The Path To Exascale Computing

R. Alshahrani
Department of Computer Science, Kent State University, Kent, OH, USA
Ministry of Higher Education, Riyadh, Saudi Arabia

Abstract—Exascale supercomputers are the future of Cluster computing. In this paper we discuss the challenges of developing exascale supercomputers and provide suggestions on how to deliver the required performance from these new machines. The major point is that the current programming systems over valued the flops and ignore the data locality and data movement which becomes increasingly important. A cheaper innovative technologies are needed to improve the memory bandwidth and density for a better data management.

Keywords: HPC, Exascale

1. Introduction

Advances in science led the scientists to face mountains of data that needs to be computed and stored. On the other hand, the notion of supercomputing and High Performance Computers (HPC) allows for more scientific breakthroughs. For example, Petascale supercomputers are able to reach performance of one petaflops which is used for advanced computation in diverse fields such as quantum chemistry, brain and weather simulations [1]. According to Moor’s law, the data is doubling from year to year which urge the need for more advanced supercomputers with higher speed and advanced capabilities. Delivering a system with exaFLOP capability is significant for more scientific discoveries in different areas. Exascale computing is the next generation of supercomputing. It will be capable of performing at least one exaFLOPS which means $10^{18}$ operations per second, a thousandfold increase over its counterpart petascale supercomputer [2].

The advancement from petascale to exascale computing is not easy. Many companies are competing to present the first supercomputer with exaFLOPs capability. However, the development of such a powerful system is constrained by many factors such as power, memory and cost. Based on the improvement chart for 20 years, the Top500 expected the first exascale supercomputer to see the light not before 2020 and maybe zetascale supercomputer by 2029 [3].

Developing such a powerful system is possible but with some difficult challenges. Apparently, the current technologies are limiting the developers of exascale supercomputers, hence, new innovative technologies are needed to come up with the first exascale supercomputer. The leading design constraint is the power efficiency as discussed in section 3. Improving the performance while lowering down the energy consumption is a challenging task. The processor can consume upto 30%. Add to that the data movement which became a significant source of power consumption.

The challenges involve frequency improvement in future machines [4]. In the old days, clock frequency used to be the main constraints of performance improvement. However, with parallelism that has became not a big issue, the only way to increase performance is to increase parallelism. Parallelism is growing by an exponential rate within a chip. The next generation with supercomputing will keep on the same track of parallelism which is discussed in more details in section 2. In addition to the previous challenges, the memory constrains the performance of the future machines. The performance of the CPU is limited by the memory speed. therefore, adding more cores without improving the memory will not improve the performance. The memory technology improvements are slowing down [3], [4]. For example, DRAM technology have not changed for the last two decades [4].

More innovative solutions are needed to produce the next generation of supercomputing. The cost of data movement within the supercomputers became dominant since the cost of data movement is exceeding the cost of performing a floating point operation [5], [2]. The overall system should be optimized to decrease the data movement costs. Additionally, the energy cost for moving data is not improving in terms of the cost of the flop. Therefore, getting the applications more aware of the data locality will lower the cost of the data movement [5].

Since the HPC are extensively used in scientific communities, the characteristics of the scientific computations must drive the fundamentals of the exascale computing design [6]. The benefit of building such a powerful system is not limited to the scientific computing. Building effective exascale systems allowed for further advances such as in cellphone performance and voice recognition. The new technologies that are needed to develop exascale supercomputers will open the doors for new innovations. In order to do what we want do, the fundamental architecture should be different because it will be influenced by the workload and the power requirements of HPC. The power consumption should not exceed 20MW and the cost should be limited by 200M$ [7] to make it available for the users.
2. Architecture of Exascale Computing

The development of high-end systems such as supercomputing or High Performance Computing (HPC) that involves from millions to several million processors is highly challenging. It could have up to million processor cores, billions of threads, memory on the order of multiple petabyte. Increasing the clock rate by adding more transistors was the first approach to improve the performance of a computer. In high-end systems, the parallelism is the dominant factor to improve the performance. The parallelism within a chip has been increasing exponentially in two dimensional design. To fulfill the needs for the future computers, the parallelism is going to take another direction with three dimensional design[4]. Exascale supercomputers will still run x86 code like the current processors to ensure the compatibility with the current applications [3], [5].

