Parallel and Cluster Computing

COMP 3036J

Dr. Brett Becker

brett.becker@ucd.ie

Please see course outline on Moodle for some high-level details on this module

A little about me

- Research areas:
  - Computer Science Education
  - Parallel and High Performance Computing
- More: www.brettbecker.com

A little about me

- Originally from New Jersey, USA
- Living in Dublin since 2000
- BA Physics, Drew University (NJ, USA)
- BA Computer Science, Drew University (NJ, USA)
- MSc Computational Science, University College Dublin
- PGCert Teaching and Education, Griffith College (Dublin)
- PhD Computer Science, University College Dublin
- MA Higher Education, Technological University of Dublin
- PGCert University Teaching and Learning (UCD)

A little about me

- Teaching since 2005
- Lecturer / Senior Lecturer, Griffith College Dublin
- Head of Faculty of Computing, College of Computing Technology, Dublin, Ireland
- Assistant Professor, School of Computer Science, University College Dublin & Beijing Dublin International College

A little about me

- Research Interests
  - High Performance Computing
    - Parallel Computing
    - Heterogeneous Computing
  - Computer Science Education
    - Novice Compilation Behaviour
    - Naturally Accumulating Process Data

A little about Brett

- I am also the maintainer of the Irish Supercomputer List
  - www.IrishSupercomputerList.org

China's Supercomputer List

- China also has a supercomputer list. Occasionally this is published in English
  - http://www.irishsupercomputerlist.org/2015/05/10/chinas-top-100-hpc-list-published-in-english/

Course Subject

- Parallel computing technologies
  - Aimed at *acceleration of solving a single problem on available computer hardware*
  - The course focuses on *software tools for developing parallel applications*: optimising compilers, parallel languages, parallel libraries
- Logical view vs. cook book
  - *Ideas, motivations, models* rather than *technical details*
  - We will follow the evolution of hardware architecture
  - Carefully selected programming systems

Course Outline

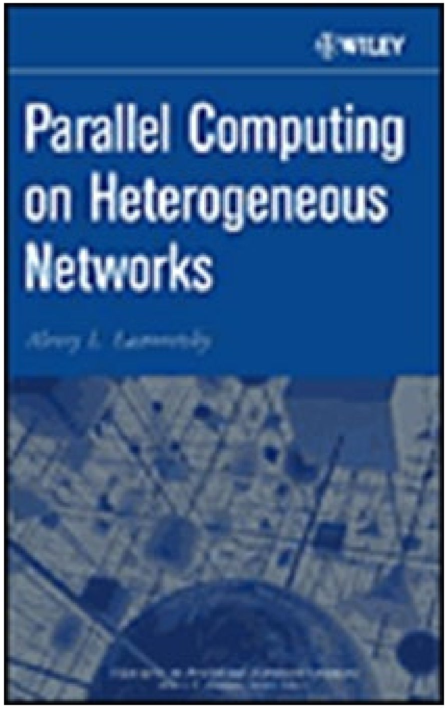
- Vector and Superscalar Processors
- Shared-Memory Multiprocessors
- Distributed-memory Multiprocessors
- Networks of Computers

Course Output

- You will be able to orient yourselves in parallel computing technologies
  - Mainly theoretical course
  - Some practical work later

