A presentation on:

"Large-scale multiprocessors and scientific applications"

Appendix I from Computer Architecture: A Quantitative Approach, John L. Hennessy, David A. Patterson

BY MOHAMMED SHAHZAD

444105788@STUDENT.KSU.EDU.SA

Contents of this presentation

- Introduction
- Characteristics of large scale multiprocessors
- Performance measurement
- Example Large-scale Multiprocessor Blue Gene/L
- Conclusion

Introduction 1of3

- **What are Large-scale Multiprocessors?**

- Large-scale multiprocessors aid in parallel computing.
- They consist of several processors bound together to run multiple tasks simultaneously.
- Parallel computing is a type of computation in which many tasks or processes are broken down into smaller tasks and carried out simultaneously.

Figure: Blue Gene P by IBM

Introduction 2of3

- **Why Large-scale Multiprocessors?**
- The primary target of parallel computing is scientific and technical applications.
- Scientific applications require use of large-scale parallel processing to attain results
- We will see the characteristics of such applications and multiprocessing ahead.
- <FIG>

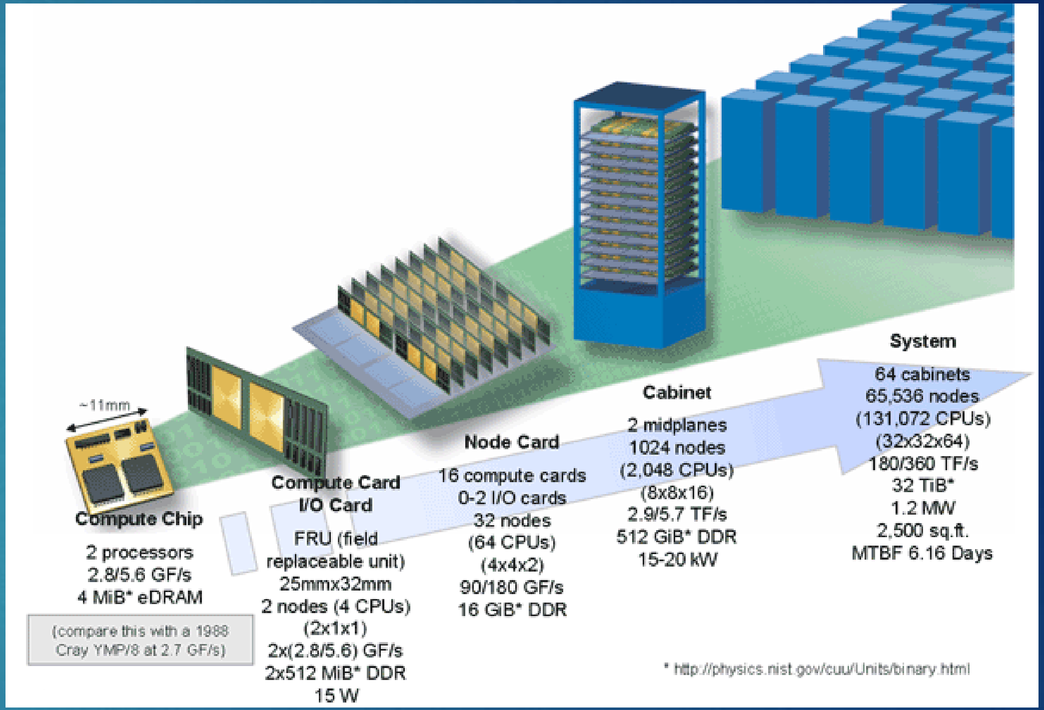


Figure: organization of Blue Gene

Introduction 3of3

- **Important performance issue in Multiprocessing-**

  Interprocessor communication: the critical performance issue

- Interprocessor communication depends on
  - Bandwidth
  - Latency
  - Latency hiding

- The critical performance issue is that, in multiprocessors with larger processor counts, interprocessor communication becomes more expensive.

- As the distance between processors(several hundreds) increases communication becomes expensive.

