University of Glasgow

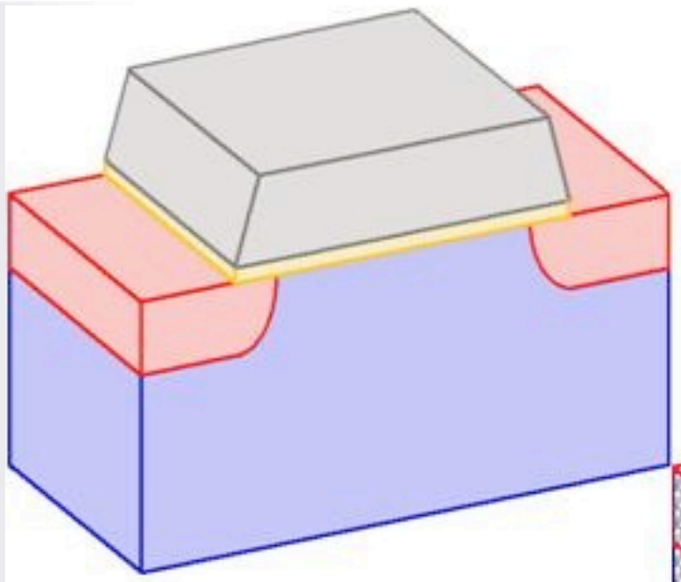
Use of AFS in the nanoCMOS project

Christopher Bayliss c.bayliss@nesc.gla.ac.uk

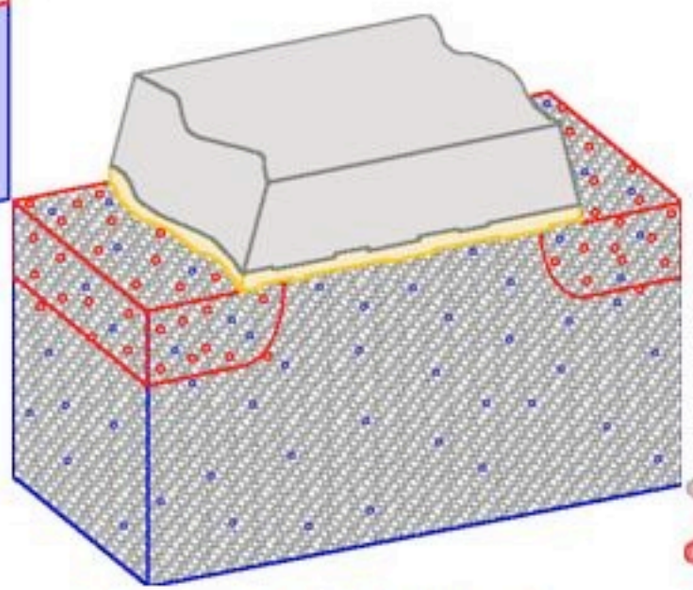
University of Glasgow

The nanoCMOS project

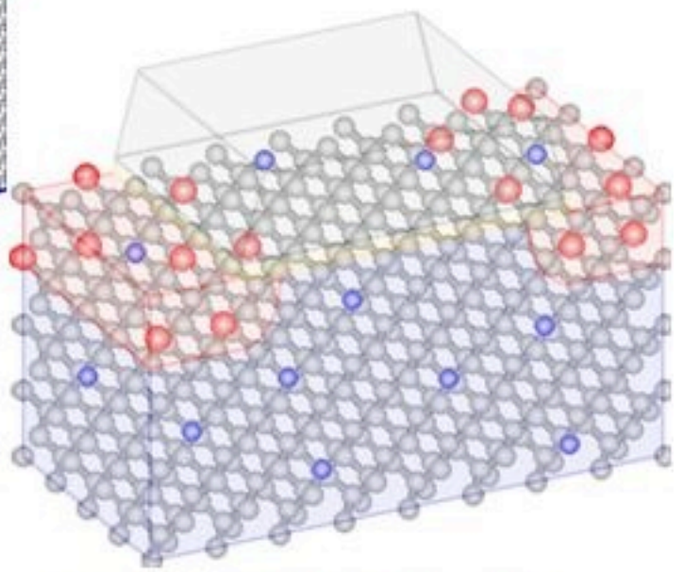
- Designing > 40n-10n scale CMOS circuits.
  - EPSRC funded project.
  - 6 Academic and 6 Industrial partners.
- Simulates CPUs from transistor up.
  - Based on transistor and circuit designs from industrial partners.
- www.nanocmos.ac.uk