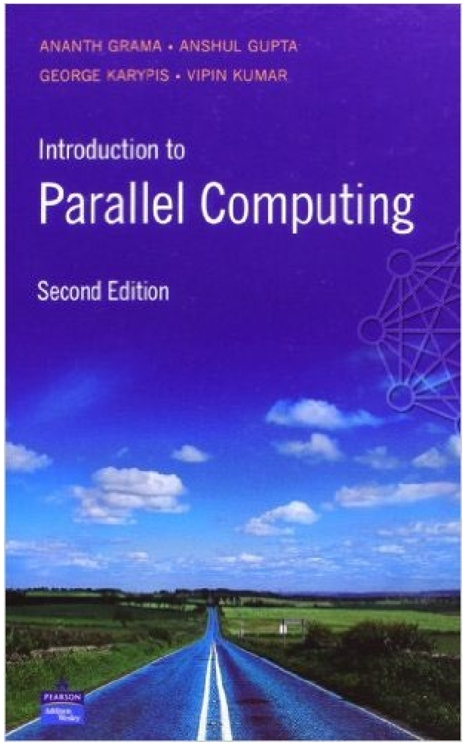
References

- Textbook 1
  - A.Lastovetsky. *Parallel Computing on Heterogeneous Networks*.
    John Wiley & Sons, 423 pp, June 2003,
    ISBN: 0-471-22982-2.



References

- Textbook 2
  - A. Grama, A. Gupta, G. Karypis, V. Kumar. *Introduction to Parallel Computing, Second Edition*.
    Pearson, 656 pp, January 2003, ISBN: 0201648652.



Course Material

- Slides, Examples & Resources:
  - All slides and examples will be available via UCD CS Moodle.
  - Videos will be available on the Tencent cloud (Weiyun). Links will be on Moodle.

Module Plan and Rules

- **See “Overview” on Moodle:**

Timetable

- **Lectures:**
  - **Tuesdays** @ 09:55-11:30
  - This is when you should review the week's ppt / video and other material. Each week the material will be on moodle by this time.
- **"Other time"**
  - **Thursdays** @ 18:00 – 19:35
  - This is time that you have dedicated to this class. You are guaranteed to have no conflicts during these times. We will not be doing anything "live" during these sessions – at least not to start.
  - However you are guaranteed to not have any other modules at this time. I suggest you use these times to study the course material and discuss the course with your classmates on WeChat, etc. As the module develops, we will have some live tutorials there. I will always post information on this in the Announcements Forum.

Assessment

- Continuous Assessment: **100%**
  - Right now a lot depends on if/when you return to BJUT

Assessments will vary according to the trimester but may include programming, lab work, reports and presentations as well as quizzes and in-class tests/exams.

  - Basically, right now we are going to wait and see.
  - This class is largely theoretical but there will be practical elements as well.

Plagiarism & UCD Computer Science

- **Plagiarism is a serious academic offence**
  - UCD Student Code of Conduct / UCD Student Plagiarism Policy / Computer Science Plagiarism Policy and Procedures
- Our staff & demonstrators are **proactive** in looking for possible plagiarism
- Suspected plagiarism is investigated by the CS Plagiarism subcommittee
  - Usually includes an interview with the student(s) involved
  - 1st offence: **typically** 0 or NM in the affected components
  - 2nd offence: more serious consequences e.g. UCD Disciplinary process
- Student who *enables* plagiarism is *equally responsible* for it
- *All students* in a group which plagiarises are held responsible for it
- **Examples** of plagiarism:
  - Copying some/all of the work of another student and submitting it as your own work
  - Copying some/all of an assignment from the Internet/book/etc without referencing it
  - Sharing individual work with another student (by e-mail, FB messenger, WhatsApp, …)
  - Making your work available (on GitHub, website, social media, …) before lecturer gives permission
  - A group of students working on a solution, then individually submitting the same work
  - Students collaborating at too detailed a level e.g. consulting each other after implementing a line/block/segment of code and sharing the results

High Performance Computing (HPC)

Overview

High Performance Computing (HPC)

- High-performance computing (HPC) is the use of parallel processing (often a cluster) for running programs efficiently, reliably and quickly.
  - Efficiency, reliability and speed however are three specifically different things!

- The term HPC applies especially to systems that function at several petaflops or more.
  - >2 years ago I used to say teraflops, but now it's petaflops!
  - A petaflop is 10^{15} floating-point operations per second.
  - Many GPU cards are now capable of more than 1 teraflop.

High Performance Computing (HPC)

- The term HPC is occasionally used as a synonym for supercomputing, although technically a supercomputer is a system that performs at or near the currently highest operational rate for computers.
- Most supercomputers work at more than a petaflop. Currently the 500 fastest computers on earth are all > 1.3 Petaflop/s.
- The top 500 supercomputers on Earth are tracked by the ‘Top500 List’: www.top500.org

Where is the fourth fastest computer on Earth?

