

Data Aware Scheduling: Taking the Compute To the Data

Peter Clapham

Wellcome Trust Sanger Institute

Overview

- Background and general Sanger overview
Is this really sustainable ?
- Data Aware Scheduling revisited.
How do we incorporate this into an existing world of pipelines and user practices ?
- Platform LSF and Sanger
A simple plugin model
- Test cases and feedback to date
- The future...
- And thanks

About the Institute

- Funded by Wellcome Trust.
 - 2nd largest research charity in the world.
 - ~700 employees.
- Large scale genomic research.
 - Sequenced 1/3 of the human genome (largest single contributor).
 - We have active cancer, malaria, pathogen and genomic variation studies.
- All data is made publicly available.
 - Websites, ftp, direct database. access, programmatic APIs.



Feature: Great expectations: human genome research

1 June 2010



June 2010 marks a decade since the draft human genome sequence was announced. How much of an impact has this development had on genetics, genomics and science in general? What are the ethical and legal issues arising from this work? Chrissie Giles spoke to people who worked on the project then, and those who use the sequence and subsequent research

in their work today, to find out.

Professor Sir John Sulston: "The key was tackling the whole genome"



John Sulston is among the few pioneers who, in the 1980s, began to develop the field recognised today as genomics. He and colleagues began a project to sequence the genome of the nematode, *Caenorhabditis elegans*, in 1990 - the success of which fed directly into the plans to sequence the human genome and set up the Sanger Centre (now the Wellcome Trust Sanger

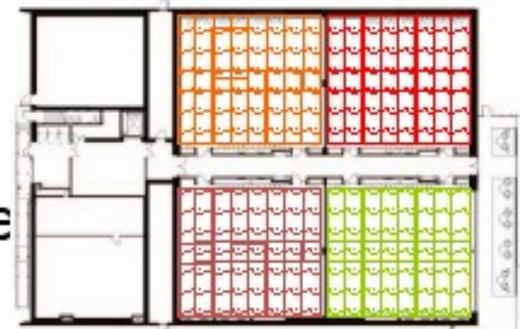
Institute) near Cambridge.

"I was very committed to the whole genome approach. My philosophy, in a Douglas Adams kind of way, was that this thing is big, it's seriously big. You're not going to solve life by looking at a handful of genes."

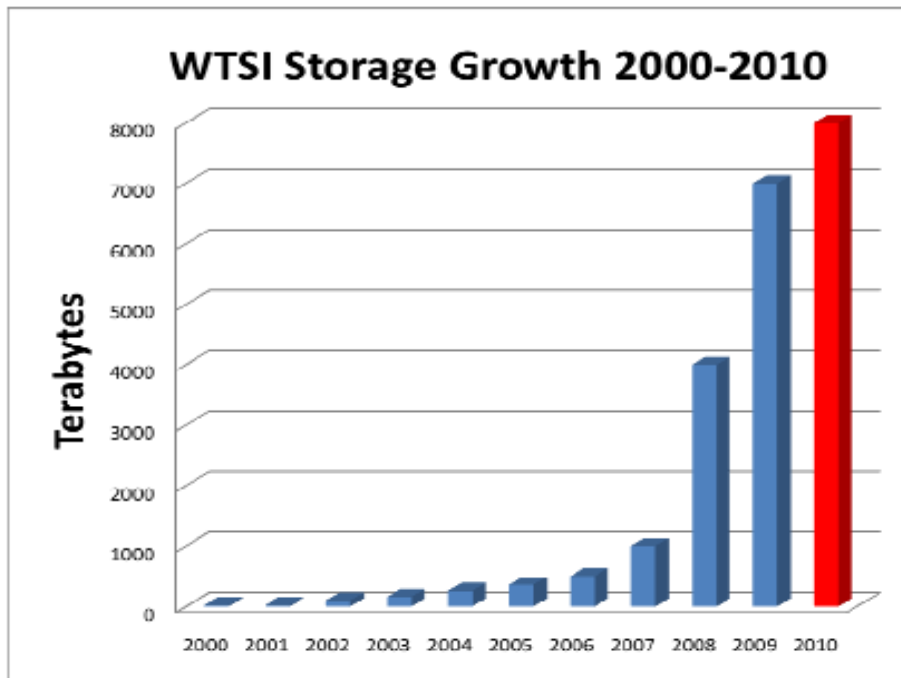
<http://www.wellcome.ac.uk/News/2010/Features/WTX059943.htm>

The Wellcome Trust Sanger Institute Data Centre

- 1,000 m² of data centre space (4 x 250m²)
- Hosts Sanger and EBI's compute and storage
- Combined totals > 13,000 cores of compute
> 13 Petabytes of storage
- Rivals the largest high performance compute facilities in the UK
- Possibly the largest Life Science data centre in Europe



Storage expansion Rate (so far...)



By 2005 we had accrued 300 Terabytes of data
With the introduction of new sequencing technologies this has grown > 20 fold In the last 2-3 years
We now have >8 Petabytes of storage capacity

Classic Sanger “Stealth project” .. Revisited ..

Summer 2007; first early access sequencer.

Not long after:

- “15 sequencers have been ordered. They are arriving in 8 weeks. Can we have some storage and computers?”

