Computer hardware for hydraulic modelling

Greg Collecutt
Jaap van der Velde

Computer hardware for hydraulic modelling

Greg Collecutt
TUFLOW (BMT)

Jaap van der Velde
TUFLOW (BMT)

"Choosing optimal compute hardware, on-premises or in the cloud, improves workflow efficiency and the ability to deliver modelling results on budget and on time. With a particular focus on GPU compute, this webinar is for IT managers as well as modellers and project managers. The webinar addresses key issues and challenges when selecting, configuring, and using compute environments, along with an overview of current trends for hydraulic and environmental modelling."

TUFLOW

aws Australian Water School

Overview

Questions to answer:

- What platform are we talking about?
- What specs/metrics matter on a GPU?
- What about the rest of the hardware?
- Can one size fit all?

Questions we won't answer:

- How to optimise your model.
- How to optimise your process / modelling pipeline.

Trends + Your Questions & Answers.

Platform

- GPU-accelerated model running over CPU
  - GPU is where the real performance gain is, and where the hard choices are.
  - Models of reasonable size spend most their time on the GPU.
  - This is particular to hydraulic modelling; not generally true of modelling.
- NVIDIA over AMD or Intel
  - Our experience is with NVIDIA, but we feel conclusions translate to other brands.
  - Optimising for one architecture pays off when compared to a generic solution.
- Windows over Linux (in part)
  - Despite licencing, performance, and cloud benefits, most 1D/2D engines are now on Windows.

Why GPUs matter most

The power of GPUs:

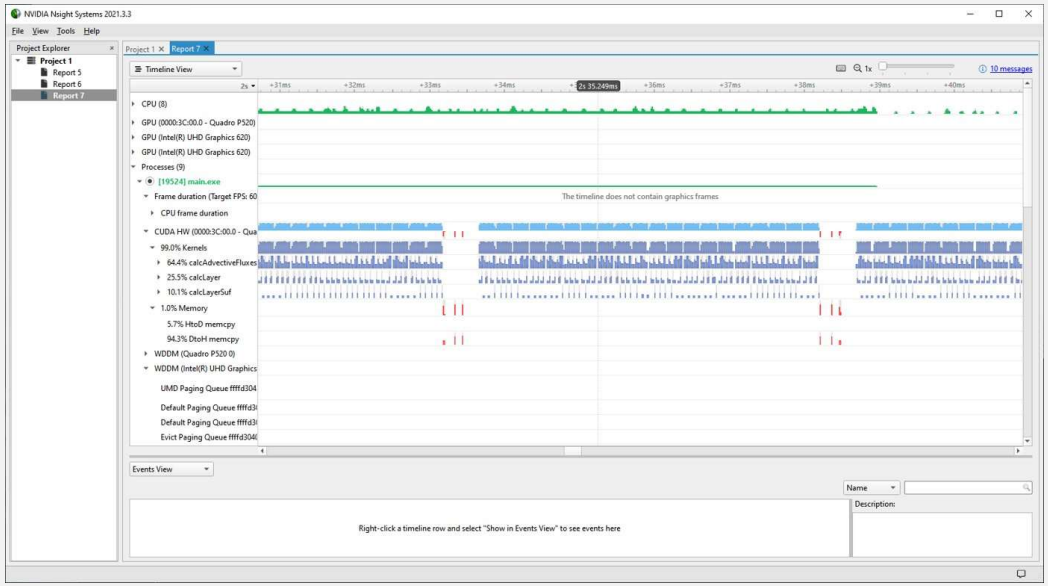
- 1,000s - 10,000s vs. 10 - 100+ of cores.
- Optimised for exactly the types of operations that matter.

GPU choice has decisive impact:

- Your software may only support specific technology.
- GPU choice dictates power requirements, system architecture, even the hardware platform (desktop vs. server).

What is the GPU up to?

A model engine may spend most of its time in computation, transferring data to and from the GPU infrequently. In this model engine example, the blue represents kernels computing, while the red shows data transfer. As a result, the process won't be hampered by lower bandwidth to the GPU.



Relevant GPU specs/metrics

- Cores
  - Architecture trumps cores, due to feature size, clock speed, connectivity, etc.
- Memory
  - Size should be sufficient for your larger models.
  - Caching and bandwidth can trump size.
- Shared infrastructure, more than one user to a GPU
  - Beware shared GPUs on VDI.
  - Consider split GPUs using MIG on server hardware.
- More than one GPU to a user
  - If your model is too large for a single GPU, it may run on two more affordable models.

Sharing on VDI vs. MIG

- In a Virtual Desktop Infrastructure (VDI), many desktops can share a single GPU. ***This type of solution works very well when just running Windows applications, but not if a modelling engine wants to take all the CUDA cores it needs for itself***.

  For example: you may see cloud-based desktop infrastructure that shares 2 large GPUs among up to 25 desktops. A very economical solution for office work, even with 3D components, but not suitable for model running.

- However, large server GPUs can be shared by multiple modellers if configured for it. ***NVIDIA's Multi-instance GPU (MIG) technology can present the GPU as multiple smaller, independent GPUs*** to an operating system, or a hypervisor (so that they can be passed on to individual VMs).