- China ☺
  - Until recently it was the fastest!
  - National Supercomputing Center in Wuxi
- Sunway TaihuLight
  - 10,649,600 cores @ 1.45GHz across 40,960 nodes
  - 1,310,720 GB RAM
  - Power: 15,371 kW ~= 15 MW
    - This is equivalent to about 15,000 homes!
    - A typical coal power plant would provide about 600 MW.
  - R_{max} (performance) > 93 petaflops

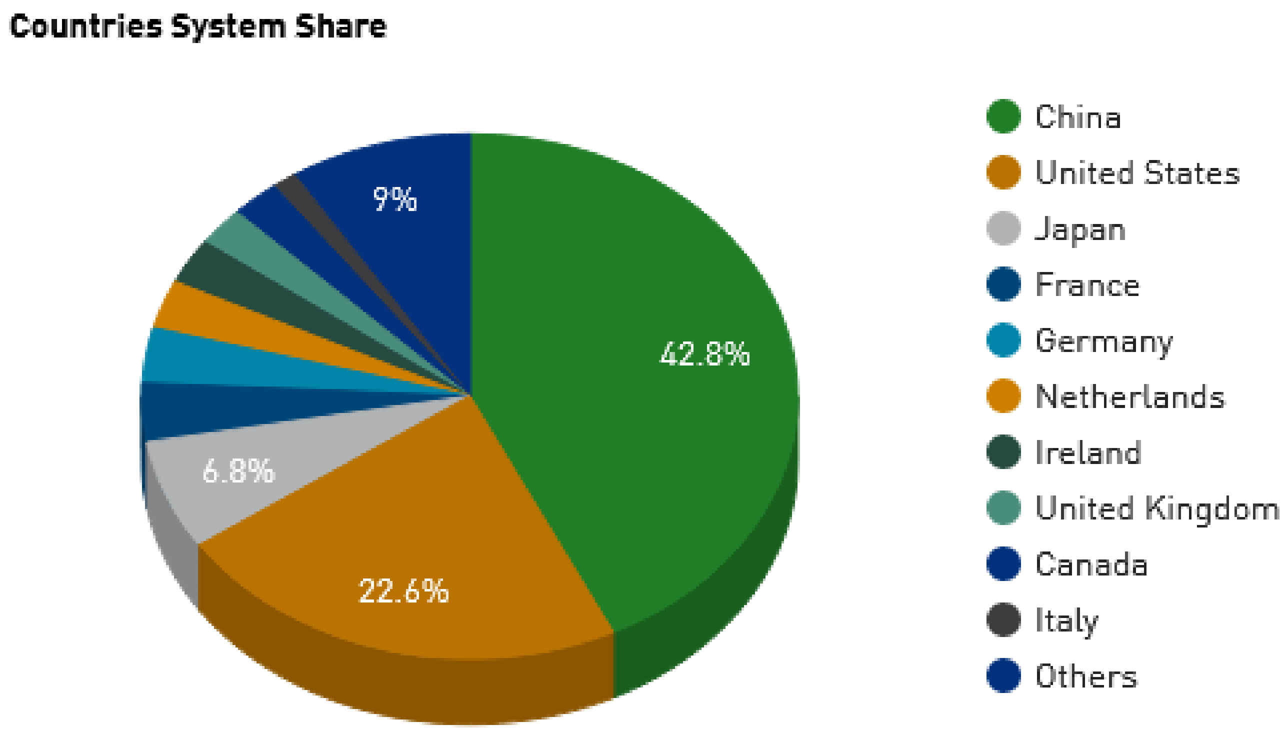
Sunway TaihuLight

Where is the sixth fastest computer on Earth?