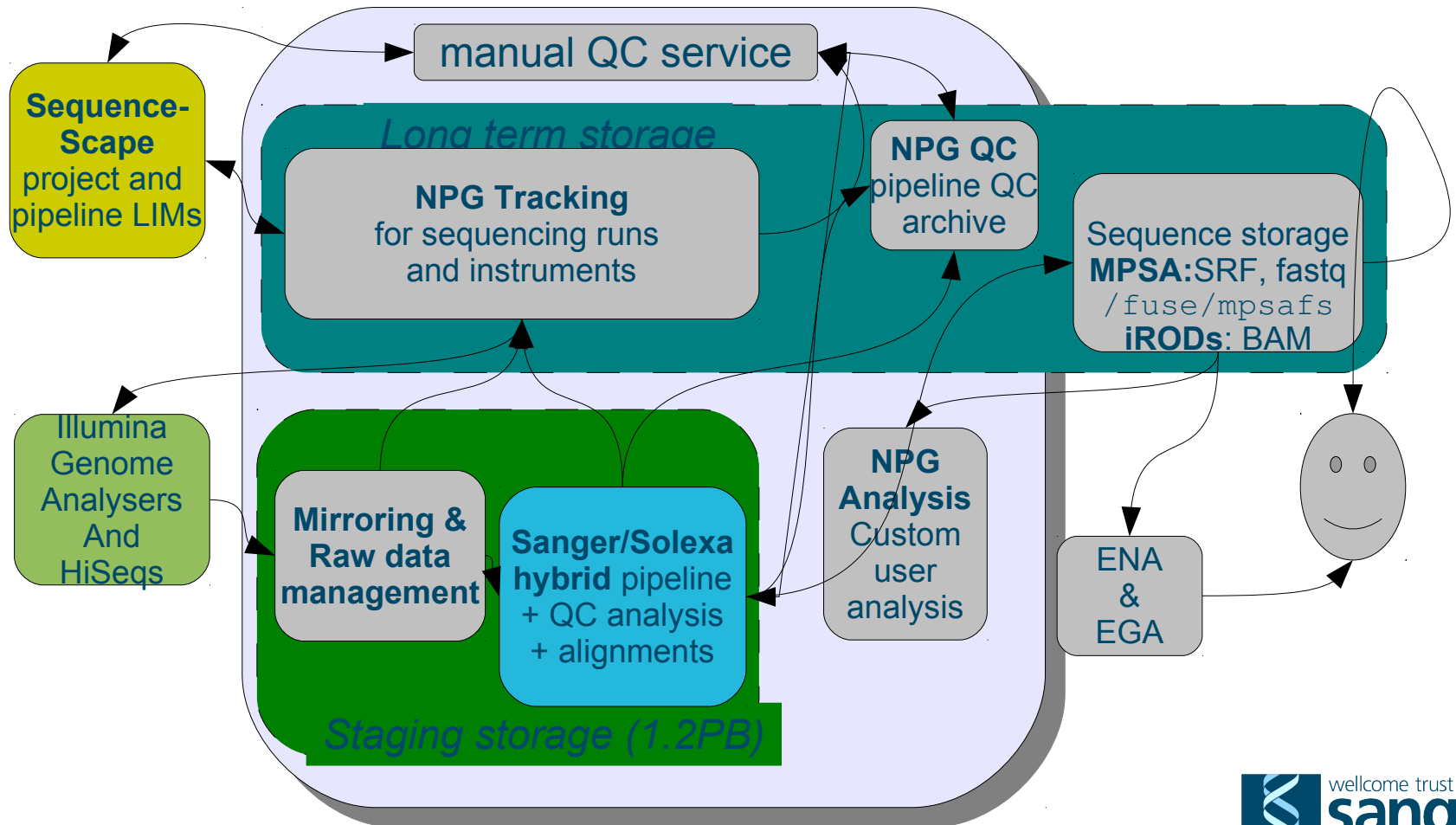
But now in 2010

- We have 40(ish) of these new technology sequencers...**BUT they are all being upgraded to Hi-seqs with 5x the output per unit / unit time**

