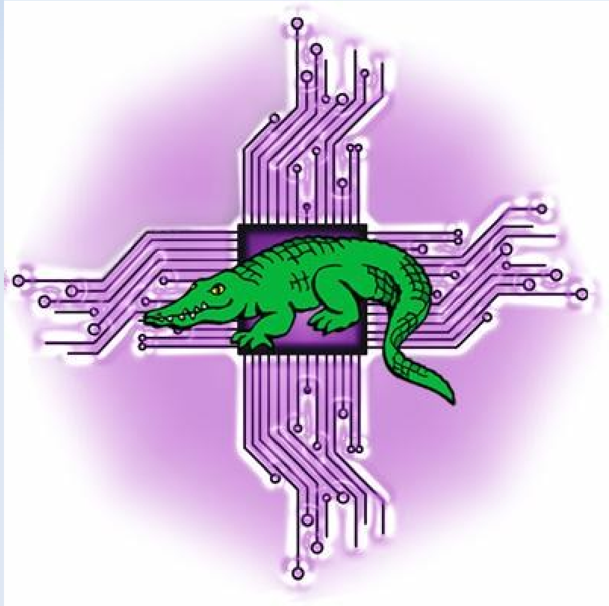
Interactive HPC Computation with Open OnDemand and FastX



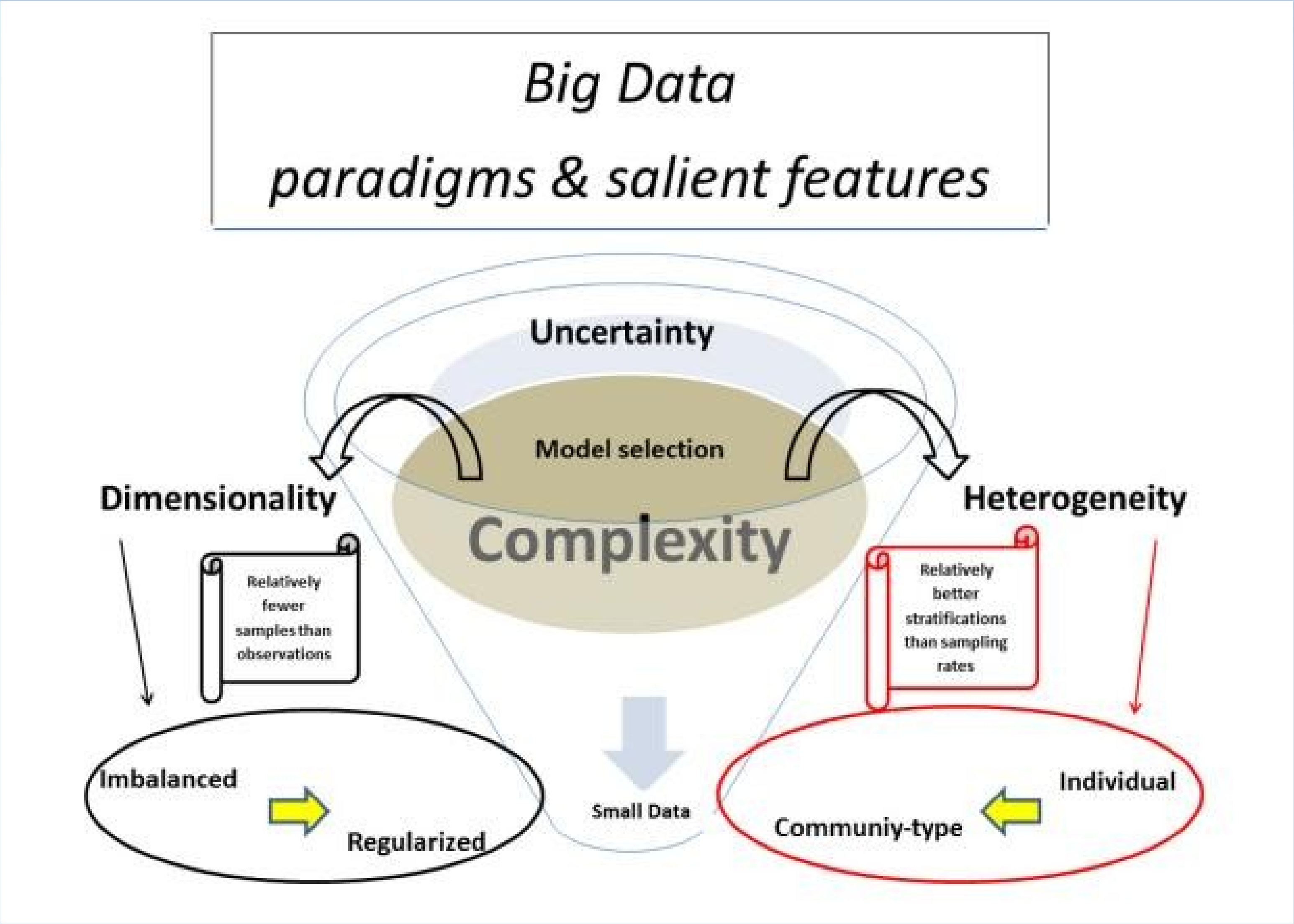
eResearchNZ, February, 2021

lev.lafayette@unimelb.edu.au

More Data, More Complexity

- Datasets and problem complexity [1] is growing faster than performance of personal computational systems.
- In late 2018 a predictive study [2] of what was described as "the Global Datasphere", data that is "created, captured, or replicated ... will grow from 33 Zettabytes (ZB) in 2018 to 175 ZB by 2025".
- Performance improvements gained by increased clock-speed on a processor come at a cost of additional heat, causing a breakdown in Dennard scaling in the early 2000s [3], that has continued to contemporary times.
- There have been several technological improvements over the decades that have mitigated what would otherwise be a very unfortunate situation for the processing of large datasets (multicore, GPUs, parallel programming, SSD, NVMe, high-speed interconnect etc)
- A preferred choice is high performance computing (HPC) which, due to its physical architecture, operating system, and optimised application installations, is best suited for such processing.
- Command-line Tools can be 235x Faster than your Hadoop Cluster (https://adamdrake.com/command-line-tools-can-be-235x-faster-than-your-hadoop-cluster.html). SQL query runtime from 380 hours to 12 with two Unix commands (https://www.spinellis.gr/blog/20180805/)