- China!
  - National Super Computer Center in Guangzhou
- Tianhe-2A
  - 4,981,760 cores @ 2.20GHz
  - 1,024,000 GB RAM
  - Power: 16,482 kW
  - R_{max} (performance) > 61 petaflops

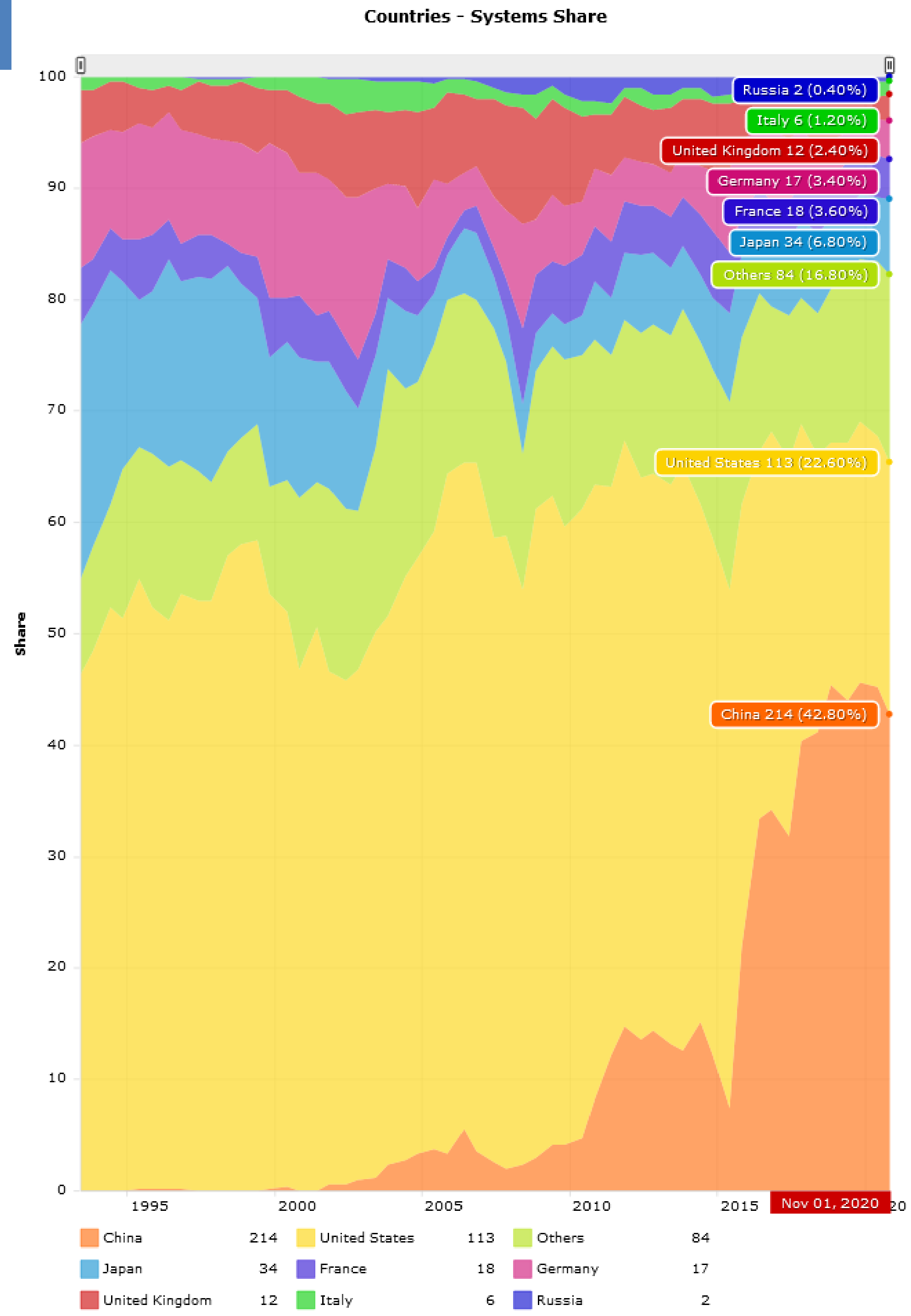
Tianhe-2

China on the Top500 List



China on the Top500 List

- As of November 2020, China has 226 Top500 systems.