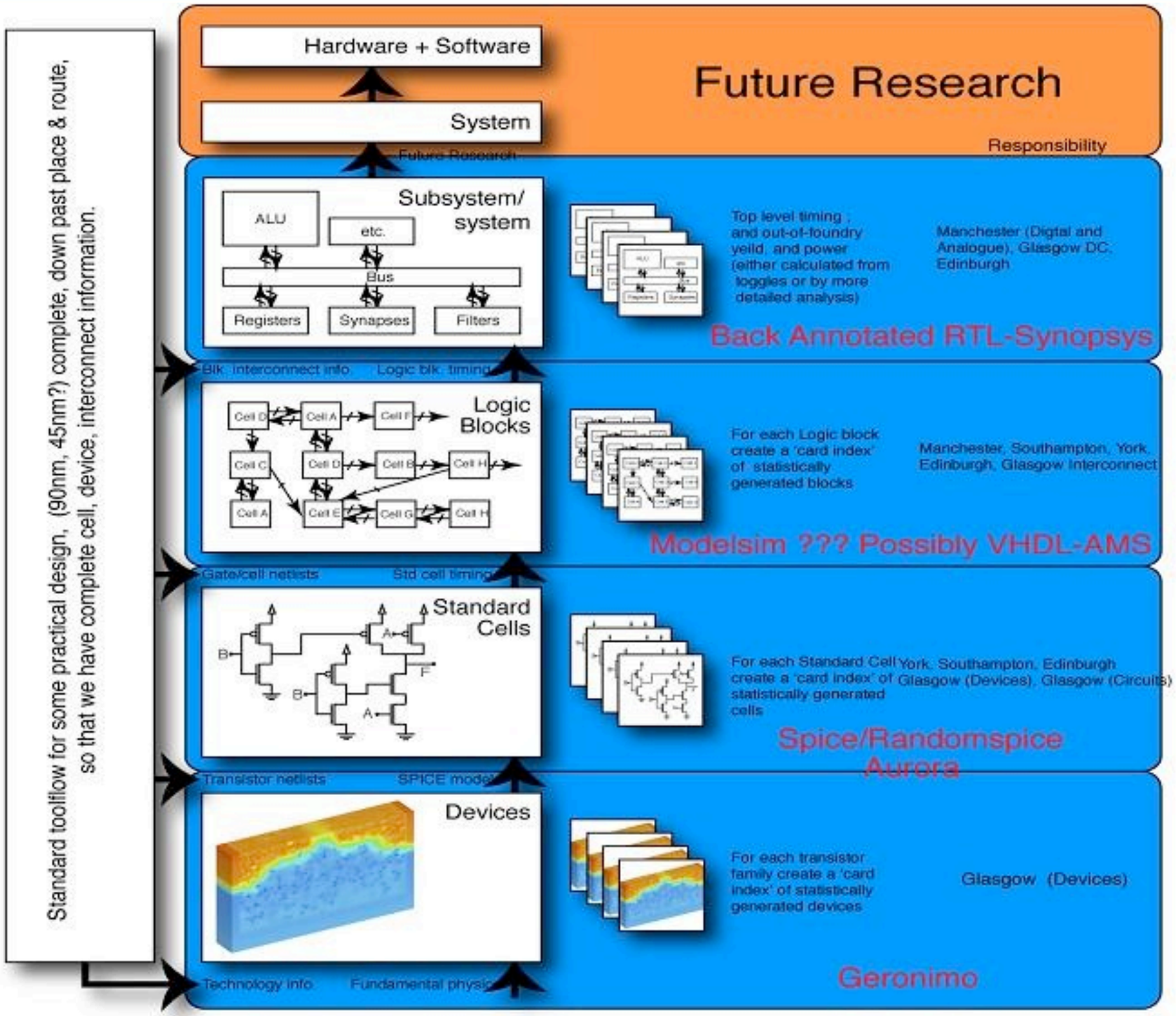
The simulation paradigm now

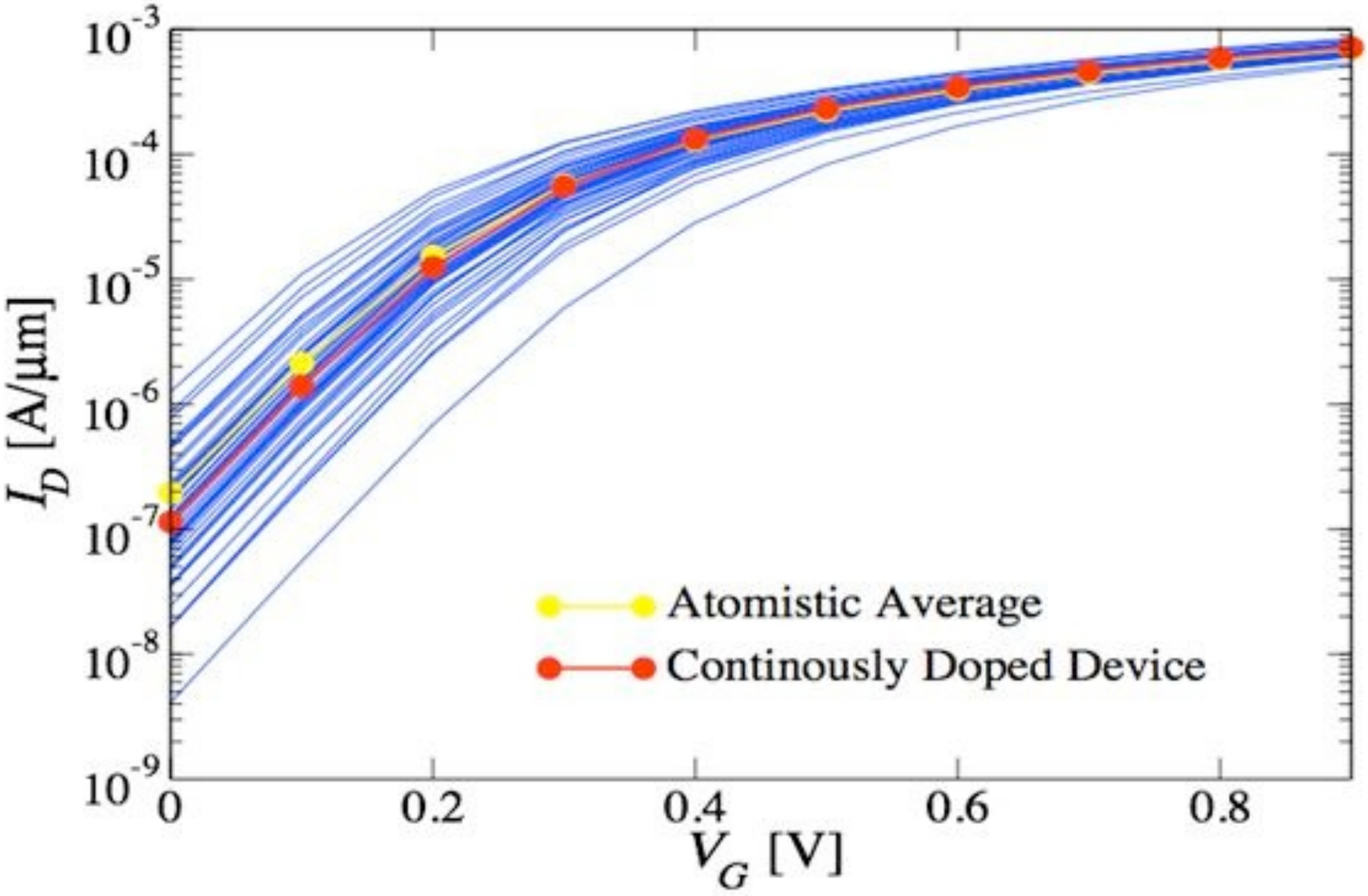


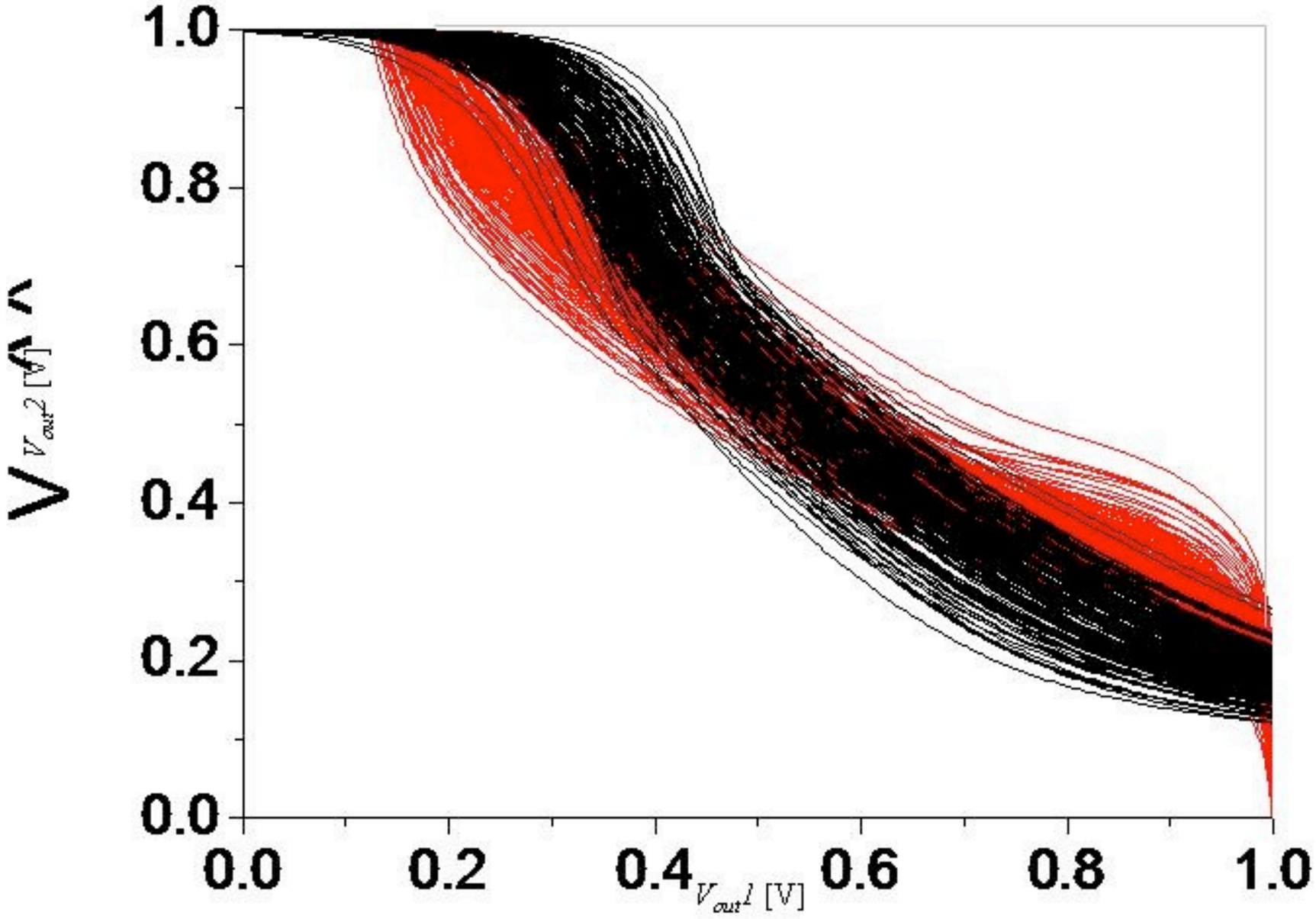
22 nm MOSFETs in production 2009

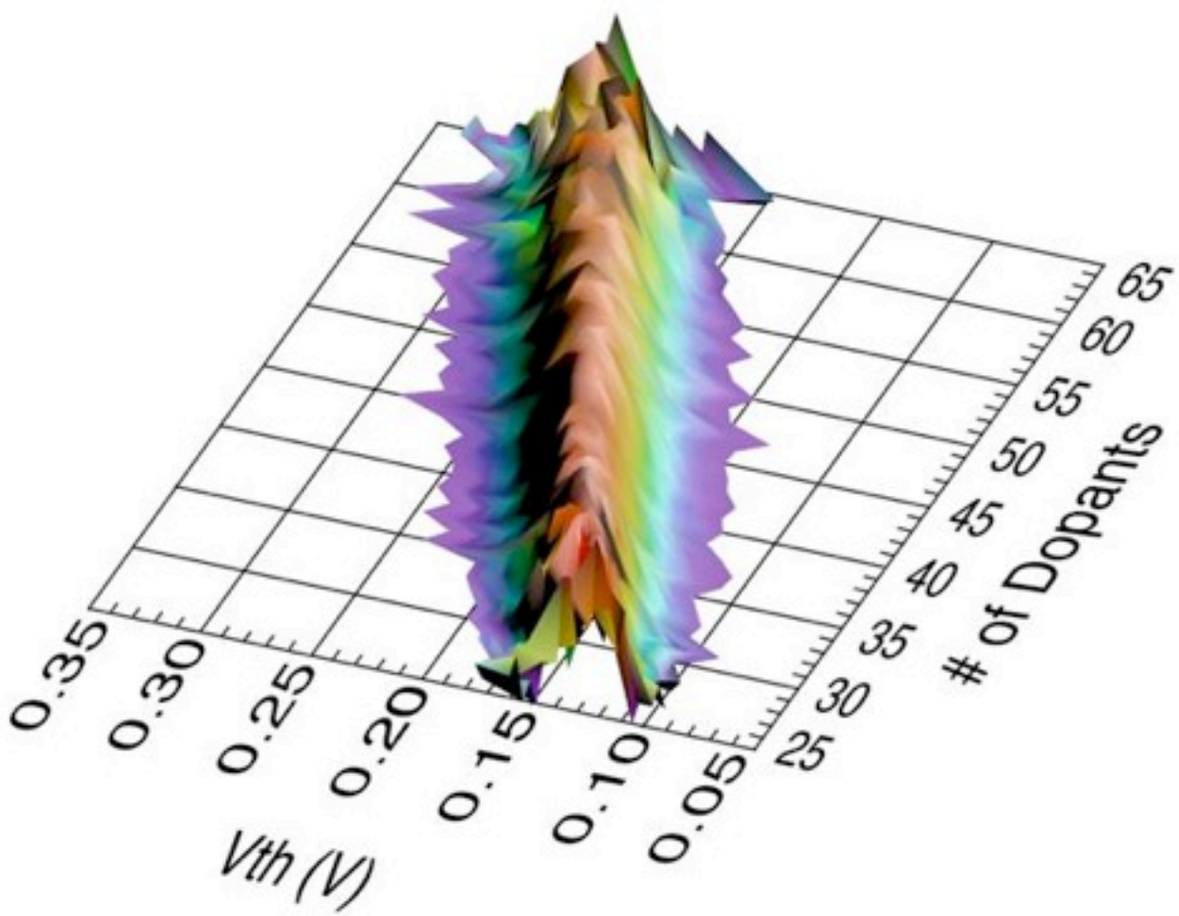


4.2 nm MOSFETs in production 2023

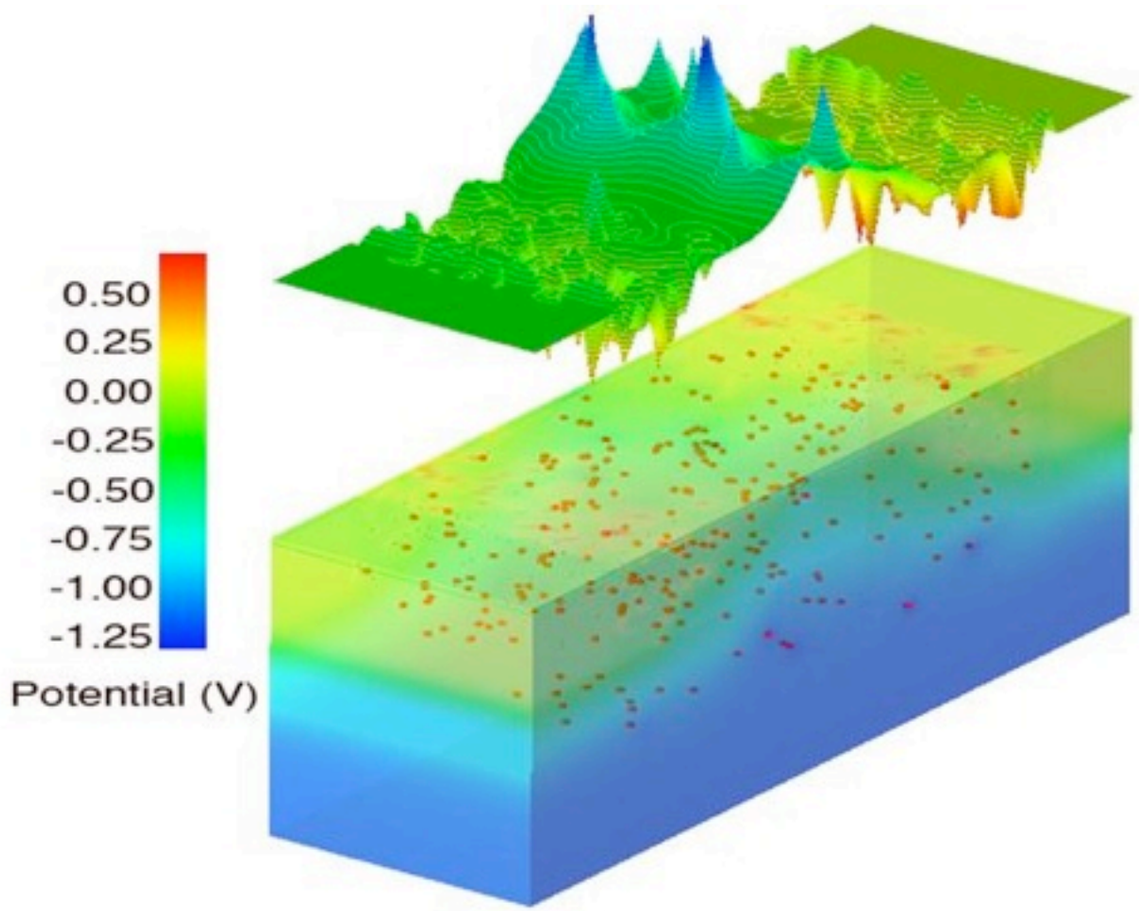

Hardware + Software
System
Future Research
Responsibility
