

MPT's Howard Cascade[®] : Expansion Rates and Effective Bandwidth



Massively Parallel Technologies
1221 Pearl Street
Boulder, CO 80302, USA
303-926-8555
www.massivelyparallel.com

Overview

We present *Howard Cascade*[®] expansion rates and the effective bandwidths achievable on applications that require collective communication operations.

Background

Massively Parallel Technologies, Inc. (MPT) develops parallel processing frameworks and communication models. All MPT communication models are *zero-entropy*. That is, all bidirectional communication channels are used the maximum amount while insuring that there is no over-subscription of data and no data collisions, without the use of synchronization messages or signals. In addition to zero-entropy communication models, MPT also insures maximum time overlap between communication and computation, known as *beta phase communication*. Thus the foundation for MPT's technologies is the use of zero-entropy, beta phase communication. These concepts are encapsulated in the *Howard Cascade* (see US Patents 6,857,004 and 7,418,470).

MPT applies its patented, proven technologies to deliver and sustain near-linear scaling to thousands of processors. Key concepts of the technology include:

- Clusters can produce high sustained computational capability even if they have relatively slow processors and high communication latencies.
- Non-blocking cascades produce the best computational capability.
- Per-processor efficiencies of 90% and greater can be achieved on clusters consisting of thousands of nodes when overhead growth rates (and thus the effective bandwidths) are properly managed.

The two fundamental principles of the Howard model are:

1. All resources should be fully consumed when completing a task, and all resources should be available when needed.
2. For many collective operations, having *more* communication channels is better than having *faster* channels.

There are three major components to MPT's technologies:

1. *Parallel operating environment*: Enables serial algorithms to operate in a parallel environment without rewrite.
2. *Software framework*: Minimizes the inherent inefficiencies of existing parallel processing architectures.
3. *I/O and communication models*: Maximizes ability to overlap I/O and increase effective bandwidth by several orders of magnitude in a fraction of the time steps of conventional models, with near-zero overhead growth out to thousands of servers.

MPT's models require no global synchronization and maximize channel use, supporting fully overlapped communication with computation for maximum resource utilization and overall system performance. As a result, MPT's effective bandwidth rates greatly exceed conventional parallel processing frameworks such as the Message Passing Interface (MPI); however, an existing MPI-based program can still take advantage of MPT technology via an MPI-compatible programmer interface.

MPT's Technology Concepts

The key to high performance lies with distributing the problem over multiple processors without incurring inefficiencies. Inefficiencies can arise from communication cost, serial steps in the algorithm, and load imbalance between times when processors synchronize. Inefficiencies grow with the number of processors but decrease with the problem size. Figure 1 shows a log-log diagram of ensemble size versus problem size, and some of the effects on efficiency:

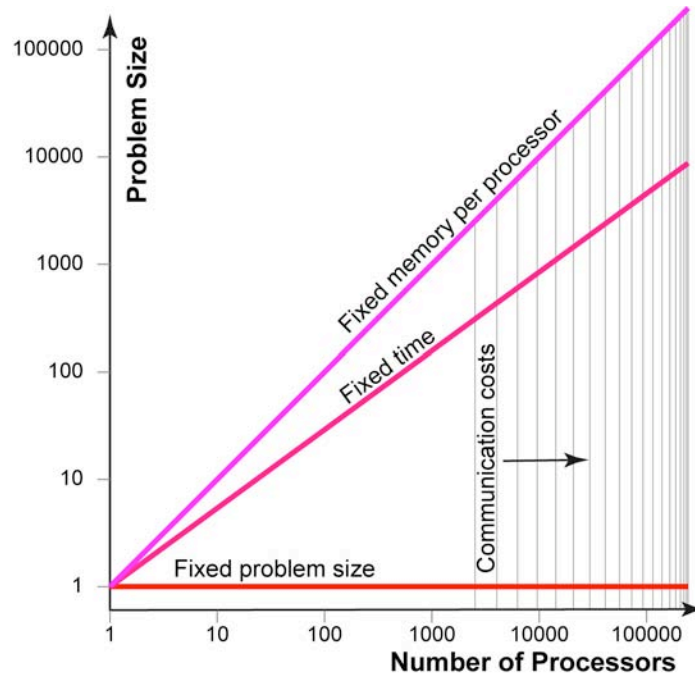


Figure 1. Problem size vs. ensemble size

Our scientific instincts are to study a problem one variable at a time, controlling all other variables. Historically, academics have fixed the size of the problem being studied, shown in the red “Fixed problem size” line in Figure 1; this model is sometimes referred to as *strong scaling*. Amdahl's law (1967) describes how the *serial part of the algorithm* impacts maximum speedup possible, regardless of how many processors are applied to the problem:

$$Speedup_{max} = \lim_{p \rightarrow \infty} \frac{1}{(1-f) + \frac{f}{p}} = \frac{1}{1-f}, \quad (\text{Eq. 1})$$

where f = the fraction of parallel activity within the algorithm and p is the number of processors. The model does a poor job of predicting actual performance since it ignores communication costs, load imbalance, and the effect of memory hierarchy (which contributes superlinear speedup effects). It serves mainly as a reminder that if you sum a slow speed with a fast speed *for a fixed amount of work*, you have to add speeds *harmonically*; that is, you have to sum the reciprocals of the speeds, not the speeds themselves.

