Computer Science Issues in Environmental Infrastructure

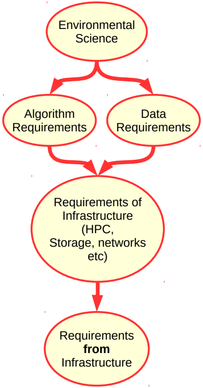
Bryan Lawrence

NERC SCIENCE OF THE ENVIRONMENT

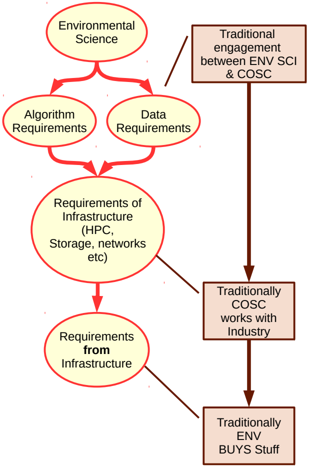
Science & Technology Facilities Council

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

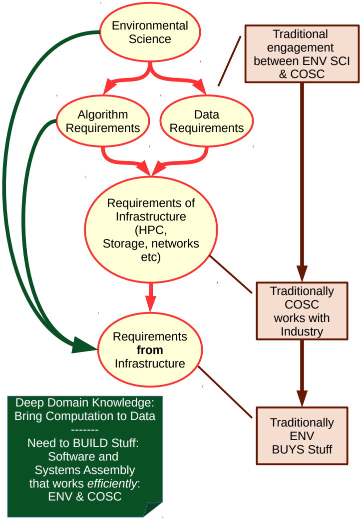
Co-Design


Environmental Science
Algorithm Requirements
Data Requirements
Requirements of Infrastructure (HPC, Storage, networks etc)
Requirements **from** Infrastructure

Co-Design

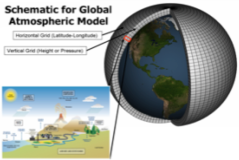

Environmental Science
Algorithm Requirements
Data Requirements
Requirements of Infrastructure (HPC, Storage, networks etc)
Requirements **from** Infrastructure
Traditional engagement between ENV SCI & COSC
Traditionally COSC works with Industry
Traditionally ENV BUYS Stuff

Co-Design


Environmental Science
Traditional engagement between ENV SCI & COSC
Algorithm Requirements
Data Requirements
Requirements of Infrastructure (HPC, Storage, networks etc)
Traditionally COSC works with Industry
Requirements **from** Infrastructure
Traditionally ENV BUYS Stuff
Deep Domain Knowledge: Bring Computation to Data

Need to BUILD Stuff: Software and Systems Assembly that works *efficiently*: ENV & COSC

Core Science Requirements



Today:	Observations	Models
Volume	20 million = **2 x 10^7**	5 million grid points 100 levels 10 prognostic variables = **5 x 10^9**
Type	98% from 60 different satellite instruments	physical parameters of atmosphere, waves, ocean

Soon:	Observations	Models
Volume	200 million = **2 x 10^8**	500 million grid points 200 levels 100 prognostic variables = **1 x 10^{13}**
Type	98% from 80 different satellite instruments	physical and chemical parameters of atmosphere, waves, ocean, ice, vegetation

→ **Factor 10 per day**

→ **Factor 2000 per <u>time step</u>**

→ **but many more time steps needed**

(Data courtesy of Peter Bauer, ECMWF)

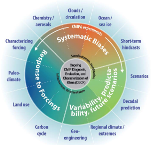
Big International Drivers:

WWRP World Weather Research Programme

GAW

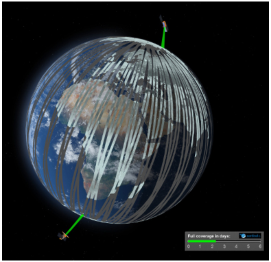
Copernicus Europe's eyes on Earth

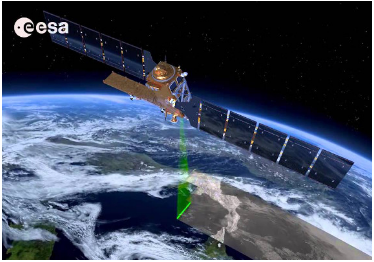
WCRP World Climate Research Programme



- aerosol cci
- cloud cci
- fire cci
- ghg cci
- glaciers cci
- antarctic ice sheet cci
- ice sheets greenland cci
- land cover cci
- ocean colour cci
- ozone cci
- sea ice cci
- sea level cci
- sst cci
- soil moisture cci
- cmug cci

The Sentinels: Big EO data crucial to NERC science!



