

be effectively used if the communications problem were to be solved in a more reasonable fashion.

[0010] Broadcasting is an essential feature of parallel computer interconnects. It is used for synchronization, and is intrinsic to many types of calculations and applications, including memory system coherency control and virtual memory. Many applications running on today's supercomputers were written decades ago for relatively small parallel computers that had good bandwidth for broadcasting. These programs run poorly on today's massively parallel machines. The commonly used interconnects based on cross bars and fat trees as well as all existing parallel computers with  $n$  interconnecting nodes consume  $n$  channels of bandwidth during broadcasting, so the per port and bisection bandwidths do not change substantially when broadcasting.

[0011] Massively parallel high performance computers using fat tree and crossbar interconnect suffer from a mismatch with the software requirement for non-blocking broadcast of short messages. Two of the most common network functions, Allreduce and Sync simultaneously broadcast one-word messages. Such broadcast uses excessive bandwidth in fat-tree interconnects which results in poor system performance. Another function, termed all-to-all communications wherein each computing node in a supercomputer frequently needs to communicate to all other nodes during the course of a computation is an essential functional capability of any modern interconnect scheme. Additionally, these all-to-all messages are typically short, being a few bytes in length. Frequently used algorithms requiring the all-to-all function include parallel versions of matrix transpose and inversion, Fourier transforms, and sorting. The most effective way to implement the all-to-all function is to base it on a true broadcast capability. Present systems can broadcast information, but only by simulating the broadcast function; thus their capability for implementing the all-to-all function is inefficient.

[0012] A poor solution to the interconnect problem leads one directly to the general assumption that the most powerful processors available should be crammed into each node to achieve good supercomputer performance, thus hiding the problems inherent in the interconnect by faster processors and higher channel bandwidth. A compromise is possible if some of these other issues are more effectively resolved. The compromise based on a more suitable interconnect would make use of processors not quite on the leading edge of integration and performance to create a supercomputer of lower cost and power consumption with just as great, or more, overall capability. Of course, nothing prevents one from using the ultra-performance processors as nodes in the proposed systems; both cost and capability would rise significantly.

[0013] Today's supercomputer architecture at most makes use of 8-way multithreading, meaning that there is hardware support for up to 8 independent program threads. Any multitasking to be found is handled by software. While theoretically alleviating the communications bottle-neck problem and helping to overcome data-dependency issues, the cure is literally worse than the disease since the nodes now spend more time managing the system's tasks in software than is gained by decomposing complex programs into tasks in the first place. What is needed is a scalable and cost effective approach to supercomputers that range in size

from a briefcase to a small office building, and in performance from a few teraflops to a few petaflops. (A petaflop is 1000 teraflops.)

[0014] Interconnect schemes today are invariably based on material busses and cross bars. As data rates increase and data processors become faster, electrical communication between data-processing nodes becomes more power intensive and expensive. As the number of processing nodes communicating within a system increases, electrical communication become slower due to increased distance and capacitance as well as more cumbersome due to the geometric increase in the number of wires, the volume of the crossbar, as well as its mass and power consumption. Electrical interconnects are reaching their limit of applicability. As speed requirements increase to match the capacity of ever faster processors for handling data, faster electrical interconnects should be based on controlled-impedance transmission lines whose terminations increase power consumption. Even the use of microstrip lines is only a partial solution as, in any fully-connected system, such lines should cross (in different board layers). Close proximity of communication channels produces crosstalk, which is perceived as noise on adjacent channels. Neither of these problems occur in a light-based interconnect.

[0015] Optical interconnects, long recognized to be the ideal solution, are still in the experimental stage with practical optical systems connecting only a handful of processors. The main problem with today's optical solutions is conceptual: they are trying to solve a more complicated problem than necessary. This restrictive view has its origins in a limited version of a task or thread: if CPU overhead is required to switch from a computational task to a communications task every time a message arrives, any conceivable computation spread across a multiprocessor system will soon be spending most all of its time on switching overhead. The way around this untenable situation is to create literal, point-to-point connections as is done for the Hypercube<sup>TM</sup> and Manhattan architectures such as the Transputer<sup>TM</sup>. Thus, the source and destination of every message is determined by hard-wired connections. This idea is carried over into optical schemes where there is an emitter dedicated to every receiver and a single receiver for every emitter. For an optical system serving hundreds of thousands of nodes, the mechanical alignment is an insurmountable nightmare.

[0016] Over the years, a number of universities and private and government laboratories have investigated free space optical interconnect (FSOI) methods for multiprocessor computing, communications switching, database searching, and other specific applications. The bulk of the research and implementation of FSOI has been in finding ways to achieve point-to-point communications with narrow beams of light from multiple arrays of emitters, typically narrow-beam lasers, and multiple arrays of photoreceivers. The development of vertical-cavity, surface-emitting lasers (VCSELs) and integrated arrays of VCSELs has been the main impetus behind research in narrow-beam FSOI area. The main problems with FSOI to overcome are alignment, where each laser must hit a specific receiver, and mechanical robustness. U.S. Pat. No. 6,509,992 specifically addresses the problem of misalignment and robustness by disclosing a system of redundant optical paths. When misalignment is detected by a channel-monitoring device, an alternate path is chosen.