Figure 1 shows two other lines that control all but one variable: “Fixed memory per processor” and “Fixed time.” Fixed memory per processor is appropriate for situations where the execution time is always quick enough to satisfy the user, but the user wants more detail or scope in the problem and runs out of memory before running out of patience. Ni and Sun (1992) have provided careful analysis

of this workload model. Given the relatively low cost of memory capacity, this model is found less and less in real workloads. One sees this model in the HPL benchmark (“LINPACK Benchmark”) used to rank TOP500 supercomputer centers; the problem is allowed to expand to the maximum memory per server, which results in execution times that expand to impractical durations for the largest systems.

Fixed time performance modeling (recently given the term “weak scaling”) uses perhaps the most representative variable to fix; the human tendency is to scale a problem to the point where it does the best job possible within some time limit. This governs such diverse workloads as the U.S. Census (10 years) to the 24-hour weather forecast (3 hours) to the response time of a video game (about 1/50 second). For fixed-time execution, speeds do not add harmonically as they do for Amdahl’s fixed-size model; they add directly, without having to take reciprocals (Gustafson’s law, 1988):

$$Speedup_{max} = \lim_{p \rightarrow \infty} pf + (1-f) = \infty \tag{Eq. 2}$$

When the number of processors increases to many thousands, it becomes clear that neither Eq. 1 nor Eq. 2 have useful predictive value. The performance for massively parallel systems is primarily *communication bound*. This is where MPT technology contributes.

MPT’s communication models greatly increase the performance of communication hardware. Because Cascades can be created using different numbers of communication channels, there are an infinite number of possible Howard Cascades. They generalize the usual point-to-point communication model and encompass partitioning of bandwidth across multiple channels as well as optimizing of “fast forwarding” of partial messages to minimize exposed latency.

Illustrated in Figures 2 and 3 are basic Howard Cascade generation diagrams.

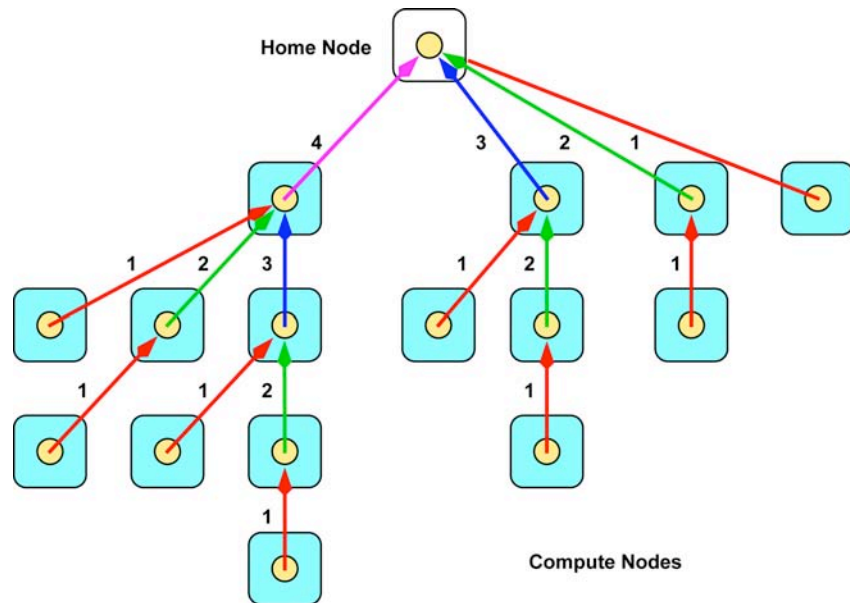


Figure 2: Howard Cascade, 1 channel per node

Space does not allow rigorous derivation of the algorithms for the production of these communication topologies; the figures are intended to give a flavor for the automatic methods that MPT uses to greatly reduce the communication cost shown in the right side of Figure 1. The case shown in Figure 2 is recognizable

as a standard binomial tree collapse commonly used in MPI libraries, but the Howard Cascade has a far more general form than the usual logarithmic-complexity collective operations.