More Data, More Complexity



Compute on HPC, visualise locally

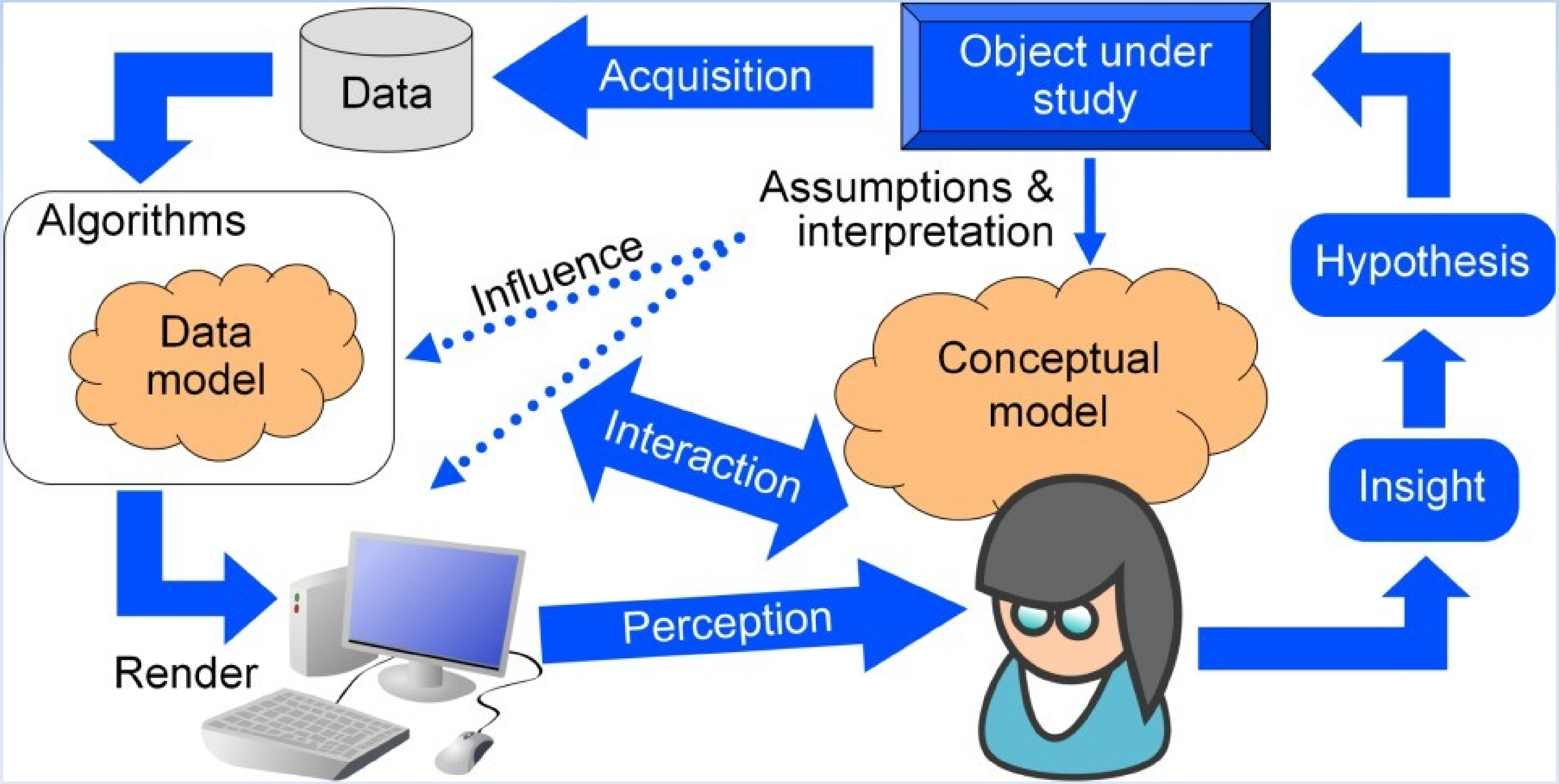
• The motto of his Richard Hamming's *Numerical Methods for Scientists and Engineers* (1962) was "The purpose of computing is insight, not numbers". How is this insight to be achieved? From raw data can you mentally create a real-world mental image of what the data represents?

• Complexity will be beyond most people's cognitive capacity, leading to the need for computer graphics as a requisite heuristic. It is requisite because for large datasets it is almost impossible to locate and represent small but important interactions between data elements. It is a heuristic as it will provide insight that will otherwise be missed. Visualisation is a cognition amplifier [4].

• Computer graphics provides for expository and exploratory representations [5]. Rational exploration begins with a hypothesis, collects data that will confirm or falsify the model. Visualisation of the data helps provide an expository sense, and exploration through modification of the model, leading to new insights and hypotheses. However, like other visual media, computer graphics require significant bandwith and attentive processing by the the human perceptual system.