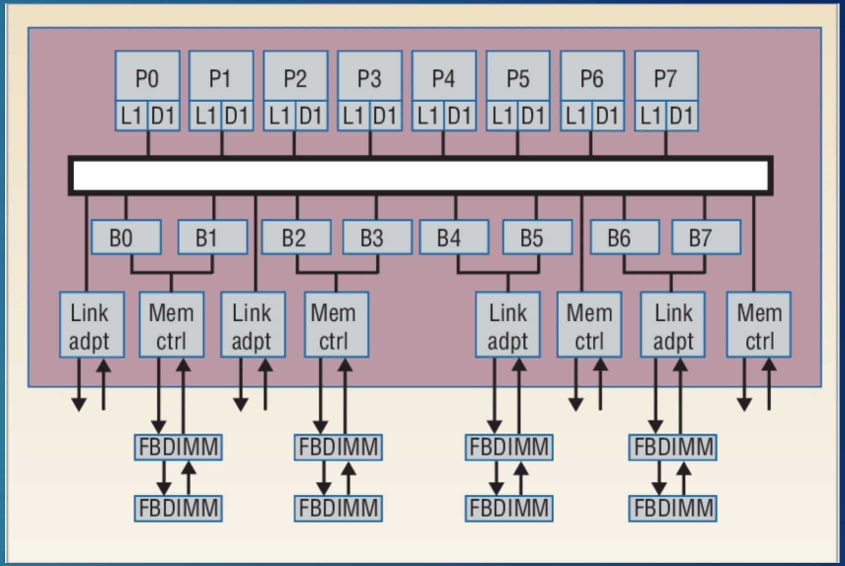


Figure: A typical supercomputer has thousands of processors and communication channels between them.

Characteristics of large scale multiprocessors 1of9

- In this section we will study the following:
  - **Characteristics of scientific applications under study**
    - FFT, LU, Barnes and Ocean as prototype applications
  - **Characteristics of Large-scale Multiprocessors for such scientific applications**

Characteristics of large scale multiprocessors 2of9

- **Characteristics of scientific applications:**

- The primary target of parallel computing is scientific and technical applications.
- The kernels commonly used are Fast Fourier Transformation (FFT) and Lu decomposition
- They were chosen because they represent commonly used techniques in a wide variety of applications and have performance characteristics typical of many parallel scientific applications.