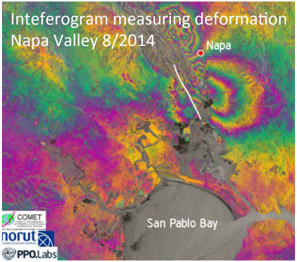


Sentinels

Sentinel 1A (2014), 1B (2016)
Sentinel 2A (2015) *2B (2017?)*
Sentinel 3A (2016) *3B (2018?)*
Data rate: o(6) PB/year

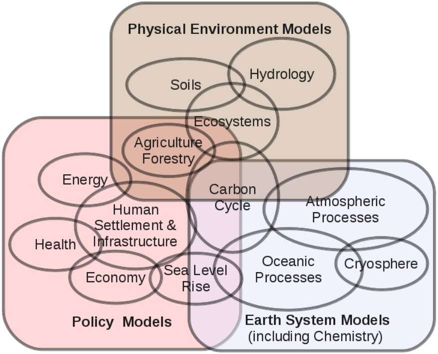
NERC SCIENCE OF THE ENVIRONMENT

COMET: Centre for Observation and Modelling of Earthquakes, Volcanoes, and Tectonics



(Picture credits: ESA, Arianespace.com, PPO.labs-Norut–COMET-SEOM Insarap study, ewf.nerc.ac.uk/2014/09/02/new-satellite-maps-out-napa-valley-earthquake/)

Communities



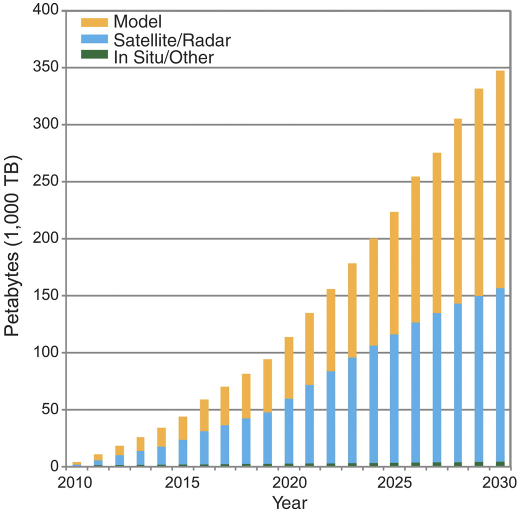
Many interacting communities, each with their own software, compute environments, observations etc.

Figure adapted from Moss et al, 2010

More Data

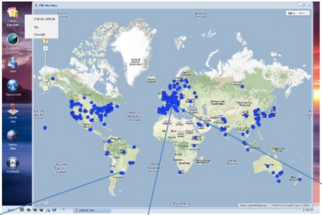
Fig. 2 The volume of worldwide climate data is expanding rapidly, creating challenges for both physical archiving and sharing, as well as for ease of access and finding what's needed, particularly if you're not a climate scientist.

(BNL: Even if you are?)



J T Overpeck et al. Science 2011;331:700-702

The trend



is-enes

Slide courtesy of Stefan Kindermann, DKRZ and IS-ENES2

Individual End Users

- Limited resources (bandwidth, storage,..)

Organized User Groups

- Organize a local cache of required files
- Most of group don't access ESGF, use cache instead!

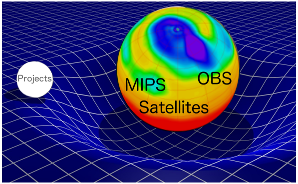
Data Centre Service Group

- Provides access to ESGF replica cache
- May also provide access to data near compute resources
- (BADC, DKRZ, IPSL, KNMI, UC)

Trend

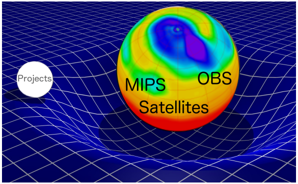
Needed: Replacement for „*Download and Process at Home*" Approach