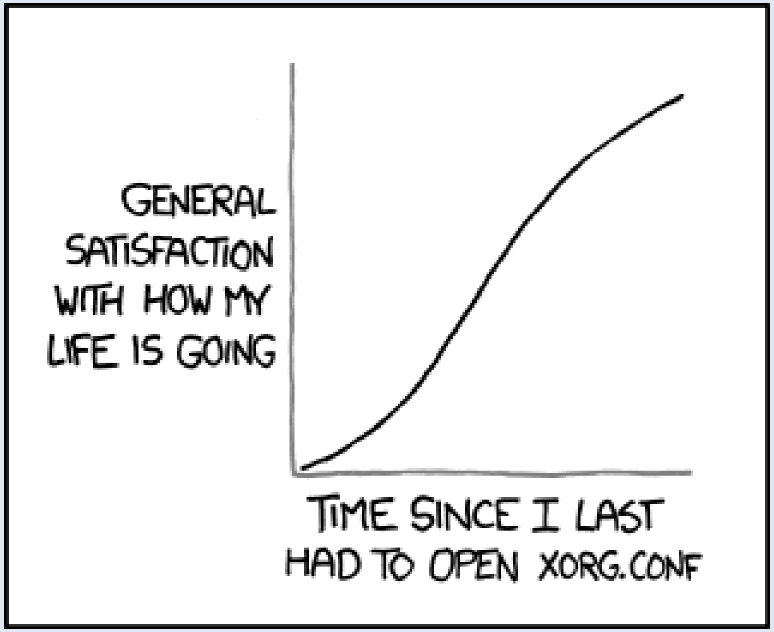
• But HPCs are traditionally quite poor at this! HPC systems have historically been less effective at visual display, and least of all in an interactive manner, leading into a general truism of "compute on the HPC, visualise locally".

Compute on HPC, visualise locally



Distance and Details

- There are three reasons for the poor performance of graphics forwarding on most HPC systems. One relates to the underlying reality of network performance measured as the congestion of a quantity data over the bandwidth and latency between two points. The second relates to researchers who seem to believe that the reality of network performance doesn't matter - such as requesting mount point on compute nodes to a remote dataset because transferring data is an extra-step they have to worry about.
- The third reason is the underlying architecture of the X11 protocol [6]. X11 wasn't designed to handle the sort of graphics operations more prevalent in contemporary HPC computation. Rather than X11 sending a screen from the remote HPC to a local display system, X11 sends the display-instructions so the X-server on the local system recreates the screen locally.
- All of this means multiple round trips between the local and remote display, which add to latency. In terms of bandwidth, it doesn't do any compression on data by default and the compression option only reaches a 2:1 ratio; by comparison formats like H.264 can obtain compression rates of 100 to 1. Also, for legitimate security reasons, the -X option is typical which adds to further bandwidth requirements..