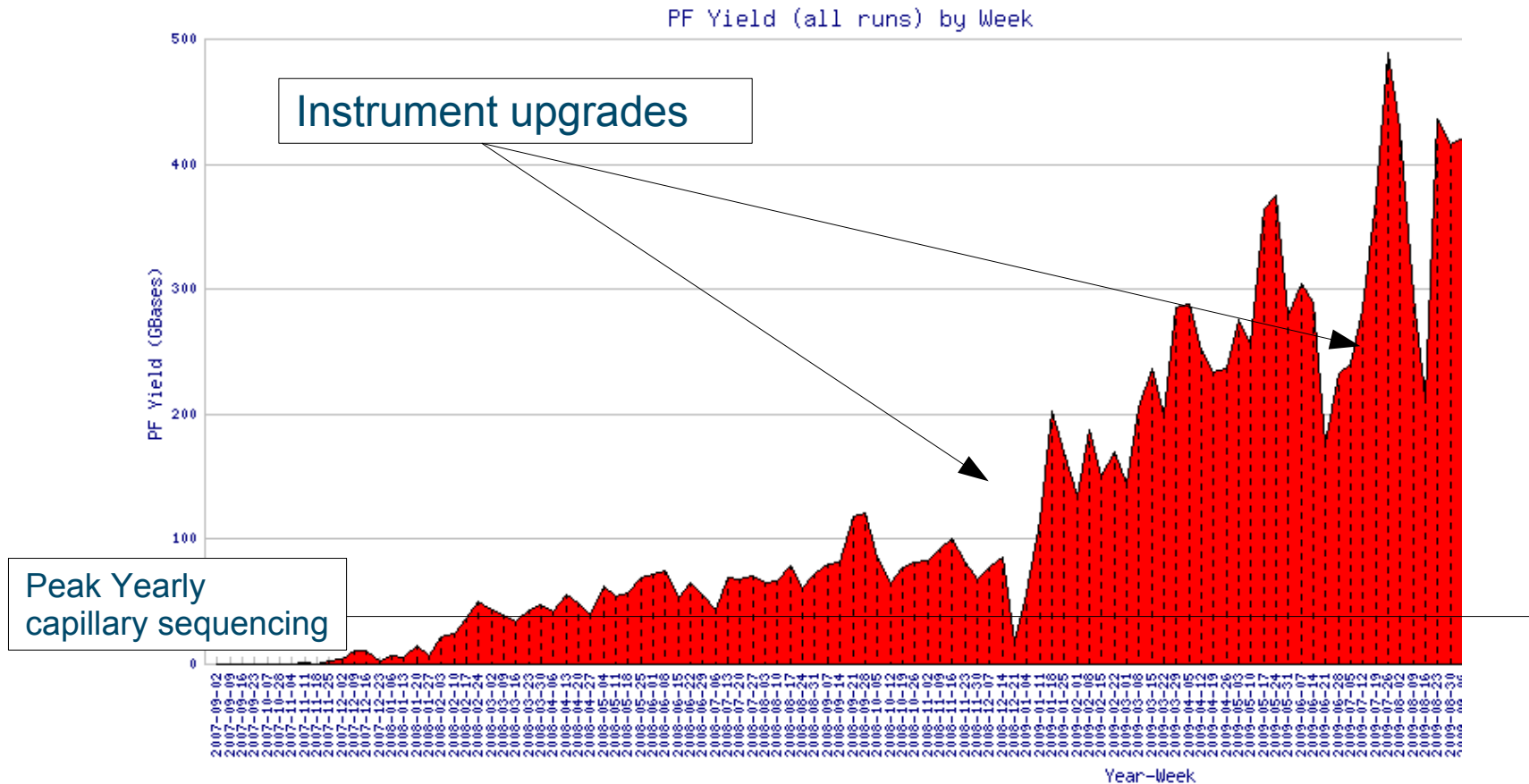
Short on notice but high on demand.



Sequencing pipeline



So the predicted data rate becomes more variable...



The Current Data Growth Prediction is..

46.5 Petabytes by 2016

If current 3.5" 3TB disks are a guide..
Then we'd be looking at 15,500 units

Stacked horizontally, this would be a tower 393 Meters in height

Statue of Liberty, 93 M
Nelson Column, 51.6 M
Eiffel Tower, 324 M

All are significant works of engineering.
This is almost certainly just for usable disk space..
Not raw !

But Sequencing is only the start

The output of the raw data from the sequencers is only the start. Various projects take this as their starting point. All of which have their own significant compute requirements.

So just getting the data in is the easy part (don't quote me on this !). The challenge is how do we architect a compute facility to actually make use of the combined data in a way that allows for cross group co-operation and not blowing away the existing pipeline world ?

Previous Compute configuration.

Separate compute silos for separate groups

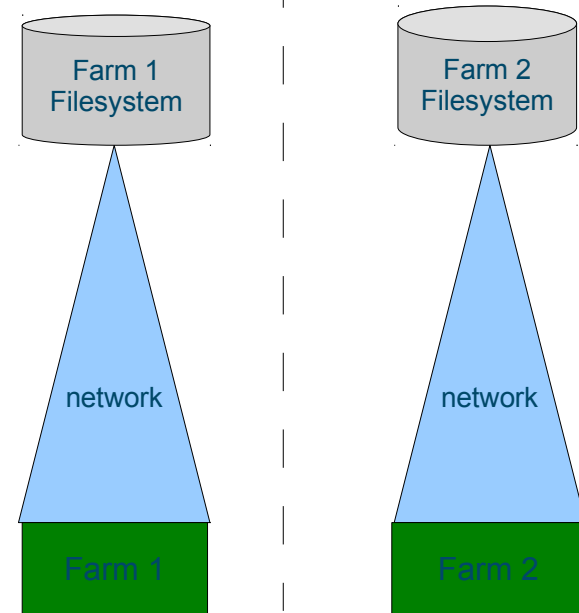
- Eg cancer, pathogen, sequencing

“Fast” cluster storage (lustre) for IO bound work.

- Constrained by network topology/bandwidth.

Problems:

- Projects span the organisation domains.
- Projects can outgrow compute + storage.
- Fast storage is always full.
- Cannot afford to buy more.
- Will we always fit into just one site ?!



The Obvious Slide

Data movement is slow.

For example, DRM's are generally tailored to accommodate the workload by presenting the first available fast node for a given job with the relevant resource requirements... Typically no consideration of data movement is involved.

So for a Nehalem of 3Ghz with a L1 – L3 cache latency of 1-2 ns backs to system memory of latency of 4-6ns latency. Then the network across our datacentre has a 100ms latency. The disks have at best 4.2 ms seek...

So all that fast compute is just waiting to be used and the users are revolting, to turn a phrase.

Clearly this is not sustainable...

Plan B.