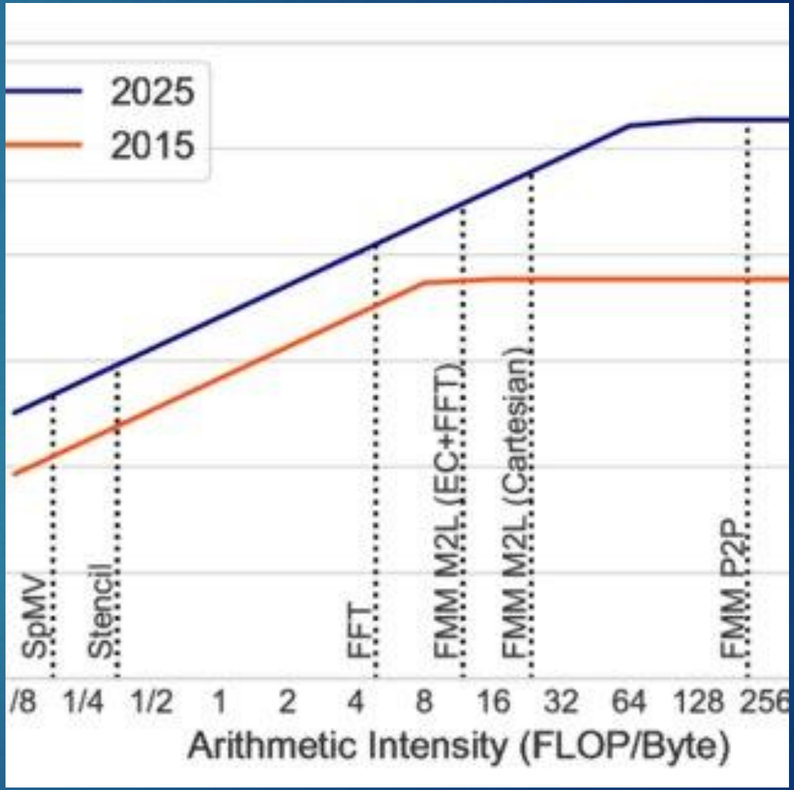


Figure: Arithmetic intensity of FFT

Characteristics of large scale multiprocessors 3of9

- **Characteristics of scientific applications: FFT and LU Kernels**
- **The FFT Kernel** The FFT is the key kernel in applications that use spectral methods, which arise in fields ranging from signal processing to fluidflow to climate modeling.
- **The LU Kernel** LU is an LU factorization of a dense matrix and is representative of many dense linear algebra computations, such as QR factorization, Cholesky factorization, and eigenvalue methods.

Characteristics of large scale multiprocessors 4of9

- **Characteristics of scientific applications: Barnes and Ocean applications**

- **The Barnes Application** Barnes is an implementation of the Barnes-Hut n-body algorithm solving a problem in galaxy evolution.
- N-body algorithms simulate the interaction among a large number of bodies that have forces interacting among them.
- **The Ocean Application** Ocean simulates the influence of eddy and boundary currents on large-scale flow in the ocean.
- It uses a restricted red-black Gauss-Seidel multigrid technique to solve a set of elliptical partial differential equations.

Characteristics of large scale multiprocessors 5of9

- **Takeaways from this sub-section**

- Characteristics of Scientific Applications:
  - FFT, LU are ideal prototype kernels for large-scale multiprocessing applications
  - Barnes and Ocean are ideal prototype applications for large-scale multiprocessing appl.

  - **Computation/Communication for the Parallel Programs** A key characteristic in determining the performance of parallel programs is the ratio of computation to communication.
  - If the ratio is high, it means the application has lots of computation for each data communicated.

Characteristics of large scale multiprocessors 6of9

- **Characteristics of Large-scale Multiprocessors for such scientific applications**

- **1. Support Synchronization: Scaling Up**

- Problem: Suppose we have10 processors and each tries to lock a variable simultaneously.

Example Suppose there are 10 processors on a bus and each tries to lock a variable simultaneously. Assume that each bus transaction (read miss or write miss) is 100 clock cycles long. You can ignore the time of the actual read or write of a lock held in the cache, as well as the time the lock is held (they won't matter much!). Determine the number of bus transactions required for all 10 processors to acquire the lock, assuming they are all spinning when the lock is released at time 0. About how long will it take to process the 10 requests? Assume that the bus is totally fair so that every pending request is serviced before a new request and that the processors are equally fast.

Answer When i processes are contending for the lock, they perform the following sequence of actions, each of which generates a bus transaction:

i load linked operations to access the lock

i store conditional operations to try to lock the lock

1 store (to release the lock)

Thus, for i processes, there are a total of $2i+1$ bus transactions. Note that this assumes that the critical section time is negligible, so that the lock is released before any other processors whose store conditional failed attempt another load linked.

Thus, for n processes, the total number of bus operations is

$$\sum_{i=1}^{n}(2i+1) = n(n+1)+n = n^2+2n$$

For 10 processes there are 120 bus transactions requiring 12,000 clock cycles or 120 clock cycles per lock acquisition!

Figure: Synchronization Problem in Multiprocessing

Characteristics of large scale multiprocessors 7of9

- **Characteristics of Large-scale Multiprocessors for such scientific applications**
- **2. Manage synchronization failures**
  - Synchronization Mechanisms for Larger-Scale Multiprocessors
- **Software Implementations** The major difficulty with our spin lock implementation is the delay due to contention when many processes are spinning on the lock.
  - One solution is to artificially delay processes when they fail to acquire the lock.
- **Hardware primitive**
  - One primitive that has been introduced is fetch-and-increment, which atomically fetches a variable and increments its value.

