# The Future of HPC

## Exascale and Challenges

Partners

Funding

Horizon 2020
European Union Funding
for Research & Innovation

# Reusing this material

# Outline

- Future architectures
  - Exascale initiatives
  - Processors
  - Memory
  - Impacts on performance
- Software challenges
  - Parallelism and scaling
  - New algorithms
  - What about software that does not scale?
- Impact for standard computing

# Future architectures

## What will HPC machines look like?

# What will future systems look like?

| | 2016 | 2020 |
|---|---|---|
| System Perf. | 100 Pflops | 1 EFlops |
| Memory | 1.3 PB | 10 PB |
| Node Perf. | 100 Gflops | 1-10 TFlops |
| Concurrency | O(1000) | O(10000) |
| Interconnect BW | 40 GB/s | 200-400 GB/s |
| Nodes | 10,000 | O(10000) |
| I/O | 2 TB/s | 20 TB/s |
| MTTI | Days | O(1 Day) |
| Power | 15 MW | 20 MW |

# Processors

- More Floating-Point compute power per processor
  - Only exploit this power via parallelism
  - Lots of low power compute elements (cores) combined in some way

# Memory

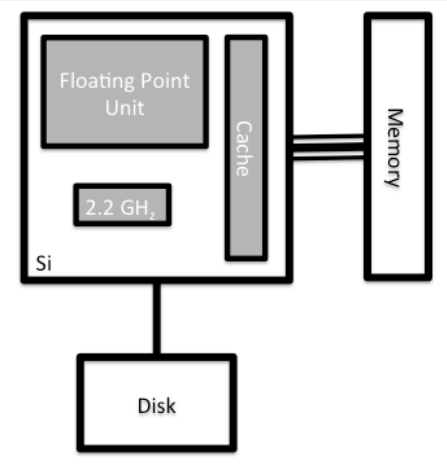
- Will be packaged with processor
  - Increases power efficiency, speed and bandwidth…
  - …at the cost of smaller memory per core
- Memory hierarchy will become more complex
  - Still unclear how this will be exposed to developers

# System on a chip

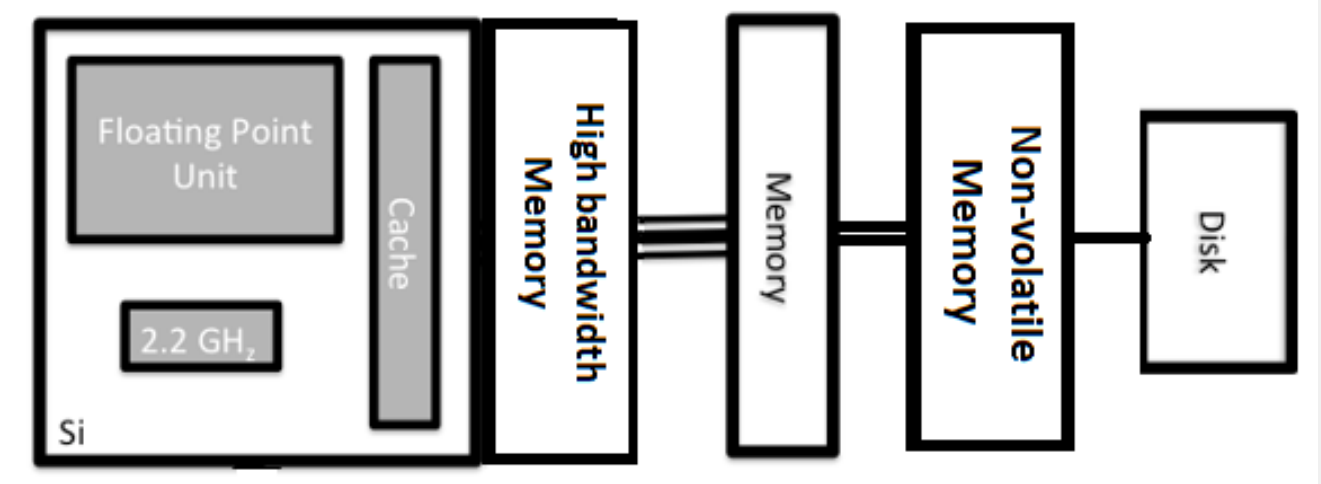
- Instead of separate:
  - Processor
  - Memory
  - Network interface
- Combined system package where all these things are included in one manufactured part
  - This is the only way to improve power efficiency
  - Less scope for customisation
  - If you need more memory than in package you will have to have levels of memory hierarchies

# Memory hierarchies

- Moving from:




- To something like:



# Software challenges

## What does software need to do to exploit future HPC?

# What does this mean for applications?

- The future of HPC (as for everyone else):
  - Lots of cores per node (CPU + co-processor)
  - Little memory per core
  - Lots of compute power via network interface
  - Increased complexity in memory and IO hierarchy
- The balance of compute to communication power and compute to memory are both radically different to now
- Must exploit parallelism at all levels
- Must exploit memory/IO hierarchy efficiently

# Algorithms

- For many problems new algorithms will be needed
- May not be optimal but contain more scope for parallelisation
- Mixed-precision will become more important

# Applications that do not scale

- The good news is that if you do not need to be able to treat larger/more-complex problems then you can access more of current resource size
  - May be caught out by decrease in memory per core!
  - Options to scale in trivial-parallel way: increase sampling (e.g. ensemble / swam / replica methods in MD), use more sophisticated statistical techniques
  - This may well be the best route for many simulations

# Impact on standard computing

## What does this mean for my workstation/laptop?

# Everything will be (even more) parallel

- All current computers are parallel
  - From supercomputers all the way down to mobile phones
  - Most parallelism is task-based on 4-8 cores – each application (task) runs on an individual core.
- In the future:
  - More parallelism per device – 10s to 100s cores running at lower clock speeds
  - All applications will have to be parallel
  - Parallel programming skills will be required for all application development.
- More system on a chip – more things will be packaged together

# Cloud Computing

- Cloud computing (AWS, Azure, etc.) has grown in use
- On-demand and flexible
- Not ideal for frequent transfer of very large amounts of data
  - Likely to be a bottleneck
  - On-site computing likely to remain important
- Suitable for high throughput but hardware architectures used in cloud computing offerings have mostly not had the quality network performance of HPC machines
- Microsoft Azure starting to offer on-demand cloud HPC

# Software Containers

- HPC machines starting to provide support for software containers (Docker, Singularity)
- These allow more freedom for user customisation of the environment, installed software, etc.
- Should facilitate sharing of workflows, software and libraries and running them on different platforms

# Summary

- Hopefully you should now have some understanding of HPC, its uses and users
  - Plenty more to learn!
- A lot of people use HPC without programming
  - Use available parallel programs and simulation packages
- Understanding HPC services and how you're intended to use them will enable you to get best use from them