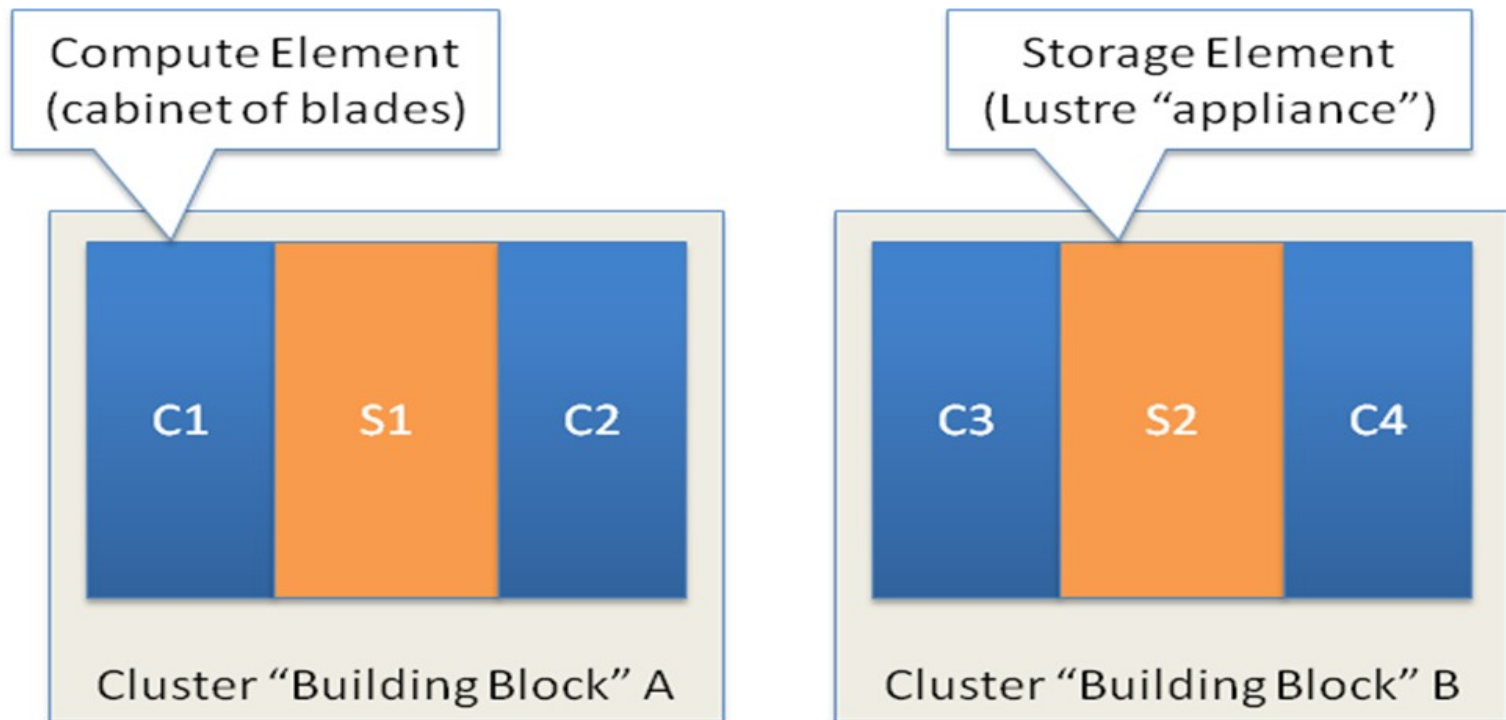
Back to basics

- Blocks of network, compute and storage.
- Assume from day 1 we will be adding more.
- Expand simply by adding more blocks.
- Potential to load balance between storage units and have a clear association between compute and “closest” or preferred storage.