China on the Top500 List

- As of November 2020, China has 214 systems

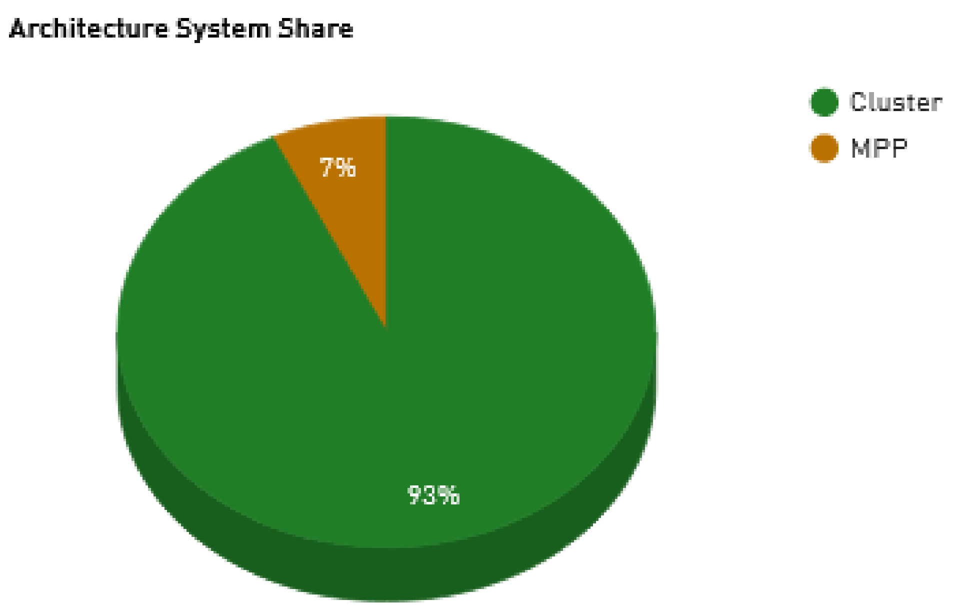
	Countries	Count	System Share (%)	Rmax (GFlops)	Rpeak (GFlops)	Cores
1	China	214	42.8	566,635,422	1,231,757,033	30,704,256
2	United States	113	22.6	668,704,300	942,339,598	15,373,432
3	Japan	34	6.8	593,700,080	766,195,745	10,947,524
4	France	18	3.6	89,828,330	135,469,318	2,669,472
5	Germany	17	3.4	131,048,770	197,689,472	2,664,446
6	Netherlands	15	3	24,736,650	31,795,200	864,000
7	Ireland	14	2.8	23,087,540	29,675,520	806,400
8	United Kingdom	12	2.4	34,067,502	44,283,532	1,248,840
9	Canada	12	2.4	26,698,060	47,707,321	716,096
10	Italy	6	1.2	78,529,000	114,511,528	1,447,536
11	Saudi Arabia	5	1	35,997,040	76,126,574	1,084,020
12	Brazil	4	0.8	10,991,000	19,270,566	214,040
13	Singapore	4	0.8	6,596,440	8,478,720	230,400
14	South Korea	3	0.6	18,720,660	31,496,620	709,220
15	Taiwan	3	0.6	12,622,710	21,651,750	247,952
16	Switzerland	3	0.6	26,215,350	33,955,305	581,140
17	India	3	0.6	10,953,340	12,082,026	244,488
18	Norway	3	0.6	8,023,010	10,432,510	287,232
19	Finland	2	0.4	7,095,250	9,748,685	209,728
20	United Arab Emirates	2	0.4	9,013,750	12,164,803	142,368

Other Countries

- Currently China has 214 systems on the Top500 list.
  - United States has 113.
  - Japan -> 34
  - France -> 18
  - Germany -> 17
  - Netherlands -> 15
  - Ireland -> 14
  - UK -> 12
  - Canada -> 12
  - All other countries <= 6