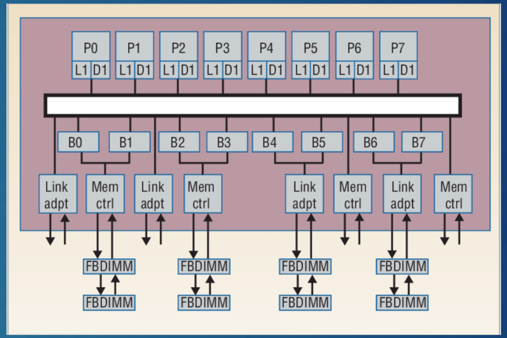
```
local_sense =! local_sense; /* toggle local_sense */
fetch_and_increment(count);/* atomic update */
if (count==total) {/* all arrived */
        count=0;/* reset counter */
        release=local_sense;/* release processes */
}
else {/* more to come */
        spin (release==local_sense);/* wait for signal */
}
```

Figure I.7 Code for a sense-reversing barrier using fetch-and-increment to do the counting.

Characteristics of large scale multiprocessors 8of9

- **Characteristics of Large-scale Multiprocessors for such scientific applications**

- **3. Maintain Cache coherency**
  - Approach will be to try to resolve the need for multiple resources. The strategy has four parts:
    - 1. A separate network (physical or virtual) is used for requests and replies. This ensures that new requests cannot block replies that will free up buffers.
    - 2. Every request that expects a reply allocates space to accept the reply when the request is generated.
    - 3. Any controller can reject with a NAK any request, but it can never NAK a reply. This prevents a transaction from starting if the controller cannot guarantee that it has buffer space for the reply.
    - 4. Any request that receives a NAK in response is simply retried.



*10000

*

Figure: Maintaining cache coherency

Characteristics of large scale multiprocessors 9of9

- **Takeaways from this sub-section:**

- Large-scale Multiprocessors should:
  - Support Synchronization: scale up
  - Manage Synchronization failures
  - Maintain cache coherency

Performance measurement 1of2

- Figure shows how the miss rates change as the cache size is increased, assuming a 64-processor execution and 64-byte blocks. These miss rates decrease at rates that we might expect