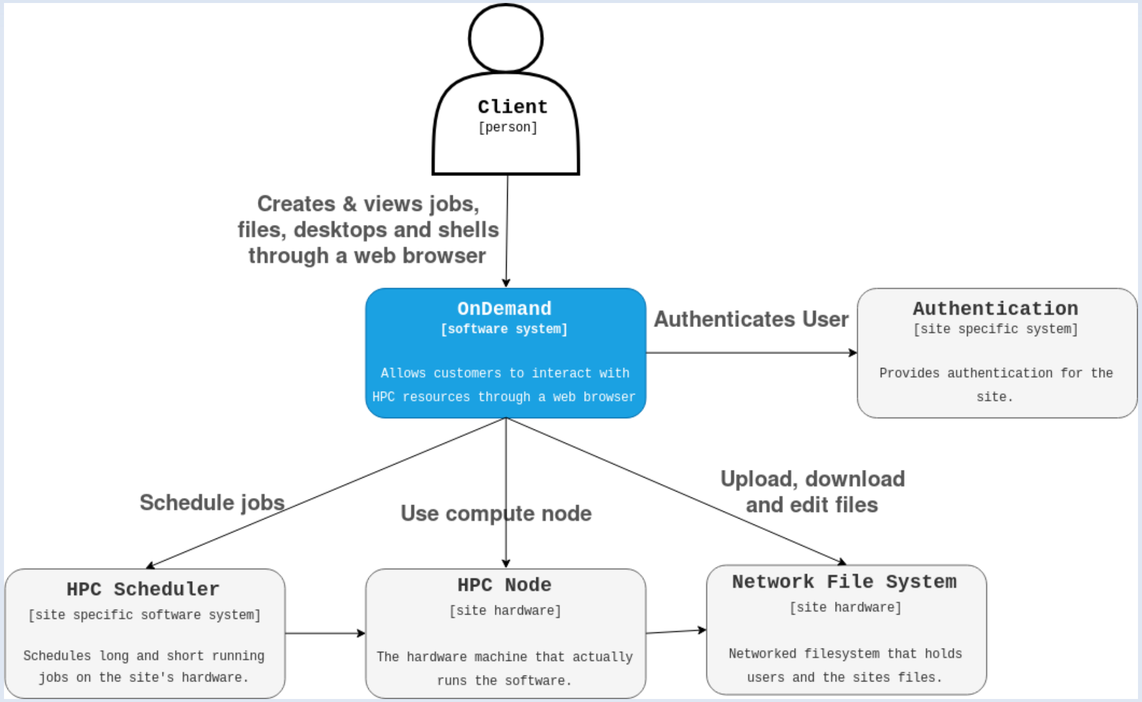
FastX for Remote Desktop

- There are several solutions to this problem. This includes simpler graphics, a simpler desktop, or make use of cache/forward alternatives such as X2go and NoMachine NX Technology (which was used on UniMelb's previous system, Edward). This can also involve VirtualGL that redirects the 3D-rendering commands from Linux OpenGL applications to 3D accelerator hardware.
- An alternative range of remote desktop software use Remote Frame Buffer (RFB) protocol used in Virtual Network Computing (VNC), such as TigerVNC, TightVNC, and Cendio's ThinLinc.
- We did use ScienTific Remote Desktop Launcher (Strudel) for a period, a TurboVNC-based system developed originally at VPAC, then Monash University, that has deployed at several Australian research institutions, including NCI, Pawsey, Monash University.
- This led UniMelb to deploy FastX from Starnet (founded in 1989) which has excellent performance. Whilst the technology is proprietary Starnet does advertise that their performance is due to well-known approaches