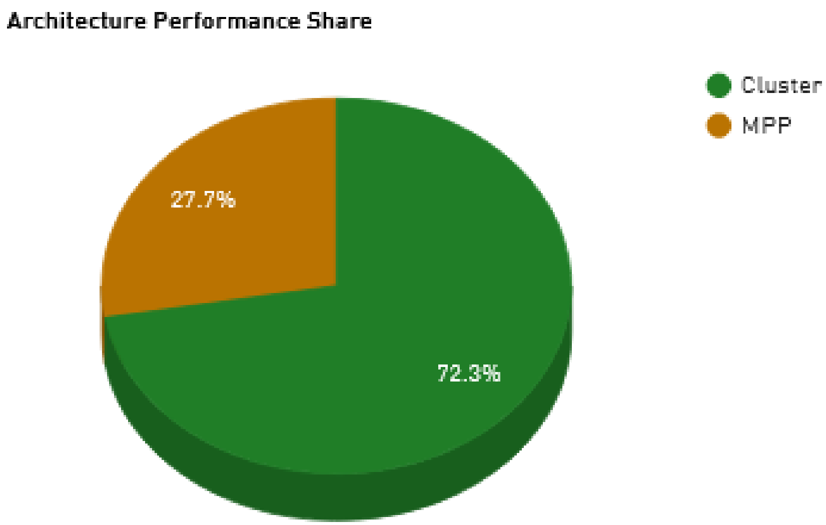
Clusters on the Top500

- Most of the machines on the Top500 are clusters.



Clusters on the Top500

- Most of the performance on the Top500 are from clusters.



Cluster - Definition

- A **cluster** is a parallel computer system comprising an integrated collection of independent nodes, each of which is a system in its own right, capable of independent operation and derived from products developed and marketed for other stand-alone purposes
- In short, you could unplug a node from a cluster and it would work…

MPP (Massively Parallel Processor) - Definition

- An MPP is more tightly-integrated than a cluster. Individual nodes cannot run on their own and they are connected by a custom network (these are often complicated and expensive).
- But, similarly to a cluster, an MPP doesn't have shared memory.

Parallel Computing

- **Parallel computing** is a type of computation in which many calculations or the execution of processes are carried out simultaneously.

- Parallel computing is closely related to concurrent computing—they are frequently used together, and often confused or used interchangeably.

Parallel Computing

- In parallel computing, a computational task is typically broken down in several, often many, very similar subtasks that can be processed independently and whose results are combined afterwards, upon completion.
- In contrast, in concurrent computing, the various processes often do not address related tasks; when they do, the separate tasks may have a varied nature and often require some inter-process communication during execution.

Distributed Computing

- While we are on that topic, distributed computing is more loosely coupled than parallel computing. **A significant characteristic of distributed computing is the lack of a global clock**. It is possible to have a distributed computing model that has minimal, or even no, inter-process communication requirements.
- These are only examples of the many definitions for parallel, concurrent, and distributed computing.
- In general the Wikipedia articles are pretty good:
  - Parallel computing
  - Concurrent computing
  - Distributed computing

Distributed vs. Shared Memory

- A defining characteristic of cluster computing is that memory is *distributed* (distributed memory).
  - **Note – don't confuse this with *distributed computing!***
  - Distributed memory means each processor has its own memory
  - In order to share information processors must pass *messages* over some form of interconnect – normally a network.
- Shared memory means that there is a single memory space used by all processors.
  - Processors can share information by using this shared memory.
  - There is no need for *messages*
  - There is no need for a network

Distributed vs. Shared Memory

- Most of the time we will be discussing distributed memory architectures, namely clusters.
- Note that there is a hybrid possibility – *distributed shared memory*, where each node of a cluster has access to shared memory in addition to a private non-shared (and thus distributed) memory.
- This is fairly exotic and not used much in practice

High Performance Computing – Why?