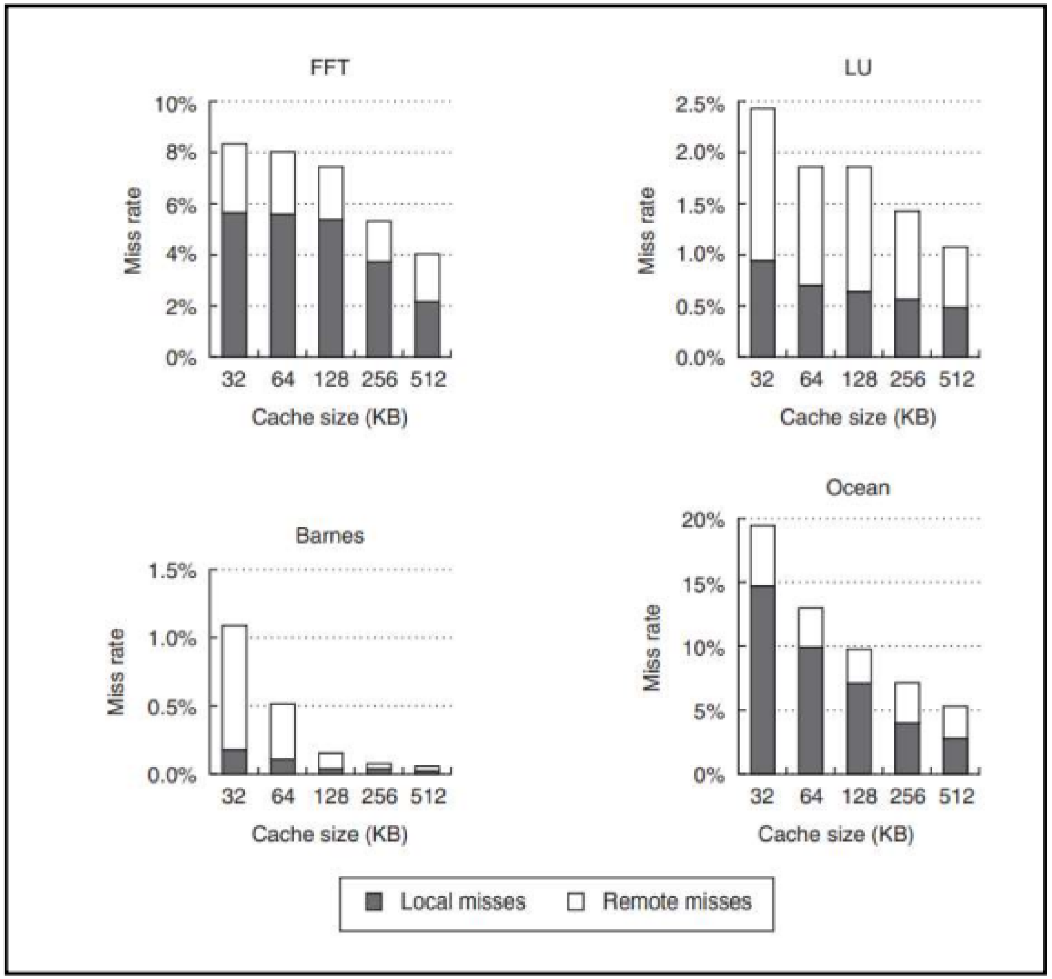


Figure : Miss rates decrease as cache sizes grow.

Performance measurement 2of2

- We examine the effect of changing the block size and increasing processor count in figures.

Although the drop in miss rates with longer blocks may lead you to believe that choosing a longer block size is the best decision, the bottleneck in bus-based multiprocessors is often the limited memory and bus bandwidth.