such as remote server rendering, improved compression algorithms, monitoring of local graphic application changes, automatic or manual adjustment of frame-rate depending on network speed, and frame windowing for interactive applications.
- UniMelb uses FastX on the Spartan HPC and managed cloud. Educational organisations can purchase a campus-wide license for c$5000USD per annum which includes technical support, an approach used by more than 120 universities world-wide.

Open OnDemand, RStudio, Jupyter Notebooks

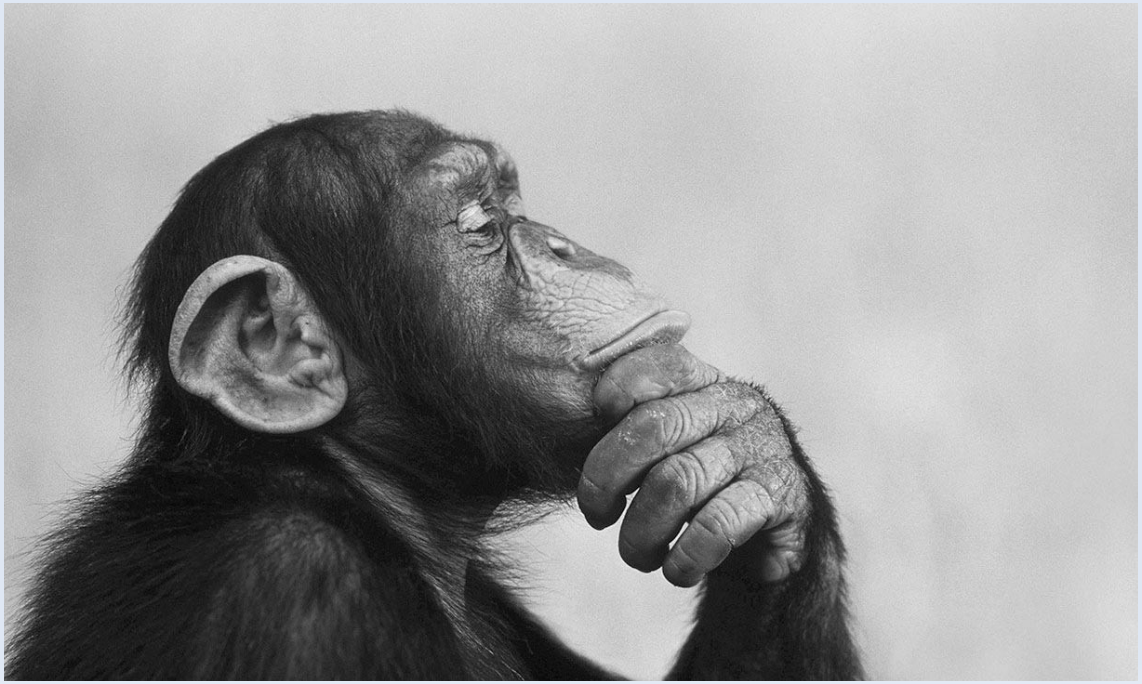
- FastX provides half the solution, that of remote graphics forwarding. The other half is the deployment of Open OnDemand, which allows users to access HPC resources through a web-browser on their local system [7]. An open-source project and NSF award recipient, OnDemand uses HTML5 standards for development providing users a familiar web-based front end to HPC systems. The use of HTML5 allows for a web proxy that provides a federated authentication, a "zero-install" and single sign-on (SSO) solution for their researchers.