- What types of problem require high performance computing?
  - **Compute Intensive**: A single problem requiring a large amount of computation.
  - **Memory Intensive**: A single problem requiring a large amount of memory.
  - **Data Intensive**: A single problem operating on a large amount of data.
  - **High Throughput**: Many unrelated problems to be executed over a long period.

High Performance Computing – Why?

- What types of problem require high performance computing?
  - **Biology**: gene sequencing, genome mapping, drug discovery, protein folding …
  - **Physics**: stellar and galactic evolution, condensed matter physics, weather prediction, …
  - **Finance**: financial forecasting, …
  - **Industry**: Oil & Gas exploration, …
  - **Entertainment**: The latest Disney animated film used *millions* of CPU hours.
  - The list goes on and on…

High Performance Computing – Why?

- Goodyear tests all of its tyres on HPC platforms before it builds real prototype tyres
- Crash-test dummies and automobile safety testing has been revolutionised by HPC, saving more and more lives each year
- The Airbus A380 was designed, 'built' and 'flew' on HPC <u>before any parts were ordered</u>

High Performance Computing – Why?

- Disney (and other 'animated' films) are not possible as we know them without HPC

- Without HPC resources, artists would not be able to design and see the results in the necessary (almost real-time) timeframes that are required for production

- If only normal computers were available, the time to make movies like Big Hero 6 and Frozen would be many, many years, not months

http://www.awn.com/news/supercomputing-boosts-disneys-big-hero-6

High Performance Computing – Why?

- The economic benefits of investments in high-performance computing are significant.

- As of August 2015, the research firm IDC found that U.S. enterprises' investments in high performance computing systems generate $515 in revenue and $43 in profits and/or cost savings per dollar of HPC investment

Ezell, Stephen J., and Robert D. Atkinson. "The Vital Importance of High-Performance Computing to US Competitiveness." *Information Technology and Innovation Foundation, April* 28 (2016).

High Performance Computing – Why?

- A study of HPC in the European Union (EU) found even higher returns, concluding that each euro invested in HPC on average returned €867 in increased revenue and €69 in profits.
- Total increased revenue for the 59 HPC-enabled, quantifiable projects in Europe reached €133.1 billion, or about €230 million per project on average.

Ezell, Stephen J., and Robert D. Atkinson. "The Vital Importance of High-Performance Computing to US Competitiveness." *Information Technology and Innovation Foundation, April* 28 (2016).

High Performance Computing – Why?

- From a representative at Proctor & Gamble:
  - I can figure out…
    - whether a bottle will break when it drops
    - how the handle will fit small hands and big hands
    - whether a diaper will leak
    - whether the closure on a diaper will mark a baby's leg because the elastics are too tight
    - whether a formula will remove a stain and still protect a new fabric
    - how many washes will it take for jeans to fade
    - can we smell a new perfume on laundry after it's been washed?
  - All of those things we now do with high-performance computing"

Ezell, Stephen J., and Robert D. Atkinson. "The Vital Importance of High-Performance Computing to US Competitiveness." *Information Technology and Innovation Foundation, April* 28 (2016).

High Performance Computing – Why?

- The Center for Pediatric Genomic Medicine at Children’s Mercy Hospital in Kansas City, Missouri, has been using HPC to help save the lives of critically ill children.
- In 2010, the center’s work was identified as one of Time Magazine’s top 10 medical breakthroughs of the year.
- Roughly 4,100 genetic diseases affect humans, and they are the main causes of infant deaths.
  - But identifying which genetic disease is affecting a critically ill child is extremely difficult.
  - For one infant suffering from liver failure, the center used 25 hours of supercomputer time to analyze 120 billion nucleotide sequences and narrowed the cause of the illness down to two possible genetic variants. Thanks to this highly accurate diagnosis, the baby is alive and well today.

Ezell, Stephen J., and Robert D. Atkinson. "The Vital Importance of High-Performance Computing to US Competitiveness." *Information Technology and Innovation Foundation, April* 28 (2016).

High Performance Computing – Why?