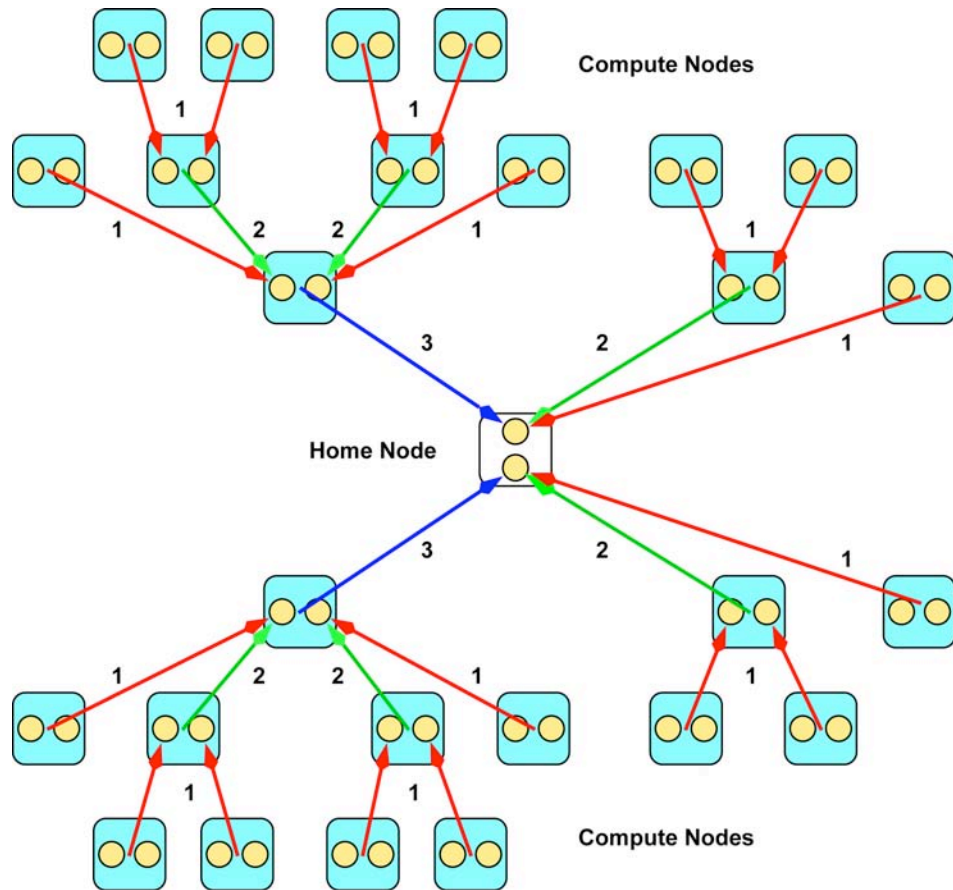


Figure 3: Howard Cascade, 2 channels per node

As the number of communication channels increases, higher efficiencies and effective bandwidths can be achieved automatically without a change in the underlying communication algorithms.

Expansion Rates and Effective Bandwidth

The number of nodes accessed as a function of the number of time steps is shown in Table 2, where a time step represents the time required to move the data set between two nodes. Table 2 shows that in just four time steps, a four-channel Howard Cascade can clear data through 624 nodes compared to only 8 nodes using a standard B-tree.

Time Units (φ)	Binary Tree	Howard Cascade, 1 Channel/Node	Howard Cascade, 2 Channels/Node	Howard Cascade, 4 Channels/Node
1	1	1	2	4
2	2	3	8	24
3	4	7	26	124
4	8	15	80	624
5	16	31	242	3,124
6	32	63	728	15,624
7	64	127	2,186	78,124
8	128	255	6,560	390,624

Table 2: Processor Count Expansion Rate Comparison

The resulting *effective* bandwidth delivered using MPT technologies and standard Cascades is as follows:

100 BaseT Ethernet, Effective Point-to-Point Bandwidth			
Time Unit (φ)	Howard Cascade, One Channel	Howard Cascade, Two Channels	Howard Cascade, Four Channels
1	100 Mb/s	200 Mb/s	400 Mb/s
2	300 Mb/s	800 Mb/s	2.4 Gb/s
3	700 Mb/s	2.6 Gb/s	12.4 Gb/s
4	1.5 Gb/s	8.0 Gb/s	62.4 Gb/s
5	3.1 Gb/s	24.2 Gb/s	312.4 Gb/s
6	6.3 Gb/s	72.8 Gb/s	1.6 Tb/s
7	12.7 Gb/s	218.6 Gb/s	7.8 Tb/s
8	25.5 Gb/s	656.0 Gb/s	39.1 Tb/s

Table 3: Effective Bandwidth

Thus, using standard networks it is possible to achieve effective bandwidths that far exceed what is possible with conventional MPI point-to-point use of the hardware alone. This advance is essential for computing to make effective use of many thousands of processors coupled to work on a single application.

Summary

Problems that depend on collective communication operations can use the Howard Cascade to eliminate agglomeration bottlenecks; effective bandwidths in the tens-of-terabits per second range are now possible. In total, use of the Howard Cascade delivers scaling efficiencies and high, sustained performance that save time and money while allowing the use of commercial off-the-shelf processing hardware and switching networks.

ABOUT MASSIVELY PARALLEL TECHNOLOGIES

Massively Parallel Technologies (MPT), Inc. develops parallel processing software based on rigorous algorithms that delivers excellent scaling and speedup on distributed memory systems for high-performance applications. Solutions built upon MPT's technologies deliver the fastest time at the lowest cost. Massively Parallel Technologies was incorporated in December 1999 and has offices in Boulder CO, Phoenix AZ, and the Silicon Valley, CA area.



Massively Parallel Technologies, Inc.

*1221 Pearl Street
Boulder, CO 80302
303.926.8555 (p)
303.926.0055 (f)*

www.massivelyparallel.com
info@massivelyparallel.com