JASMIN — The Data Commons

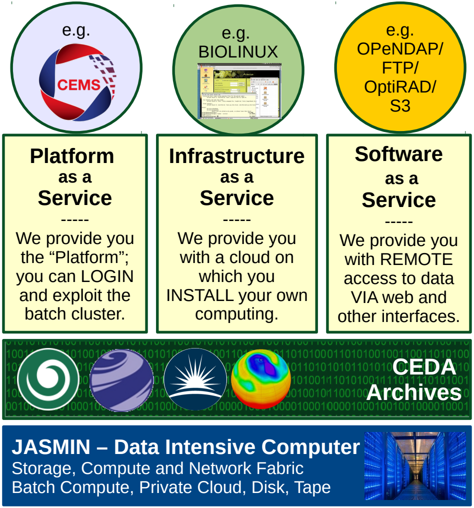


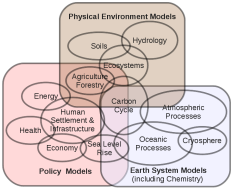
- Provide a state-of-the art storage and computational environment
- Provide and populate a managed data environment with key datasets (the "archive").
- Encourage and facilitate the bringing of data and/or computation alongside/to the archive!
- Provide FLEXIBLE methods of exploiting the computational environment.

JASMIN — The Data Commons

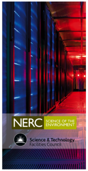


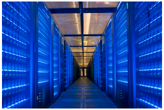
- Provide a state-of-the art storage and computational environment
- Provide and populate a managed data environment with key datasets (the "archive").
- Encourage and facilitate the bringing of data and/or computation alongside/to the archive!
- Provide FLEXIBLE methods of exploiting the computational environment.





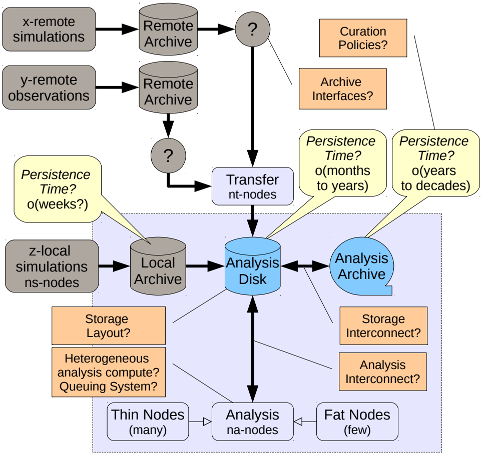
Many interacting communities, each with their own software, compute environments etc.





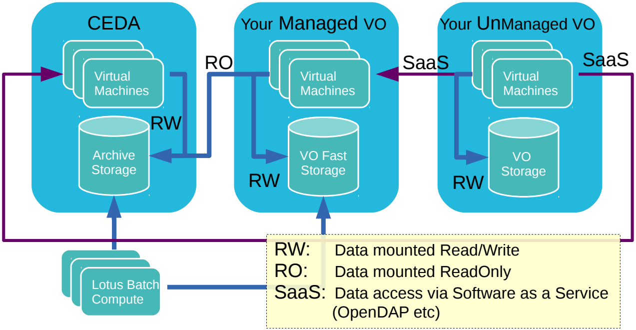
16 PB Fast Storage
1 PB Bulk Storage,
Elastic Tape
4000 cores
(hypervisors & batch cluster)

JASMIN SuperData Environment



The issue: Handling petabytes of storage with terabytes in each of hundreds of workflows, each of which has different software requirements: from single threaded, to MPI, to containers … and soon to be exabytes with petabytes in each of hundreds of workflows.

Objective is to provide an environment with high performance access to curated data archive **and** a high performance data analysis environment!



Curated environment one virtual organisation within o(100) such virtual organisations. Key issues include:

(1) how to provide high performance data access and analysis in the managed environment for multiple users, multiple workflows, intersecting in some of the data,

(2) between unmanaged (infrastructure as a service) and the data held in (our) managed environment, and

(3) data growth that exceeds the Kryder rate (volume/bandwidth etc).

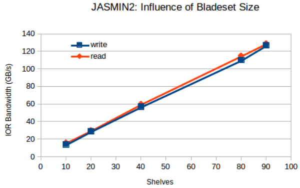
The seven deadly sins of ~~cloud~~ computing research
Schwarzkopf, Murray, Hand
Hotcloud, 2012

Pick five, all in play:

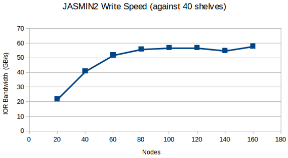
- *Unnecessary distributed parallelism:* We need to support (nicely) high memory and other nodes inside our environment.
- *Assuming performance homogeneity.* This is a real problem for us in a mixed VM/batch environment ... Help.
- *Forcing the abstraction (Map-Reduce, HADOOP or bust)* We avoid this by having a parallel file system, but how do we know we are getting value?.
- *Unrepresentative workloads.* We really don't know how to optimise our jobs (yes, we can give people exclusive access to nodes, but it's harder to give them exclusive I/O bandwidth).
- *Assuming perfect elasticity.* We haven't worked out how to schedule to use our resources, or how to cloud burst properly.