- For more on the applications of HPC, see:

http://sc16.supercomputing.org/exhibitors/hpc-impact-showcase/

https://ec.europa.eu/digital-single-market/en/blog/why-do-supercomputers-matter-your-everyday-life

High Performance Computing – The Possibilities

- In 2013 a supercomputer read 100,000 research papers in 2 hours
  - **All of these papers were up to the year 2003**
  - **The researchers intentionally didn't give the computer newer papers.**
- The computer was programmed to look for information on a protein called p53, which suppresses tumors in humans, and a class of enzymes that interact with p53 called kinases
- **The computer identified 7 of the 9 kinases that were discovered by scientists after 2003**
- **It also identified 2 kinases that were previously unknown in 2013**

http://www.newscientist.com/article/mg22329844.000-supercomputers-make-discoveries-that-scientists-cant.html#.VADnHvmwJcQ

Where is HPC going?

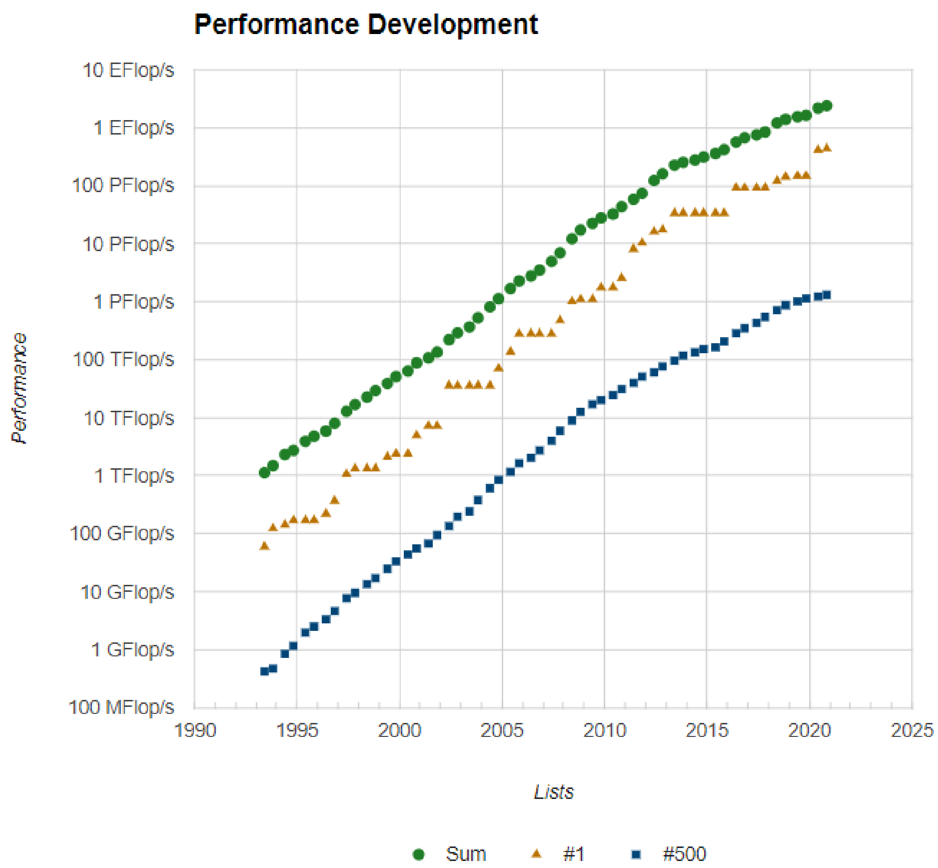
- Right now there is great excitement about *exascale* computing.

- An exascale computer is one capable of one exaflop (one billion billion calculations per second)
  - Or, a thousand petaflops or a quintillion (10^{18}) floating point operations per second

- Currently, the Supercomputer Fugaku is 0.44 exaflops, or almost half of an exaflop

Where is HPC going? Up!

- https://www.top500.org/statistics/perfdevel/



Where is HPC going?