working better.

Core Concept of HPC

- Computing System.
  - Processors +Memory.
  - Interconnect.
  - Storage
    - Fast parallel/local.
    - Working spaces
    - Archive.
  - Software stack.
  - Configuration and Operation.
- But it is more.

New Ideas.

How long will it take.

- Community has to learn lessons.
- We have to teach them.
- Are there shortcuts.
- Is there an optimial route.
- Not overnight,

What are the good things to take?

- Computational portals.
- Training
- Already parallel applications.
- Best practices.

- In all these be selective?!

What pitfalls to avoid.

- Windows.
- Over keen vendors.
- Managed services.
- Tell them it is simple.

Shocks to the community.

- Costs,
  - Hardware, performance costs……
  - Human costs, hidden support
    - Students DIY

Security in a shared environment.

- ‘Each project has a machine.’
- ISO27001.
- Very important to the users.
- Making LSE policy HPC compatible.

Different Nature of Computing.

- Man and machine.
  - Working together in perfect harmony….
  - Computing is a lot more interactive.
  - Still have things that take days…….

Shocks coming.

- Reproducibility.
- Meta studies.
  - Where work in only on data from others computstionsa.
- Greater sharing/collaboration.
- Data as a research outcome.

Finding Out what the users are doing.

- Gathering statistics.

Providing a system of opportunity.

- Not aiming at a high utilisation.
- A happy user ☺
- Educating folks this is what it delivers.
  - In particular the support contribution.
  - Need to be prepared to make this work.

Focus.

- What has impact.
  - As a community.
  - As individuals.
    - Can we make this scalable.

The team.

Current Activity.

Future Expectations.

How to achieve this?!

History.

Differences to

Challenge of the cloud.

Benefit of the cloud.

- Lots of resources.
- Can ‘burst’ out to it as required.
- Cheap.

Impact of the cloud.

- ‘Burst’ out as required.
- Single node still suffices.
- Costs are low.
- Once setup…….
- Used by Peers.
- Doesn’t have policies, costs, metrics,
- No ‘real’ support.
- Need to do own development.
- Build it yourself.
- Nobody tells you what to do.
- Just like their current resources.

Building an environment.

- Really it is a culture that is needed.
- A never ending story/process.
- Grab opportunities.
- Don’t leave users behind.

New Trends.

Collaborate

- Lessons of others to speed approach.
  - Get the basics covered.
  - Can we build community faster.

Outcomes.

- Key research component.

Thoughts/Conclusions.

- Lots of hidden ‘expertise’ in ‘established’ HPC Projects.
- Big step for a new community to HPC, in the classical, or STEM, sense.
- Biggest danger/risk/issue.
  - Build high expectations/overhyping.
  - Making HPC seem simple/cheap/quick.
  - Not building the necessary commitments.