Future Research
Subsystem/ system
ALU
etc.
Bus
Registers
Synapses
Filters
Top level timing ; and out-of-foundry yeild, and power (either calculated from loggies or by more detailed analysis)
Manchester (Digtal and Analogue), Glasgow DC, Edinburgh
Back Annotated RTL-Synopsys
Blk interconnect info
Logic blk timing
Logic Blocks
Cell D
Cell A
Cell F
Cell C
Cell D
Cell B
Cell H
Cell A
Cell E
Cell G
Cell H
For each Logic block create a 'card index' of statistically generated blocks
Manchester, Southampton, York, Edinburgh, Glasgow Interconnect
Modelsim ??? Possibly VHDL-AMS
Gate/cell netlists
Std cell timing
Standard Cells
A
B
F
B
A
For each Standard Cell create a 'card index' of statistically generated cells
York, Southampton, Edinburgh Glasgow (Devices), Glasgow (Circuits)
Spice/Randomspice Aurora
Transistor netlists
SPICE model
Devices
For each transistor family create a 'card index' of statistically generated devices
Glasgow (Devices)
Geronimo
Technology info
Fundamental physic
Standard toolflow for some practical design, (90nm, 45nm?) complete, down past place & route, so that we have complete cell, device, interconnect information.


10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
I_D [A/μm]
0
0.2
0.4
0.6
0.8
V_G [V]
Atomistic Average
Continously Doped Device


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
V_{out2} [V]
V_{out1} [V]

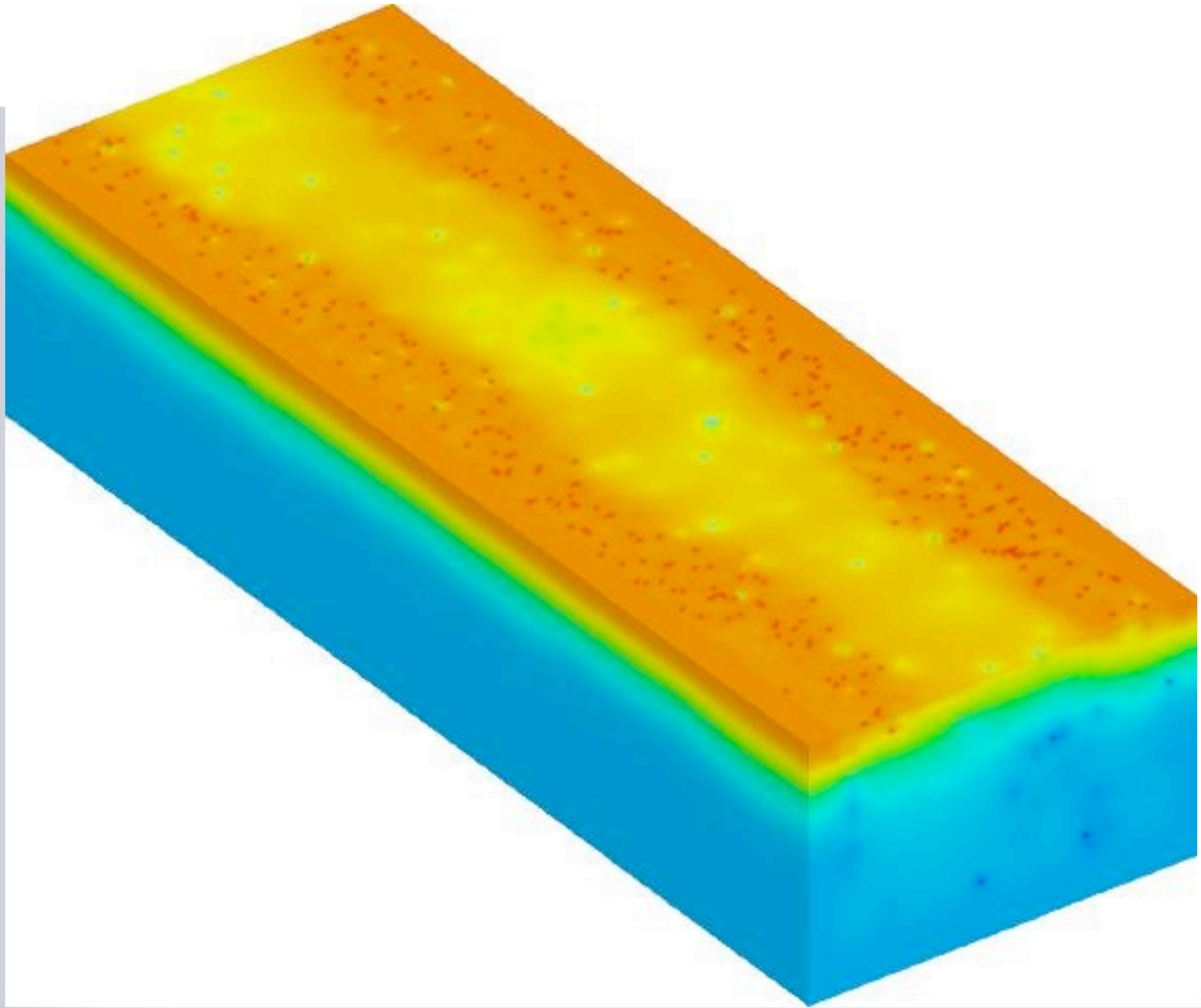


Potential and dopant position of a statistically-rare device.