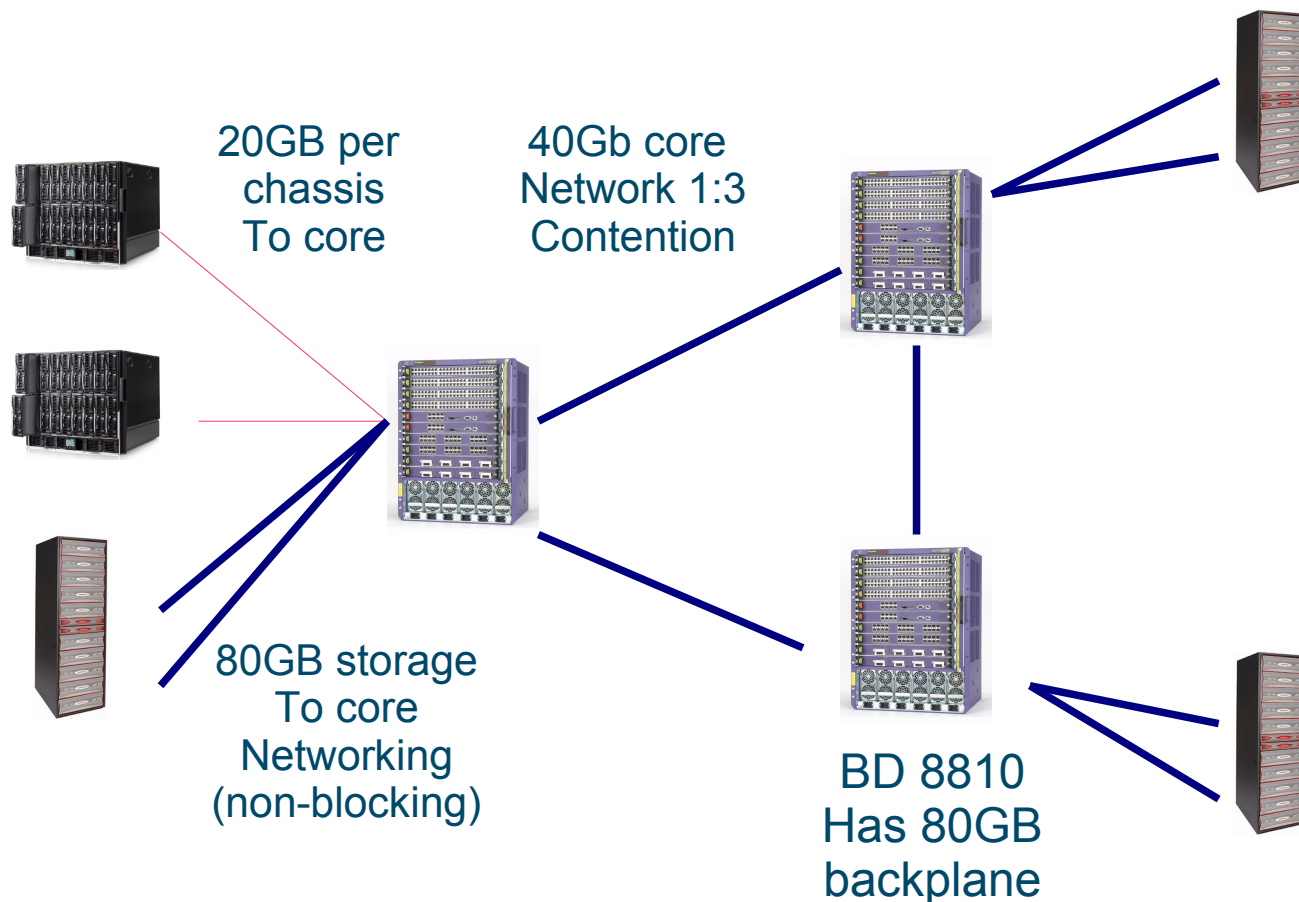


With Modular Compute Comes..

So we now have a server-storage setup which is split between data centre Segments. We can therefore consider our server data as building blocks.



And here's what it looks like.



So if data stays within direct switch environment there is no contention and double the bandwidth. The reduced cross network contention is an added bonus.

Data Aware Scheduling (revisited)

This is not a new idea. For example:

Stork DRM
Glite Grid Infrastructure
Etc etc...

Each involves a significant change in the way things are done locally and they are generally considered unacceptable to our user base.

In short.

Not likely to change the world to solve this issue. But the pain is still there, so what do we do.

Platform LSF and DAS

The Sanger institute has long used LSF for workload management and many pipelines are developed with this in mind.

If this DRM is already managing the workload then surely it's the right place to start in attempting to associate compute with data location. How hard can it be right ?

If we can just manage the basics of the data flow without being too clever we should be able to deal with the majority of our data compute issues.

The lowest hanging fruit philosophy applies.

Big advantage for us, our users are LSF command line savvy, so are not overly concerned by generic LSF flags

Initial DAS objectives

To permit a user via LSF to:

- Be aware of relative “closeness” of a given storage unit with the available compute hosts.
- Submit a job with a scratch space reservation limits, quotas only help space management so much.
- Permit registration within LSF of datasets.
- Submit a subsequent job that can later use the initial data registered on a compute node preferentially associated with the data location.
- View and modify dataset.
- To be simple for the end user to use and follow path of least surprise.

The DAS Plugin

All based on Platform LSF 7.0 Update 6

So how is this achieved ?

- A new datamanager daemon
- We only have 2 new binaries:
 - Datainfo
 - Datactrl
- In the background there's a storage.collect script.
- An updated bsub.
- And a couple of config files to associate storage with compute hosts.

So all fairly minimal impact.
But does it work ?

Server side DAS configuration.

The config file is pretty straight forward and v. flexible taking the format:

```
Begin Storage
Storage           = scratch1
Location          = /warehouse/isg_warehouse/scratch1
Host_Distance    = [bc5_1, 0] [bc6_1,1]
End Storage
```

Where:

Storage is LSF referenced as scratch1

Location... what it says on the tin

bc5_1 and bc6_1 are chassis bc-5-1-* and bc-6-1-*

[bc5_1, 0] is a relative distance of 0 from this storage area (i.e v.close !) but bc6_1 nodes are further away.

Best of all set this to 5 and nothing will use the storage.

We can seamlessly drain storage based jobs for scheduled downtime !

View into LSF view of the storage

Datainfo is a window into LSF's awareness of the storage.

- Options include:
 - Datainfo storage, show current space / load (in future)
 - Datainfo dataset, show which datasets are registered.
- Example output:

```
pc7@oik:~/Documents$ datainfo storage
STORAGE      LOCATION      STATUS      MAX      AVAIL      RSV
scratch3     /warehouse/isg_wh_sc open      500.0G   500.0G     0.0G
scratch2     /warehouse/isg_wh_sc open      500.0G   202.0G   220.0G
scratch1     /warehouse/isg_wh_sc open      500.0G   256.0G   180.0G
```

View into Datasets Registered

Datactrl is the LSF data ctrl (control) bit.

- Options are:

```
pc7@bc-6-1-01:~$ datactrl
```

```
Usage: datactrl [-h] [-V] [command] [command_options] [command_args]
```

where 'command' is:

storage	Open or close the specified storage volume(s)
dataset	Register dataset to specified location or storage volume
	Unregister a specified dataset or all the datasets that are registered by the user from specified location or storage volume
restart	Restart datamanager
help	Get help on command

Using the plugin.