We need work on understanding all these things

Pick one issue: I/O ~~optimisation~~/control



Do we understand the performance at the user/app level?
We can break our file system up into pools ("blade sets") in Panasas.
Give communities access to resources on one blade set.
Now their I/O does not interfere with VOs using other blade sets.



Issues:
— This isn't very flexible! We can still nail a PB bladeset with 80 nodes! How do we get more and flexible I/O parallelisation?
— When we run out of physical space for disk, how are we going to efficiently use tape in our workflow?

Common Software/Algorithm Patterns

Supporting a wide variety of algorithms and workflows: (but much to do to exploit parallelism)

“Big Data Ogres” by analogy with the Berkely Dwarves for computational patterns.

Different Problem Architectures, e.g:

1. Pleasingly Parallel (e.g. retrievals over images)
2. Filtered pleasingly parallel (e.g. cyclone tracking)
3. Fusion (e.g. data assimilation)
4. (Space-)Time Series Analysis (FFT/MEM etc)
5. Machine Learning (clustering, EOFs etc)

Important Data Sources, e.g:

1. Table driven (eg. RDBMS + SQL)
2. Document driven (e.g XMLDB + XQUERY)
3. Image driven (e.g. GeoTIFF + your code)
4. (Binary) File driven (e.g. NetCDF + your code)

Sub-Ogres: Kernels & Applications, e.g:

1. Simple Stencils (Averaging, Finite Differencing etc)
2. 4D-Variational Assimilation/ Kalman Filters
3. Data Mining Algorithms (classification/clustering) etc
4. Neural Networks

Modified from Jha et al 2014 arXiv:1403.1528[cs]

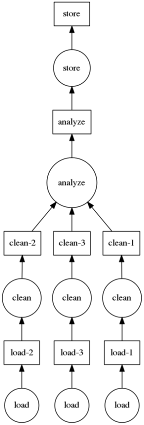
Uncommon software solutions: How to make these play nicely with each other?

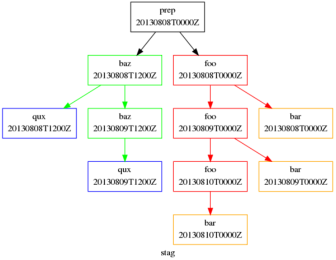
Plethora of parallel architectures and tools out there

Contrast between two very different parts of our workflow:

- Many of our analysis tasks are build once, use once, throwaway. No room for optimisation (or MPI), yet
- Much of our workflow is repeatable: “build”, “run”, “move”, “reduce/reformat”, “analyse”. Much room for automation.

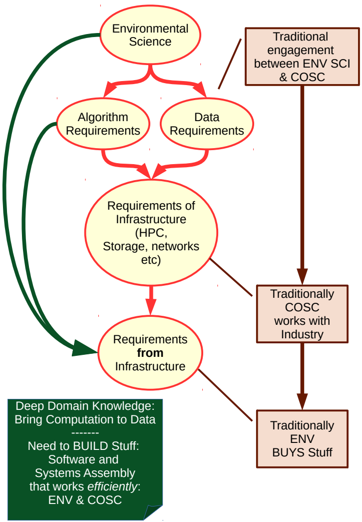
Whatever tools we use, we'll need to get use to generating, understanding, and exploiting concurrency in more complicated ways:





Much to do, as infrastructure providers, and users, to harness these tools to accelerate our workflows!

(These two examples: dsk, and cylc, representing analysis and scheduling, reduction and proliferation.)



Lots of interesting problems on the left:

- System-acious: Cloud computing, Cloud-bursting, Storage Paradigms (HDF in object stores), Raid to Tape in real workflows?
- Workload: How to schedule and manage high-performance environmental ogres on bare metal and in clusters with and without containers?
- Metadata: How to efficiently search, maintain and find data amongst millions of CF compliant files?
- Algorithms: Refactoring our *analysis* algorithms for high volume on next generation computing.