Threshold voltage variation as a function of the number of dopants.

University of Glasgow



Current Status of nanoCMOS

- Still in development
- Running small device simulations on internal and partner clusters.
  - NGS deployment in testing.
- Non-developer users soon.

nanoCMOS

- A grid project
  - x509 certificates
    - GSI proxies
  - SOAP / HTTP
  - Globus / OMII
    - Required for NGS access.
  - Ideological attachment to grid solutions.
- Uses the Virtual Organisation model.
  - No overarching organisation.
  - Set of subsets
  - Lots of administrative boundaries.

Virtual Organisations

- Common pattern
  - User creates proxy certificate
  - Proxy used to access service
  - Service accesses user's resources.
- Traditionally x509 certificates and GSI proxy certificates.
  - Users have problems with them
  - Complications with CAs
- Kerberos friendlier.
  - Built in or easily available support.
  - Many apps support it.
  - Drop to PAM when not natively supported.

Why choose AFS?

- Need for some form of distributed storage.
  - Secure
  - WAN
- Initial options AFS, SRB and custom code.
  - Those that had used it said SRB was horrible.
  - We liked the idea of a conventional file system.
  - We had some AFS expertise on the project.
    - Two sites run AFS cells.
  - BaBar tried this before.
- According to my email first suggested Oct 2007 as part of some other work.

Heterogeneous with a capital H

- 6 academic partners
  - No policies governing HW or SW.
- Using 3rd party compute resources.
  - No common job submission.
  - No common software environment.
  - No common architecture.
- Uses purchased and “acquired” equipment.

What does nanoCMOS use AFS for?

- Hosting:
  - Input data
  - Simulation software
  - Authentication tools
  - User space
- Storage for:
  - Simulation log files.
  - Simulation results.
  - Individual user accounts.

Current AFS infrastructure

- One AFS cell.
  - NESC.GLA.AC.UK
  - Hosted at NeSC Glasgow
    - 2x Sun 12TB X4500
    - 407GB currently in use
    - Solaris
    - ZFS
    - SMF management scripts
    - Currently OpenAFS 1.4.10
- Kerberos cell.
  - Also at NeSC.
    - Single master / slave pair.
- GSSklog

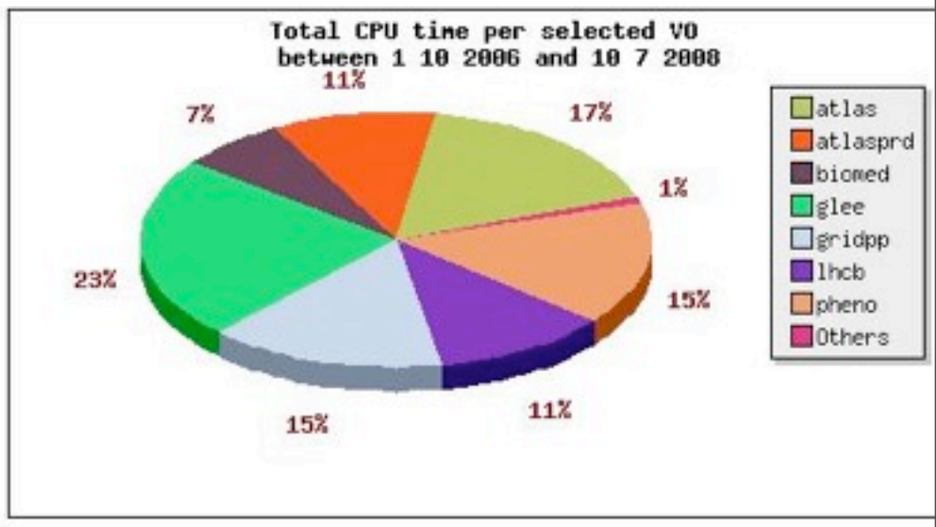
The Cell

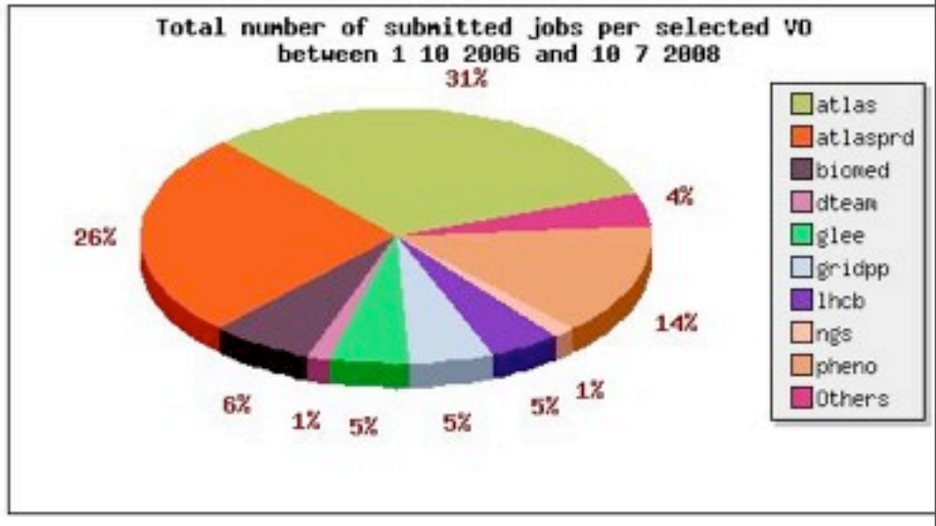
Current AFS infrastructure cont.