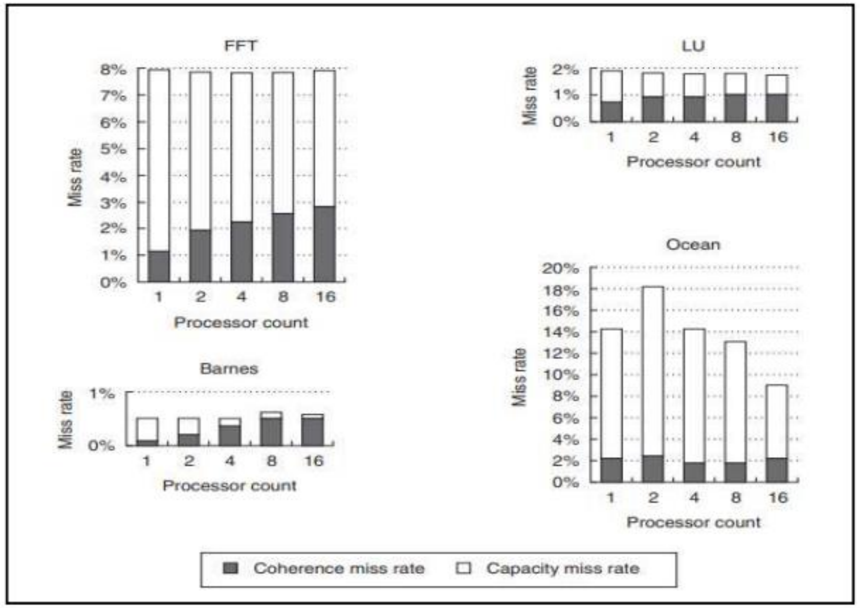


Figure 2: Data miss rates for processor count 1 to 16

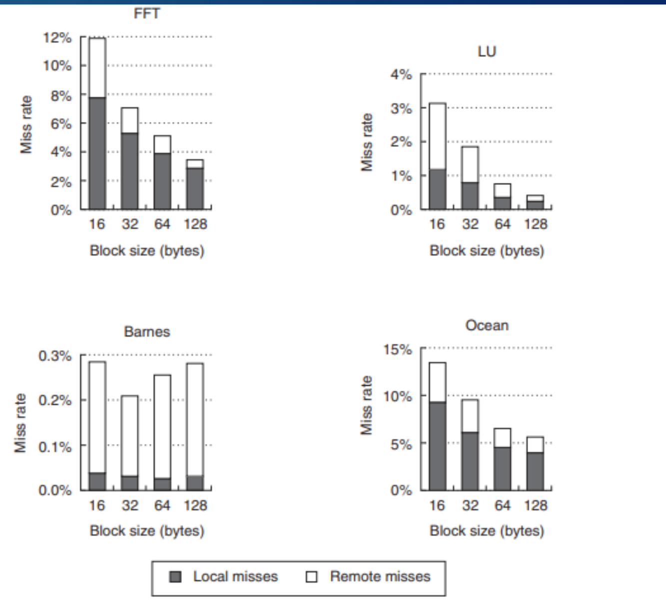


Figure I.14 Data miss rate versus block size assuming a 128 KB cache and 64 processors in total. Although difficult to see, the coherence miss rate in Barnes actually rises for the largest block size, just as in the last section.

- Due to good spatial locality, increases in block size reduce the miss rate

EXAMPLE Large-scale multiprocessor: The Custom Cluster Approach: Blue Gene/L 1of2

- Blue Gene/L (BG/L) is a scalable message-passing supercomputer whose design offers unprecedented computing density measured by compute power per watt.
- By focusing on power efficiency, BG/L also achieves unmatched throughput per cubic foot.
- High computing density, combined with cost-effective nodes and extensive support for RAS, allows BG/L to efficiently scale to very large processor counts.

Figure: Blue Gene/P by IBM

EXAMPLE Large-scale multiprocessor: The Custom Cluster Approach: Blue Gene/L 2of2

- Each BG/L node consists of a single processing chip and several SDRAM chips. The BG/L processing chip, shown in figure, contains the following:
- BG/L consists of up to 64 K nodes organized into 32 racks each containing 1 K nodes in about 50 cubic feet.
- The density intern allows the interconnection networks to be low latency, high bandwidth, and quite cost effective

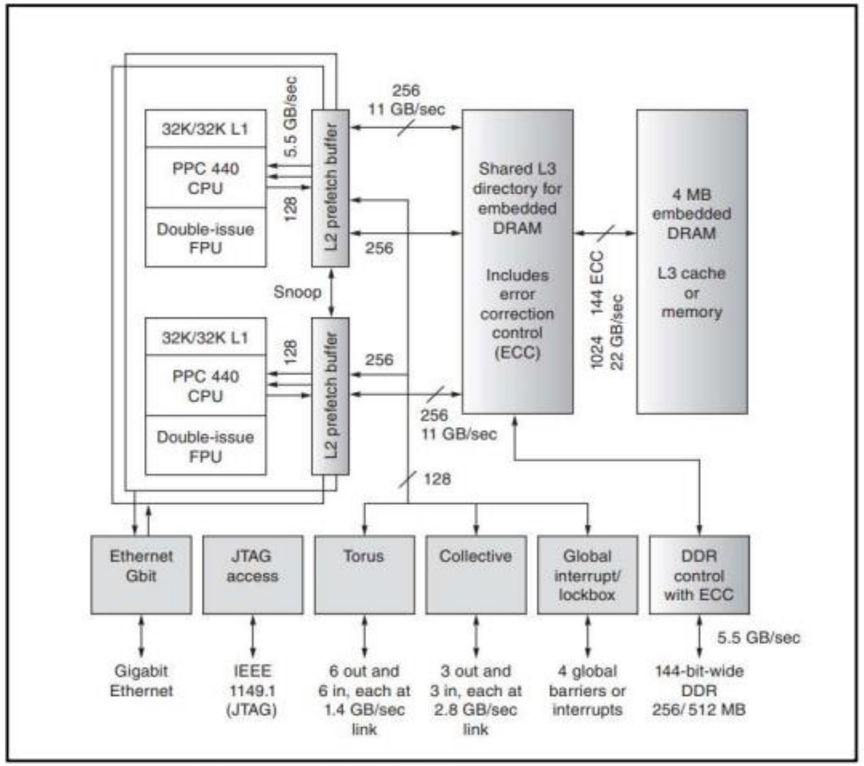


Figure : The BG/L processing node

Conclusion 1of2

- The landscape of large-scale multiprocessors has changed dramatically over the past five to ten years.