  For example: a single NVIDIA A100 can be split up into up to 7 GPUs, each with about 1/7th of the total cores and memory available to it and in our experience, each of those GPUs can perform as well as a single 40xx-series GPU for the purposes of running a model.

The rest of the hardware

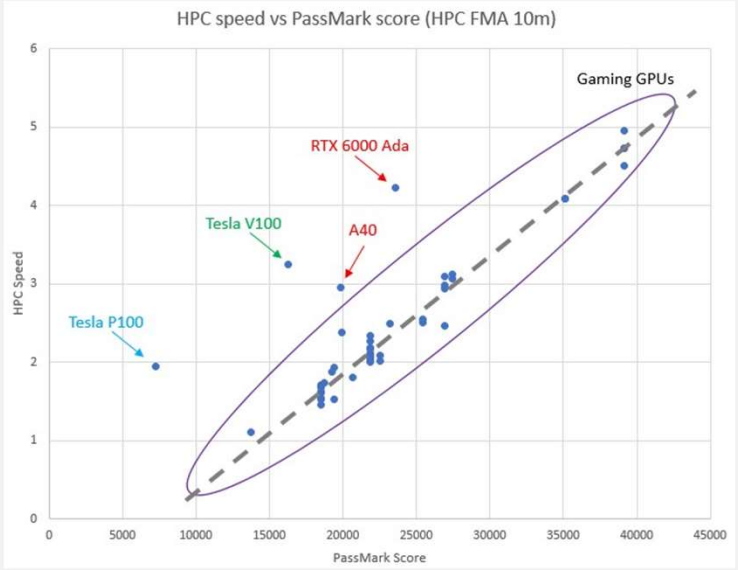
- Processor: 1 CPU core alongside each GPU
  - During model startup and post-processing, you may benefit from a *few* more.
- Memory: size 4-6x GPU memory
  - Faster is better. Note that memory beyond 128GB is considered ‘server-grade’.
- Storage: write locally, SSD vs. HDD is negligible
  - Input mostly affects startup, output is dominant.
- Mainboard: durable and sufficient size to fit all GPUs
  - A chipset that allows for sufficient PCIe lanes and fast memory
- Power: wattage and connectors to power it all, at capacity

Not hardware, but important…

- Consider your model configuration first
  - What formats are you writing?
  - How frequent do you need outputs?
  - Do you need everything for runs in the current phase of your project?
- Clever automation beats expensive hardware
  - Moving data around is more efficient than more storage.
  - Writing automation scripting is a good use of an engineer's time.
  - Script coding is in everyone's reach with ChatGPT, Claude, CoPilot, Bard, etc.
  - Consider PowerShell, Bash, batch, as well as Python, JavaScript, Go, etc.

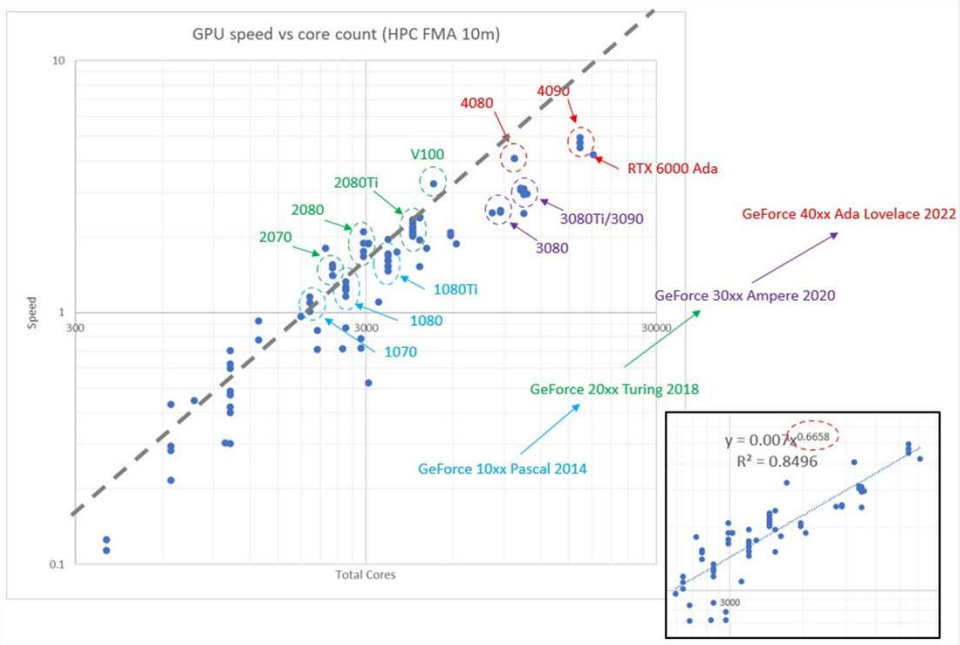
Benchmarks are king

- What matters is how fast *your models* are on the hardware
- For hardware without comparable benchmarks, consider a proxy