- Users have ssh access to an AFS client
  - Play around without installing.
  - Can push / pull files straight away.

Where?

- Client deployments
  - Windows, OS X and Linux clients
  - Clients on developer's systems at all 6 sites.
  - Clients installed on 5 clusters.
    - EE at Glasgow - 145 nodes [1160 cores]
    - ScotGrid at Glasgow - 309 nodes [1916 cores]
    - NeSC test cluster at Glasgow - 14 nodes [28 cores].
    - Manchester - 48 nodes [256 cores]
    - Edinburgh - 246 nodes [1456 cores]


Total CPU time per selected VO
between 1 10 2006 and 10 7 2008
11%
7%
17%
1%
23%
15%
15%
11%
atlas
atlasprd
biomed
glee
gridpp
lhcb
pheno
Others


Total number of submitted jobs per selected VO
between 1 10 2006 and 10 7 2008
31%
4%
26%
14%
6%
1%
5%
5%
5%
1%
atlas
atlasprd
biomed
dteam
glee
gridpp
lhcb
ngs
pheno
Others

Issues

- The batch systems we use have x509 based authentication.
  - No Kerberos on the worker.
  - GSSklog stored in the cell readable by anyuser
- Admins not expecting to let a UDP protocol through their firewall.
  - Shouting / Patience.
  - tcpdump

Getting AFS installed

- Most clusters complex and fragile
  - Multiple submission mechanisms
  - Multiple users
    - Differing and conflicting needs.
  - Heavily loaded
- Admins do not like the words kernel, firewall or reboot.
  - A constant stream of 30 day jobs on some workers.

Issues

- Bursty load
  - x000 jobs reading and writing from the same file, directory or volume.
- Many clients.
  - 5000 cores available on partner clusters. ~8000 when primary NGS sites included.
- Capacity
  - Users with dozens of 1GB files to store.
- No metadata
  - Built own metadata service
- Directory permissions

Pitfalls

- The network is outside our control.
  - Debugging connectivity issues is painful and slow.
- Most clusters outside our control.
  - Must ask local admins nicely to add or fix AFS.
  - Must work through local nanoCMOS people.
- Most clients outside our control.
  - Random versions of AFS on a random OS.
  - Homogenisation not an option.

Good aspects of AFS

- Allows us to bypass staging data and executables to nodes.
  - Can be a real problem on the NGS for several reasons.
    - pre-WS GRAM job submission.
    - Unreliable environment at sites.
- PAGsh
  - Useful when running jobs on a shared cluster node.
- Easier to set up than equivalent NFSv4
- Kerberos useful for services other than AFS.

Good aspects of AFS

- @SYS
  - We run a lot of code from AFS.
-

Bad aspects of AFS

- Requires kernel module.
  - Not easy thing to get system admins to install.
- Debugging
  - Either no information or too much
- NAT
  - Fiddly to set up.
  - Almost all clusters use NAT.
- Not simple to modify client config.
  - You can’t simply make changes to the config and restart afsd.
  - Can’t modify client if you need some setting
    - Currently test then continue / quit.

Good features of Kerberos

- Users understand passwords
- Can use Kerberos for other services.
  - Currently ssh with forwarding.
- Create users with expired passwords.
  - If they can log in and change their password they probably set it up OK.

Wish List

- GSSAPI authentication.
- Better NAT behaviour.
- Better out of the box settings.
  - Especially for clients.
- Single cache on a cluster.
  - Save bandwidth on parameter sweep jobs.
- Consistent command line parameters

Future AFS usage at Glasgow

- Starting to be used for data for other projects from other departments.
  - We need somewhere to store data that isn't a drawer of USB keys.
- Quick way to backup data from servers.
-

Future AFS usage in nanoCMOS

- Improve server design.
- Probable move to storing cell and realm details in DNS
  - Centralise configuration somewhere we control.

- Global file space simplifies user's problems
  - Put data onto AFS from desktop
  - Copy paths into job
  - Retrieve data from path specified on desktop.