Paying attention to a specific component of the system such as improving the performance in terms of FLOPs is not enough. Looking at the big picture of the system from different angles and how those components are interacting with each others is a key point. The overall performance in terms of FLOPS, memory speed, power consumption, data movement are all dependent. In other word, the processing speed is limited by the memory speed. To deliver this supercomputer to the market, a complete revolution is needed. In the following section, we provide an overview of the challenges to build the main architectural elements of the exascale supercomputers and what have been done so far.

2.1 CPU

Back in 80s, supercomputers used to be designed with specialized, custom-built processors which are very expensive. In early 90s, the research community has shifted into evolving more commodity components in the supercomputers with a better cost-to-performance ratio. In 2008, IBM revealed the Roadrunner machine which was the first supercomputer with hybrid processing scheme [8]. That brings the trend back to the specialized components as co-processors. The idea of hybrid processing was first invented for Sony PlayStation gaming console in 2008. The design of hybrid processors involved processing elements and specialized co-processing elements. Involving the specialized co-processing elements has improved the performance significantly. Other hybrid processing systems were built with (GPUs) as co-processors. Recently, the design of supercomputers with hybrid processing system became dominant.

2.2 Memory

To meet the performance requirement of exascale computing, the memory bandwidth have to increase, which in turns increases the power consumption. The more capabilities we put in the system the more the stress on the memory bandwidth. In order to meet the bandwidth needs stacking up more chips is not efficient due to data movement and power consumption. Additionally, increasing pins count for the sake of increasing the memory bandwidth will increase the cost without any performance improvement. That is because of the gap between CPU performance and the memory performance. Adding more cores to the system does not improve the memory bandwidth because adding pins is expensive. Nowadays, it is a matter of what we do about the power and the cost.

For the last 30 years, the memory technologies have not been changed. It is getting faster but basically it has the same architecture. We have to rethink the memory technology to make it more efficient. There are many thoughts on how the memory subsystem would look like in exascale computing. More innovative packaging and IO solutions are needed such as 3D stacking. For example, stacking the memory on top on the CPU which can be done by using new materials that can absorb the heat generated by the processor [9]. That will reduce the IO power consumption significantly while increasing the bandwidth.

A new technology is non-volatile memory technology that are beyond nano technology [9], [10]. However, the problem with this new technology is the limited lifetime which contradicts the reliability of this kind of computing system [10], [6]. The non-volatile memory does not consume energy while reading. However, it takes more energy than DRAM to write a bit. Memory technology have not changed for a long period of time. The design of the memory is probably will change permanently [4].

3. Power Consumption and Cost

Minimizing the power consumption while maximizing the performance is a key issue. The power consumption for the current high-end systems does not exceed 10MW. The future high-end supercomputers are expected to consume upto 20MW which is double the current consumption rate [4]. That would raise the cost of these super machines which contradict the design goals and requirements. A first step to find a solution for this problem is to figure out the sources of power consumption. In the current systems, the processor can consume upto 30% and the data movement became increasingly a source of high power consumption. Specifically, we should identify which part in the processor is consuming more power. Series of measurements and simulations have been done by Brooks and his colleagues to test the performance of the current processors and examine alternative designs [8]. Finding a balance between the performance and the clock frequency rate is complicated.

The memory is another resource of power consumption. Moving the data either horizontally or vertically has its own cost that could be higher than the cost of a FLOP [11]. All these factors combined urges the need for designing a new generation of chips and rethink the algorithms to compute efficiently. High power consumption will generate more heat,
hence, cooling will be needed. Finding the roots of the problem is a key point to develop a supercomputer with more efficient power management and control. A significant amount of research is needed to overcome this problematic issue.

4. Data Movement in Exascale Computing

While the FLOP used to be the most expensive component in the performance of HPC [7], the cost of data movement in a copper wire is not trivial and as important. The data movement could be vertical or horizontal. Vertical Data movement cost is the cost of moving the data from the memory into the processor and moving it back to the memory. Horizontal data movement includes moving the data between the interconnected [4]. For the exascale computing, the cost of the data movement could be more than the cost of a FLOP. Data movement management became increasingly important due to the cost in terms of power consumption and latency. That means, in average, every time we move a bit from the internal caches into the core we consume 10P of energy which is considerably high. For the future exascale computing, the cost could be even higher and it would cost upto 20P of energy. The consequence is that the cost for data movement costs more than the cost of a FLOP [11].