- For UniMelb, Open OnDemand provides web access to HPC resources, such a file management, command-line access, job management, and access to certain web-enabled applications. These include RStudio, an integrated development environment for the R programming language for statistical computing and graphics, and Jupyter Notebooks, that enable the creation and sharing of documents with live code, visualisation, and narrative text across multiple programming languages, especially Julia, Python, and R.



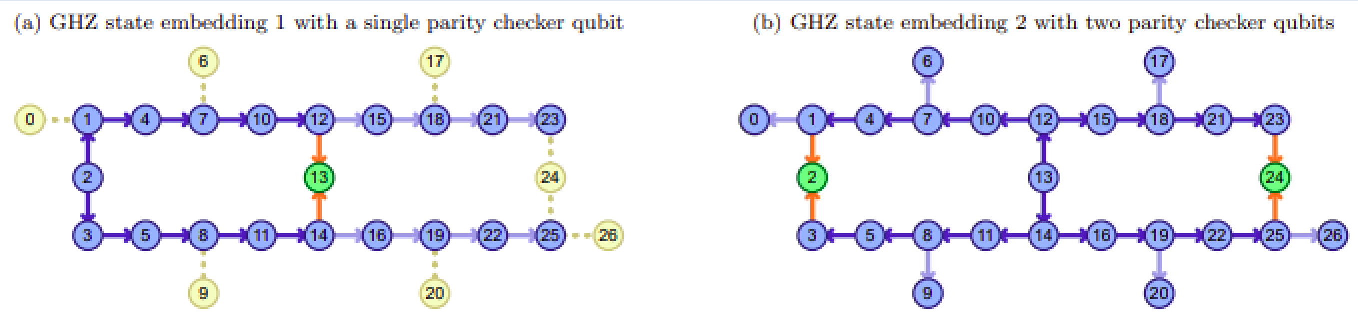
RStudio Case Studies

- RStudio has been used for cytometry data in R with openCyto along with analysis for single cell RNAseq experiments in the context of stromal cell subsets. The researchers had to use HPC system as the datasets were very large and RStudio provided for step-wise interaction and immediate graphic output. Another group used RStudio for high-dimensional sequencing datasets such as single-cell RNA-seq.
- It was also used on a project for analysis for a large sales dataset, where researchers could make use of additional RAM, the ability to load multiple files, and receive real-time feedback; this was in addition to using MPI in preprocessing.
- In addition, RStudio was used for deep Gaussian process discriminant analysis comparing five classification methods on various datasets, including a Covid-19 Mass Spectrometry dataset.
- Another project, dedicated to functional genomics, uses RStudio to analyse large human and chimpanzee genetic variation datasets to study human evolution and determine how changes in the DNA sequence lead to differences between species and populations.