A new LSF resource parameter

```
$LSB_STORAGE_LOCATION
```

Once a job is submitted to LSF this parameter will be set to the storage location itself. So in the case above S1 = /lustre/scratch101, As a job has a data area assigned by the scheduler the following Variable is automatically set:

```
$LSB_STORAGE_LOCATION=/lustre/scratch101
```

A very useful beastie.

Putting this together to do stuff. An artificial test case.

- A test case:

To submit a local benchmark suite as an LSF job array with a scratch area requirement of 40GB per job. The sort of thing our users do daily.

```
bsub -o $HOME/DAS/bench.%l.out -n 2 -J"bench[1-20]" -R"span[hosts=1]"  
-extsched "storage[size=40;distance=0]" $HOME/DAS/das-bench.sh
```

Where:

-o directs LSF output to a file

%l is the array index, no naughty writing all output to the the same file !

-n2 2 cores per job

-J"bench is the name and we're firing off 20 jobs, indexed 1 - 20

-extsched is the new magic
It's asking for 40GB directly associated with the
node the job lands on.

`$HOME/DAS/das-bench.sh` is the script to run

\$LSB_Storage

That useful \$LSB_STORAGE_LOCATION
Within the das-bench.sh we have:

```
echo "Storage resource - $LSB_STORAGE_LOCATION"  
  
tarball="/lustre/scratch101/sanger/pc7/benchmark.tar.gz"  
testdir="$LSB_STORAGE_LOCATION/users/$USER/bench.$LSB_JOBINDEX"
```

So no matter which node the job lands on it can parse the data directory of choice, use it and pass this on to subsequent scripts etc etc.

DAS and Sanger Using the plugin

In the test case here:

bc5_1 nodes are closely associated with Scratch1 storage
bc6_1 are associated with Scratch2 storage.

The storage has a maximum of 500 GB so. If this works then we should expect to see jobs arriving on both sets of blades as the space reservation etc sets in.

DAS and Sanger Using the plugin

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-6-1-09	bench[1]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-5-1-01	bench[2]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-5-1-02	bench[3]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-5-1-03	bench[4]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-5-1-04	bench[5]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-5-1-07	bench[6]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-5-1-08	bench[7]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-6-1-10	bench[8]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-6-1-12	bench[9]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-6-1-13	bench[10]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-6-1-14	bench[11]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-5-1-10	bench[12]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-5-1-12	bench[13]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-5-1-13	bench[14]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-5-1-14	bench[15]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-6-1-03	bench[16]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-6-1-04	bench[17]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-6-1-07	bench[18]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-5-1-05	bench[19]	Jul 14 12:12
643320	pc7	RUN	das_test	bc-5-1-09	2*bc-6-1-06	bench[20]	Jul 14 12:12

DAS and Sanger Using the plugin

The script also added a little something to register the datasets within LSF in case of subsequent need.

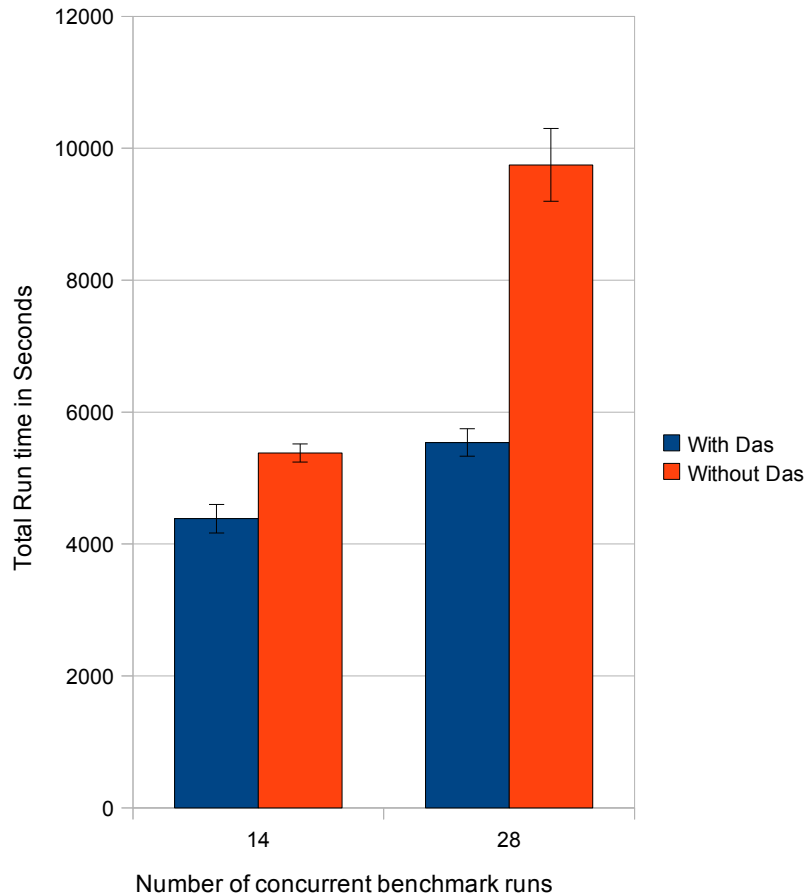
```
datactrl dataset reg -k $LSB_STORAGE_LOCATION bench.$LSB_JOBINDEX
```

So the end result is datainfo dataset now gives:

```
pc7@bc-5-1-09:~/DAS$ datainfo dataset
DATASET      STORAGE      LOCATION      USER      REG_TIME
bench.16     scratch2     /warehouse/isg_wh_sc pc7       Jul 14 12:18
bench.17     scratch2     /warehouse/isg_wh_sc pc7       Jul 14 12:19
bench.20     scratch2     /warehouse/isg_wh_sc pc7       Jul 14 12:19
bench.18     scratch2     /warehouse/isg_wh_sc pc7       Jul 14 12:20
bench.4      scratch1     /warehouse/isg_wh_sc pc7       Jul 14 12:22
bench.5      scratch1     /warehouse/isg_wh_sc pc7       Jul 14 12:22
bench.10     scratch2     .....
```

So those jobs that landed on bc-5-* really did use scratch1 and those which landed on bc-6-* used scratch2. Looking useful.

