A Note on Scalability

Scalability as it is used in the computer industry is such an ill-defined term that Mark Hill, a well-known computer architect, once challenged researchers to either define it or stop using the term. However, parallel computing is one area where the term does have a rigorous meaning. In parallel computing scalability refers to the ability to improve performance by using more than one computing device. Performance does not only refer to solving a given problem faster; it can be more general than that. Performance improvements can also be the ability to solve a larger problem in the same amount of time where the problem is larger, either in terms of the amount of data or in terms of doing more computation. For example, weather prediction does not need to be faster, but can benefit from either shrinking the grid size or doing more accurate modeling within each grid sector, thus leading to better predictions of the weather.

A parallel program can have many data streams and execution streams communicating among themselves. With so much happening in parallel and the different streams depending on each other for progress it is challenging to ensure that all parts of a system scale in a balanced way to avoid one part becoming the bottleneck and limiting the ability to improve performance. Amdahl in 1967 captured the essence of this in a simple analysis now known as "Amdahl's Law", which loosely stated, is that in any system one cannot go faster than the slowest part. This slowest part or bottleneck is what eventually limits scalability.

Parallel computing tries to create software, which to the extent possible, balances out the resources to reduce the effect of the bottleneck. In some problems, often for big data problems with large amounts of independent computation, the bottleneck is computational and adding more processors makes it easy to scale up. But most problems are not so independent and for those problems the overheads are important and the bottleneck and how it may shift between memory, I/O -- disk and network, and computation for different problems and problem sizes is difficult. One needs to consider the entire system since any one component may in the end be the limiting factor. Parallelism needs to address the entire system to ensure it is "turtles all the way down". The right hardware needs to meet the right software to achieve a balance.

Parallel computing tries to identify the bottleneck, eliminate inefficiencies, and reduce the communication overheads, all of which limit performance. Sometimes the limits are computational. For example a problem of size N whose computation is proportional to N3 is going to hit a practical limit. At N equal to one million this results in more than 1018 instructions which at a one nanosecond cycle time requires more than 30 years of computation time; completely unfeasible. The objective of parallel computing is to add more processors, reduce the overheads, so that solutions can be computed within the required time.

There are some aspects of scalability that are easier than others. Adding capacity, more DRAM, more disk space, more CPUs, and more network bandwidth are the types of issues that are easier to address because capacity continues to dramatically increase. However, disk access times, DRAM access times, CPU clock speeds, and communication latency are more difficult challenges that can limit scalability for which there are no easy technological solutions. Fundamental limits associated with communication latencies can only be mitigated not eliminated.

Parallel computing is also closely connected to the design of scalable algorithms. For example, in machine learning there is considerable interest in designing algorithms for large data sizes to achieve the same results with less work. These algorithms in combination with parallel computing pushes the practical limit of solving larger and larger instances of these problems.

Scalable Analytics' goal is to provide parallel software for computing analytics with a state-of-the-art cluster computing platform to achieve the right software and hardware balance that can scale to the practical limits of the system.

Alan Wagner
Chief Science Officer
January 2012