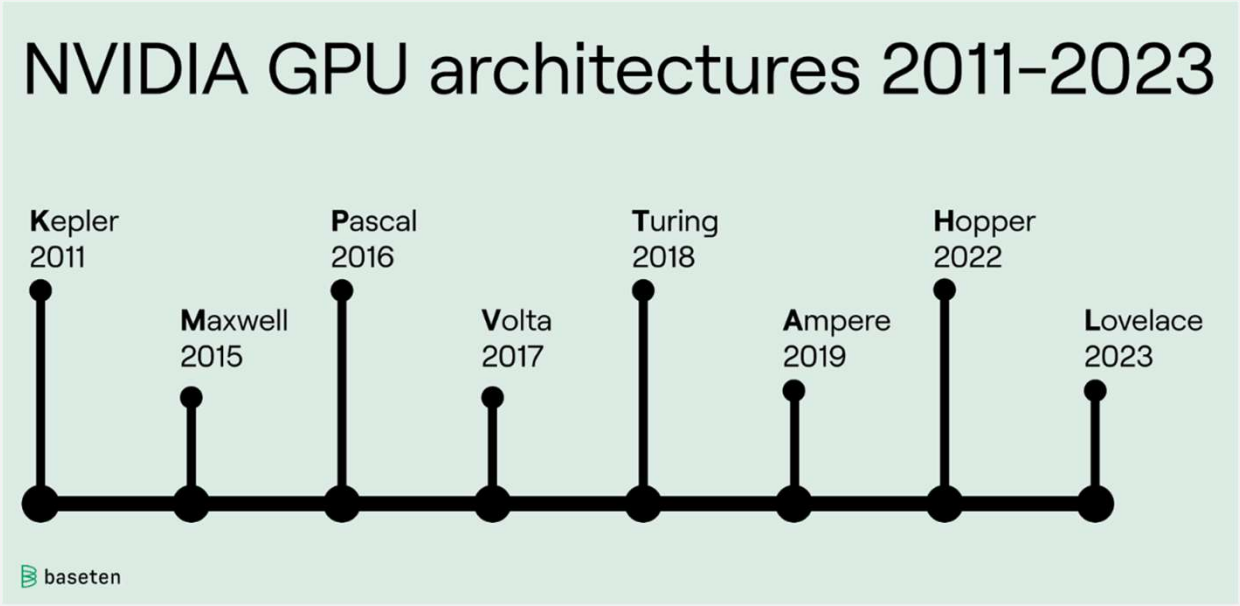
Benchmarks are king

- Consider running a standard benchmark on your hardware



An aside

- Actually…
  (for NVIDIA, the letters match architectures, the numbers are just model numbers)



One size fits all?

- Model design and calibration vs. production runs
  - Design and calibration benefit most from minimising runtime.
  - Production benefits most from minimising compute cost.
- Time on licence vs. time on hardware
  - Faster makes the best use of your licences.
  - Slower is better for hardware and power cost.
- Owning vs. renting compute
  - Owning can be desktops, servers, or semi-permanent cloud servers
  - Renting can be PAYG on the cloud, SaaS/PaaS, or third party runs
  - Additional renting complexities: provisioning, licencing, data management

One size fits all?

- Finding the right mix
  - Have sufficient fast compute for the design and calibration phase.
  - Fast hardware allows for more targeted design, avoiding parameter scans.
  - Develop a model on a CPU workstation and run it with a GPU machine.
  - If the size of your business doesn't allow for diversity, only get the fastest.
  - Balance the purchase of more hardware with the cost of licences.
  - Consider that over time the latest and greatest becomes the low end and plan for it.
- When the cloud is better
  - Preparing for on-demand computing allows scaling for large projects or emergencies.
  - Can you bring your entire process into the data centre or cloud?

Trends

- Most, if not all, engines will move over to GPU; *CPU won't catch up*.
- AI has driven innovation in high performance GPUs well beyond what games require, in part in ways that hydraulic modelling benefits from.
- Hot and loud desktops are being replaced with central, shared compute in the server room accessed from cool laptops or from home.
- Up to and including the 40xx and L40 (Lovelace) generation, GPU technology is showing no sign of slowing when it comes to modelling.
- Individual cards have gone well beyond what a single model needs, but if your engine allows sharing, the improved speed is still worth it.
- Putting 4 (or more) cards in a single machine is getting harder, unless you look at specialised hardware.

Questions?

TUFLOW

aws Australian Water School

Questions? (encore)

A few hardware-related questions we commonly hear in support queries:

- "Will my computer <<list of parameters>> run TUFLOW?"
  Almost always 'yes, on CPU at a minimum'; the real questions: 'at what speed?', 'does it support my GPU?', and 'do I have enough RAM?'
- "My model failed, what is wrong with TUFLOW or my licence?"
  In most cases there is nothing wrong with the software, but rather there is a problem with hardware, connectivity, OS, or licencing.
- "Is this computer <<parts list>> better than this computer <<parts list>> for TUFLOW?"
  Most of these questions can be resolved by looking at the provided benchmarks.
  (See https://wiki.tuflow.com/index.php?title=Hardware_Benchmarking_-_Results#GPU_Results or start at wiki.tuflow.com)

In general, it pays to involve your IT staff in improving your modelling workflow and performance.