But was it really useful ?



Even at **very** small scale testing it's clear that allowing LSF to take care of the very basic storage allocation works.

Yes users could do this manually BUT where we have pipelines of job arrays in the 1000's – 10,000's this won't happen without this.

How much would it cost to purchase 2x the performance from other sources ? Cores CPU / faster networking etc etc.

All this and pretty straight forward for local users to take advantage of.

But Will it Catch on ?

Yes is the simple answer.

For example:

We already have various groups who have been experiencing significant issues with disk contention and flexibility

We currently have a couple of significant cases where users are pushing things forward...

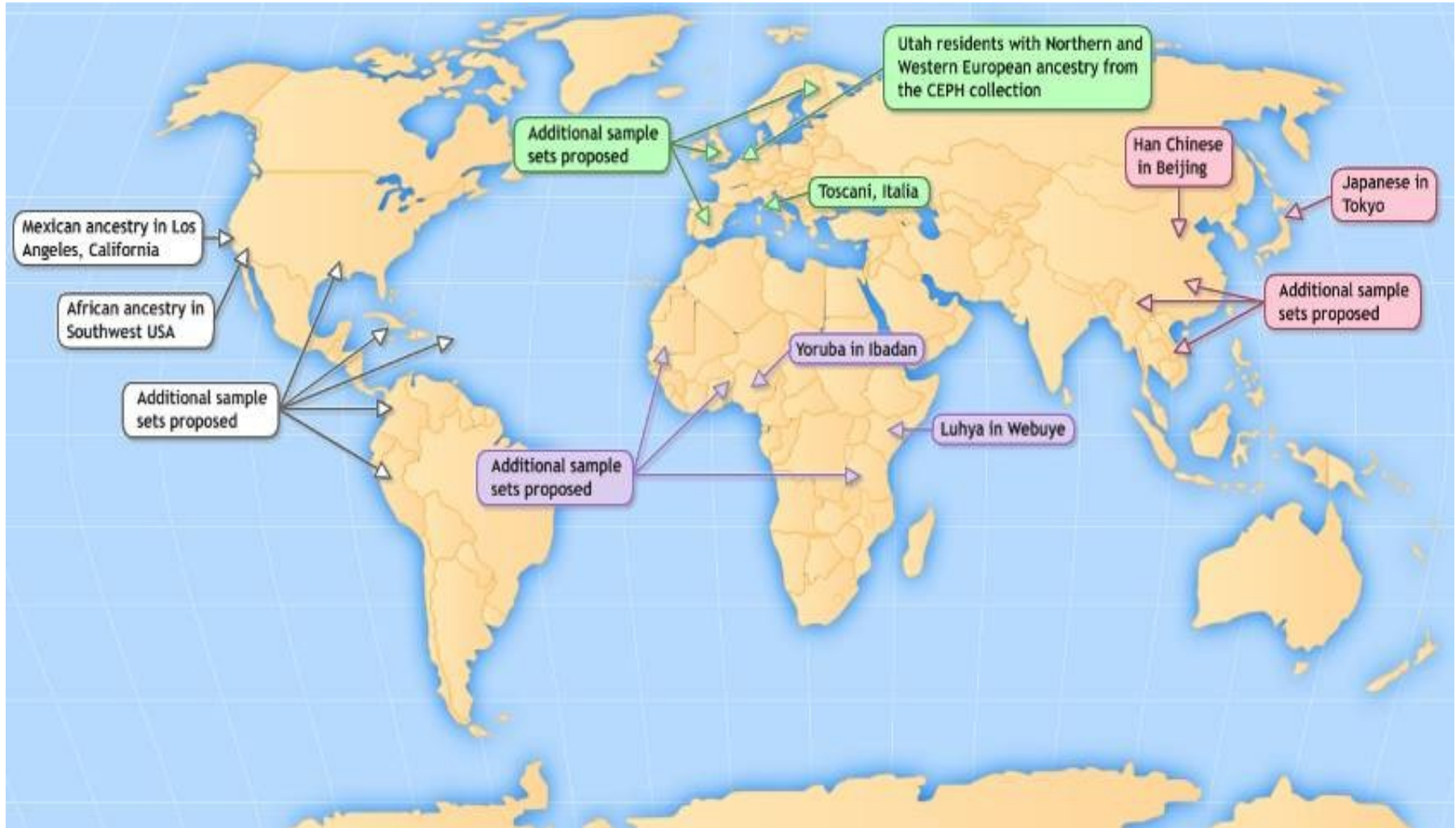
For example:

The 1,000 genome project intends to sequence at low coverage 1,000 individuals.

<http://www.1000genomes.org/page.php>

The data coming into this project, like many, is global.

1000 Genomes project



1000 Genomes project

On receiving the data an initial step is a QC, recalibration and alignment of the data received.