Jupyter Case Studies

- Jupyter was used to process large datasets (approximately 15 million rows and 12 GB of data) for business insights over an eight-month research project.
- It was also useful as the IBM quantum computer echo-system is in Python, that made it an extremely good match for our research students, and is eing used for quantum tomography and the development of a new for non-Markovian quantum systems, along with a demonstration of of some of the largest genuinely multipartite entangled states on any quantum computer, along with initial investigations into moment-based techniques, and tensor network techniques to represent quantum states and processes.
- In addition, Jupyter has been used extensively to process experimental data from Belle in Japan and ATLAS at CERN, using Spartan's CernVM-File System (CernVM-FS) providing a global interface to bespoke experimental software is essential and the Python interface to the ROOT data analysis framework, especially its visualisation capabilities. The Jupyter-notebooks provide a fast edit-code/display-the-results cycle for interactive work as we tune the models.



Observations and Future Developments

- There are only 32 projects on the Spartan HPC system using RStudio and 48 using Jupyter Notebooks; a tiny percentage of the 1462 projects that Spartan has overall. A very clear gap is that web-enabled access to the HPC have only just been incorporated into our introductory training workshops. If numbers do increase there will be pressure on the current, small interactive partition in Slurm for such users.

- All the contributors from projects in this survey of the major users indicated strongly that they used HPC in the "normal" manner, that is for performance and scale in batched job submission to complete their processing requirements, even though tools like RStudio and Jupyter were very convenient in their workflow for testing and interactive feedback. Of note, users have found the applications on Spartan "is as intuitive and easy to use as ... on local machine".

- A number of contributors in the survey did encounter complexity when dealing with the need for library extensions for their chosen applications, although issues were quickly resolved by Research Computing staff. This again, is a matter for stronger incorporation in the introductory training workshops.

.

Acknowledgements and References

Thanks to the following researchers who contributed to the content: Dominik Schienstock, Akriti Sharma, Charles Hill, Martin Sevior, Dave Jain, Weichang Yu, Irene Gallego Romero, and Hyun Jae Lee.

References

[1] Capobianco, Enrico., "Born to be Big: data, graphs, and their entangled complexity", Big Data and Information Analytics, Volume 1, Issue 2: 163-169, 2016

[2] Reisnel, F., Gantz, J., Rydning, J., "The Digitization of the World: From Edge to Core", International Data Corporation, Nov 2018

[3] Bohr, Mark. "A 30 Year Retrospective on Dennard's MOSFET Scaling Paper". Solid-State Circuits Society, January 2007

[4] Valle, M., Visualization: A cognition amplifier, International Journal of Quantum Chemistry, 31 May 2013

[5] Kirby, R., Keefe, D.E.., Laidlaw, D.H., Painting and Visualization in Hansen, C.D., Johnson, C.R. (eds)., The Visualization Handbook, Elsevier Butterworth–Heinemann, 2005

[6 Hania Gajewska, Mark S. Manasse and Joel McCormack, "Why X Is Not Our Ideal Window System", Software – Practice & Experience vol 20, issue S2, October 1990]

[7] Chalker, A., Franz, E., Rodgers, M. et al., Open OnDemand: State of the platform, project, and the future, in Concurrency and Computation: Practise and Experience, Wiley, 2020

Images

Slide 3 from Capobianco, Enrico., 2016, ibid. Slide 5 from Valle, M, 2013, ibid. Slide 6 from Munroe, R., https://xkcd.com/963/. Slide 8 from the Open OnDemand documentation https://osc.github.io/ood-documentation . Slide 9 from medium.com . Slide 10 from "Generation and verification of 27-qubit Greenberger-Horne-Zeilinger states in a superconducting quantum computer"


THANKS FOR WATCHING
& LISTENING PATIENTLY
makeameme.org