DATE	GFLOP/S (R_MAX)	COMPUTER
June 1993	60	TMC CM-5/1024
November 1993	124	Fujitsu Numerical Wind Tunnel
June 1994	143	Intel XP / S140
November 1994	170	Fujitsu Numerical Wind Tunnel
June 1996	220	Hitachi SR 2201 / 1024
November 1996	368	Hitachi CP-PACS / 2048
June 1997	2,379	Intel ASCI Red
November 2000	7,226	IBM ASCI White, SP Power 3
June 2002	35,860	NEC Earth Simulator
November 2004	478,200	IBM BlueGene / L
August 2008	1,105,000	IBM Roadrunner
November 2009	1,759,000	Cray Jaguar – XT5-HE Opteron
November, 2010	2,566,000	NUDT Tianhe 1A
June 2011	10,510,000	Fujitsu K Computer, SPARC64
June 2012	16,324,751	IBM Sequoia BlueGen / Q
November 2012	17,600,000	Cray Titan – XK7 Opteron
June 2013	33,860,000	NUDT Tianhe 2

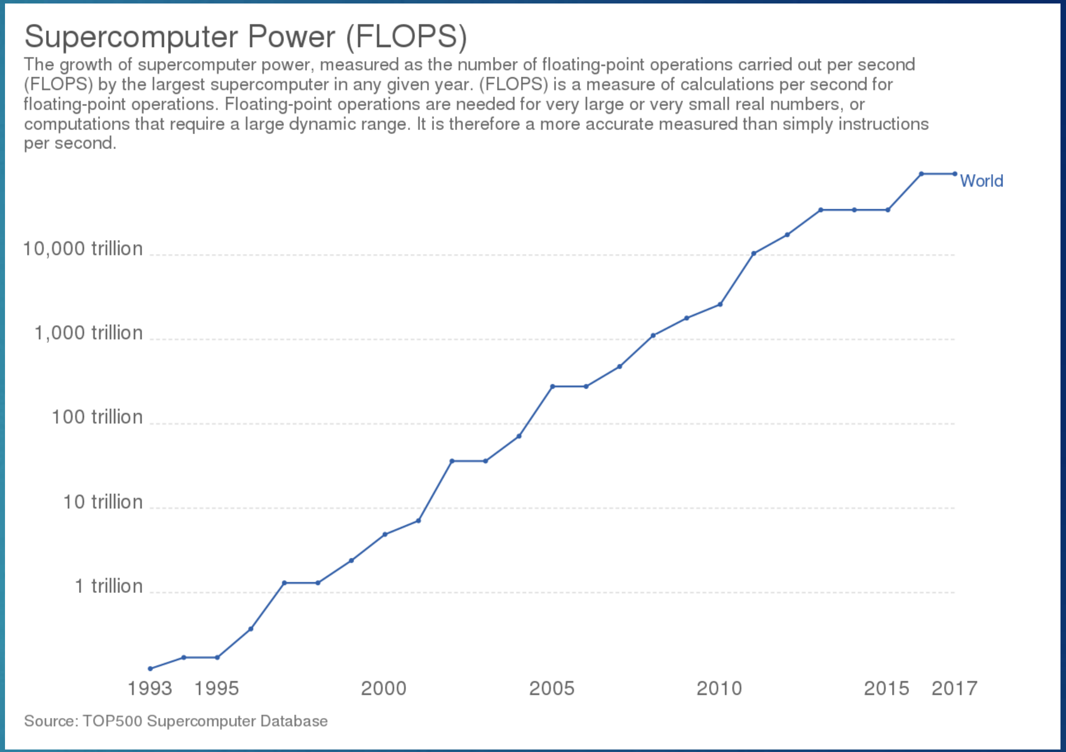


Figure: Multiprocessors Power (FLOPS) rising standards over time

Conclusion 2of2

- Software, both applications and programming languages and environments, remains the big challenge for parallel computing, just as it was 30 years ago.