Previously the compute has been performed against monolithic lustre chunks.

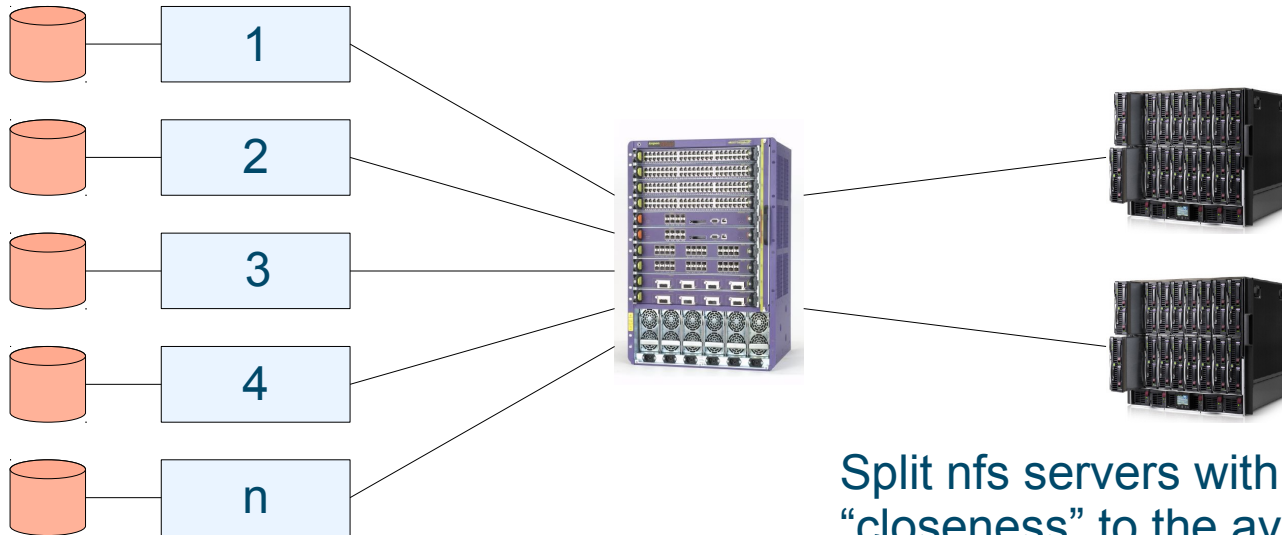
We've found that small linux based NFS units work well for this task, and they're cheap.

If they migrate to using the DAS setup available then LSF can provide load balancing, allowing wide scale out of the jobs.

Given the recent start of the 10,000 UK human genome project within the same group the need for such features is highly significant.

IO is already bursting at the seams

So the same model applies



Linux nfs
servers

Split nfs servers with varying “closeness” to the available compute (as per scratch example) and basic load balancing and data Provisioning for subsequent steps is all in place.

Where next ?

Well, 46.5 PB is rather a lot for a single site, especially if we consider this only as usable space.

Co-locations and cloud tech are potential ways ahead for both disaster recovery and future scale out.

We've tested LSF in spanning between UK and a small physical California Ensembl mirror. All works well as does the same within Amazon, so we can grow and apply the same model.

Data produced in co-locations needn't come back to site. Best if it doesn't. We share a 2 GB network link currently with the EBI and getting large scale data into / out of the cloud is challenging.

Once registered why move the data at all. If LSF is aware of the location within the Sanger LSF environment this is all taken care of. The best data transfer is the one you don't have to make.

Where next

Various collaborators already have access to certain Sanger compute facilities.

It's only time before we look at scaling external access to compute for others.

We have been investigating Grid and Glite for such provisioning of access and we have joined the CERN grid infrastructure.

Could be useful for us as local users can continue as before BUT external users could potentially use the external Grid methodology.

Once jobs arrive via the Glite stack they are passed onto LSF locally and DAS is again potentially available.

This is where the future gets interesting...

Further integration of DAS

Integration with iRODS, a particularly useful storage resource manager.

Users tend to think of it as a subversion alike for project files, but with hugely useful metadata magic.

This is already becoming a significant aspect of Sanger central and potentially the way ahead for data distribution between co-operating internal groups and external organisations and collaborators.

All new sequencing data is being pushed into this repository area.

Using groups and iRODS for sourcing data

iRODS allows grouping of storage units as a group and assign what goes where. It's clever and deals well with wide area networks and federation between trusted sites.

www.irods.org

If LSF knows which storage element the data is on it can again pick the compute closest to this and again reduce the time to collect the data from the repository.

- This is a significant pipeline workload on both storage and networking locally.

iRODS is also a contender for a secure fast ftp alternative for sharing data between external collaborators.

- If we link this in with, for example external access to local compute or resources the extent or data transfers can again be potentially reduced.
- Sharing is good(tm)

What comes next ?

“The crystal ball is covered in tea leaves.”

Nick Maclaren

But we do know that..

Tomorrows disks are getting bigger → fewer spindles
Reduced data access rates

More data → more compute
Faster data rates required

More cross-site data-sets
Slower external networks, out of our control

Data security.

This isn't going to go away. BIG kettle of fish. Moving jobs to originating data repository may be best approach for some.

With thanks to

Platform LSF

And in particular:

**Chris Smith
Chris Duddington
Da Xu**

Carine Kriegenhofer

Sanger WTSI

**James Beal, Guy Coates,
Gen Tao, Simon Kelley and
Phil Butcher**

And to our users.