The increased cost of data movement is ignored in the current data programming systems. The current programming systems such as OpenMP values the flops and ignores the cost of data movement assuming it is free which is not the case [2], [9]. Developing applications that are more data locality aware could decrease the data movement significantly. For example, the old model of OpenMP describes how to parallelize loop iterations evenly among processors while ignoring where is the data located. A new programming system and algorithmic models should be more data-centric. These systems should describe how data is laid out in memory and the loop statements should operate locally on data similar to MapReduce [11].

From hardware point of view, the cost to move a bit is proportional to distance. The emerging hardware constraints are increasingly mismatched with the current programming paradigm [11]. Hardware/software co-design must consider better decisions about the future programming environment together with performance. Intel has already established co-design centers world wide to understand what the system developers need so they can develop a hardware that can be efficiently utilized by the system developers [11].

It is not only about the flops we count, it is about seeking a balance in terms of data movement and FLOPs. A better data movement management system and innovative solutions to minimize data movement across the system is extremely vital. We need to come up with HW/SW co-design for a better data movement management.

5. Algorithms

The advances in the architecture of super computers raise the need for changing the programming systems approach. The applications in supercomputing systems need to run more efficiently in terms of scalability, reliability and data movement. Dealing with parallelism and data locality for the exascale supercomputing is challenging. It is impossible for the programmer to manage a high-end system with several million processors in terms of load balancing, failures, and data locality. These machines should have the capability to manage the applications at the runtime [7], [4], [5].

Improving the algorithms could result in more efficient power management system. Self-aware systems is one approach toward solving this problem [5], [2]. In other words, the code should be able to exploit the efficiency of the machine. For example, shutting down the network while the machine is doing calculations to save the energy consumed by the network and turn the network on when data transfer is needed. That is essential for energy management. Traditionally, in computing environment that has always been handled by the operating system. Nowadays, having hardware that is able to implement event-based power management system is a necessity.

The programming models are increasingly mismatched with the reality of the underlying hardware architecture. Rewriting the current algorithms and applications can contribute to define the architecture of the future exascale computers through the co-design with hardware architects. Data structures and algorithms should be optimized to minimize data movement in the system [7], [4], [11]. The programmer can specify what the machines need to do to run the applications efficiently and the architect will build the hardware based on that [4], [11].

The software developers understand the requirements for exascale applications so they can provide feedback to the hardware architects and provide guidelines to build exascale HW and SW prototypes [4]. Because the programming model should be a reflection of the underlying machine architecture, the co-design is essential to seek balance between cost-to-performance ratio. Even if there is a new technology in the hardware, we have to change the software to use this technology efficiently. That can be achieved by understanding the performance consequences for the software running on that machine.

6. Reliability

Reliability and fault tolerance are essential for high-end computing. The reliability in a system with several million processors is a difficult challenge because of their scale and complexity [10]. When operating the transistors at a high level, the chances of failure increases which make the
reliability an essential factor in developing such a high-end system [3]. Failure in HPC is very costly and might result in the loss of a very large amount of computed work. Additionally, these systems involved long-running jobs that should finish in a timely manner. The applications and the system in general should be more resilient and able to recover from errors.

Traditionally, defining a checkpoint-restart was a solution. However, at this scale, checkpoints are problematic due to the overhead involved in terms of cost and energy [7], [9], [10]. It is used to be the hardware job, for the future HPC it became a hardware and software challenge [3]. Beside, Memory should have more significant and sufficient error correction codes should be implemented.

To develop a resilient system for future HPC with exaFLOP, the more transistors we have the more reliable the system should be. Many approaches have been proposed such as using state machine replication [7] which was adopted from the high-availability systems. Other studies [9] suggested improving the data locality in order to improve system reliability. However, adopting well-known fault tolerant techniques are not the solution. Data integrity and consistency in the event of failure is an important point [6]. Therefore, the entire software stack running on the system should be fault tolerant and aware. That’s to ensure the integrity of the whole system and avoiding building a reliable system over unreliable one [10].

7. Conclusion

The current emphasis is on preserving the FLOPs. The real cost now are not FLOPs, it is data movement. That requires shifting to data locality centric programming paradigm and developing hardware features to support it. The programmers should focus on developing applications with significant parallelism capabilities. With the new technologies involved we can get the capacity and density we need. Better simulation tools are important to predict the performance of theses machines without the cost of building real machines.

References