- Until better progress is made on this front, convergence toward a single programming model and underlying architectural approach will be slow or will be driven by factors other than proven architectural superiority.

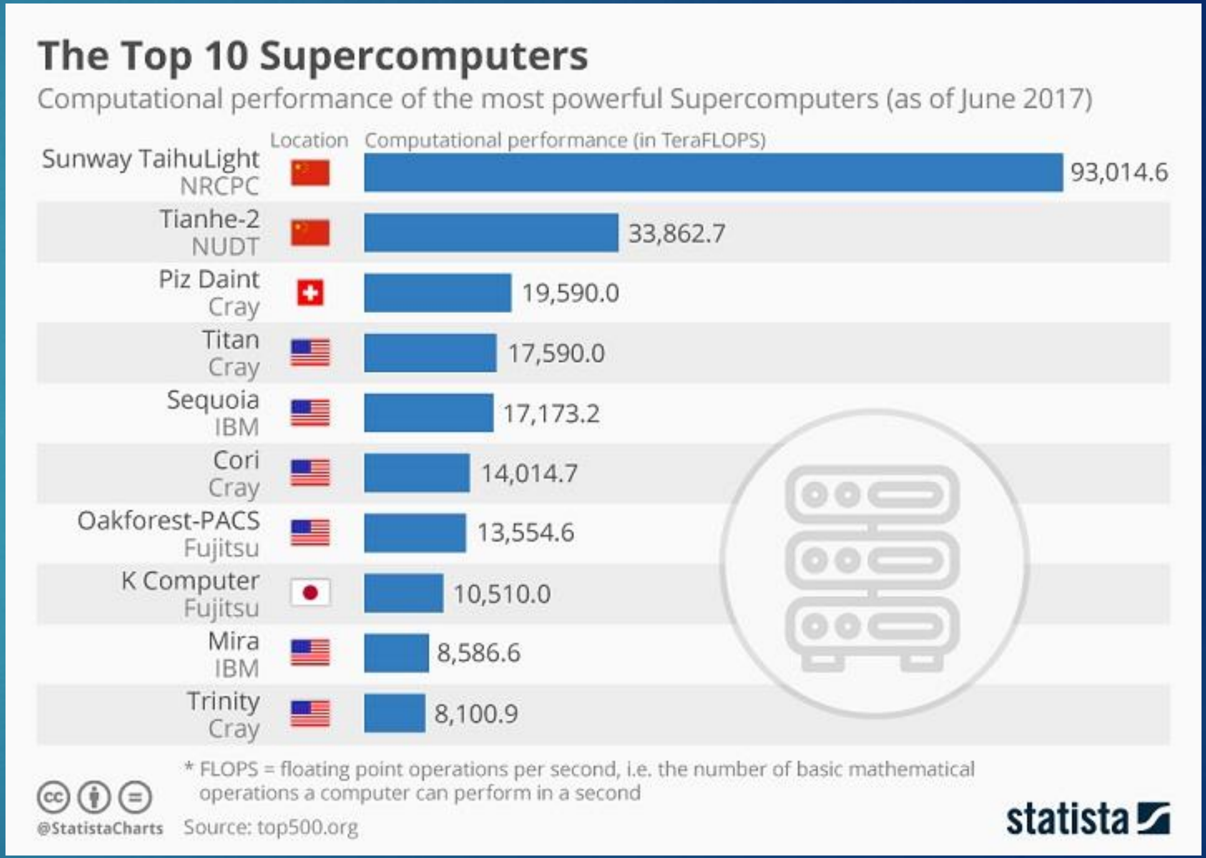


Figure: Fastest Supercomputers as of 2017 by country

Thank You!