STAP Benchmark Evaluation of the T3D, SP2, and Paragon[1]

Choming Wang*, Cho-Li Wang, and Kai Hwang****

* University of Southern California, Los Angeles, California

** The University of Hong Kong, Pokfulam, Hong Kong

Abstract

The STAP (*Space-Time Adaptive Processing*) benchmark suite consists of programs for signal processing. It requires to perform $\boldsymbol{O}(10^{10})$ floating point operations over $\boldsymbol{O}(10^2\sim10^3)$ MB of data. In this paper, we report the STAP performance results on three MPP (*massively parallel processors*) systems; SP2, Paragon, and T3D. We observed that no machine exceeds 10 Gflop/s sustained speed in executing any of the STAP programs up to 128 nodes. We analyze the strength and weakness of the three MPPs according to their aggregate bandwidth, communication to computation time ratio (CCR), and excessive overheads in disk/memory access. The aggregate communication bandwidth of the three MPPs at machine size 128 for SP2, Paragon, and T3D are 578, 1084, and 2870 MB/s, respectively. Paragon and T3D have relatively effective communication subsystem and both show a rather stable CCR over a wide range of machine sizes for fixed workload problems. The disk access delay is the major bottleneck in performance for machines with limited local memory. It has been observed that the disk access delay can be as high as 73% of the overall execution time. The memory copy overheads can be as high as 50%, 40%, and 90% of the total communication time on the SP2, Paragon, and T3D. We conclude that the computation power is the most important factor in the STAP application. The 3D-torus network of T3D shows the best communication performance. The Paragon 2D-mesh network suffers from the four-link-per-link topology. The SP2 architecture is scalable, but a much higher bandwidth network is required to further improve the performance.

Keywords: Message passing, massively parallel processors, STAP benchmark, performance measurements

1. Introduction

The Cray T3D, IBM SP2, and Intel Paragon are three MPPs currently in use at many supercomputing sites. In this paper, we assess the performance of these machines by the STAP (*Space-Time Adaptive Processing*) benchmark for radar signal processing. In a STAP application, the radar signal changes with time, corresponding to the continuous generation of radar beams and the successive processing of a sequence of data by the backend computer in real time. The backend computer processes the radar data and generates a target list among many targets detected.

The STAP benchmark suite was originally developed by MIT Lincoln Laboratory in sequential C code. It consists of five programs: (1) Adaptive Processing Testbed (APT), (2) High-Order Post-Doppler (HO-PD), (3) Beam Space PRI-Staggered Post Doppler (BM-Stag), (4) Element Space PRI-Staggered Post Doppler (EL-Stag), and (5) General (GEN). The GEN program consists of four component algorithms to perform sorting (SORT), Fast Fourier Transform (FFT), vector multiplication (VEC), and linear algebra (LIN). The STAP benchmark suite has become popular for evaluating radar signal processing architecture among the signal processing community [1, 2].

In general, the STAP benchmark suite is evaluated from the viewpoint of exploiting massive parallelism. The benchmark requires to perform $\boldsymbol{O}(10^{10})$ floating point operations over $\boldsymbol{O}(10^2\sim10^3)$ MB of data in a fraction of a second. This translates to a speed demand of $\boldsymbol{O}(10)$ Gflop/s in real-time applications. In our benchmark experiments, we choose a nominal input data set to simulate the real-time adaptive radar signals. Our work is one of the attempts to evaluate the effectiveness of commercial MPPs for large-scale, real-time signal processing [6, 8, 12].

To compare the overall performance of the three MPPs, we report the *sustained speed*, which is the ratio of total workload divided by total execution time, and the *utilization rate*, the percentage of the sustained speed to the total peak speed of all processors. To examine the communication capacity of the three MPPs, the important metric is the *aggregate bandwidth*, which is defined as the ratio of the total communication volume to the total communication time. This bandwidth shows the achieved application-level communication capability of the MPP. We define the CCR as the ratio of communication time to computation time of the fixed-sized problem over various machine size. While running the parallel code on various MPPs, lower CCR value indicates that the communication mechanism of the MPP is relatively more effective to its computing mechanism. A stable CCR over large range of machine size for an MPP indicates the system is more scalable.

We analyze the performance results and identify the strength and weakness of all three MPP systems according to their node architecture, interconnection network, and excessive disk/memory events. The rest of the paper is organized as follows. In Section 2, we briefly introduce the STAP Benchmark characteristics and outline the STAP parallelization strategy. The machine configurations used in our experiments are

1. This research was supported by MIT Lincoln Laboratory, University of Southern California, and in part by the University of Hong Kong. The authors can be reached by Email addresses: choming@diana.usc.edu, clwang@cs.hku.hk, and kaihwang@cs.hku.hk.

described in Section 3. The STAP experiments and performance results are presented and interpreted in Section 4. Architectural implications are described in Section 5. Finally, we conclude with lessons learned and provide suggestions to build future systems to meet the increasing demand of scalability and programmability.

2. STAP Benchmark Characteristics

Adaptive nulling radars can be built with several variations of the STAP architecture. The STAP benchmark suite is evaluated from the viewpoint of exploiting massive parallelism. Three integrated STAP architectures are possible: (1) Factored STAP and then doppler processing, (2) STAP integrated over a *coherent processing interval* (CPI), and (3) Factored doppler processing and then STAP.

STAP radars improve upon the conventional radars in adaptation in both space and time to reject clutter/jamming noises simultaneously. The main advantage lies in integrated space-time digital beamforming. A cost-effective choice among the three is driven by the number of adaptive weights used and by the frequency these weights must be updated; both are rather application-specific. Let M be the number of antenna elements; K be the number of PRI (*Pulse Repetition Interval*) for AMTI (*Airborne Moving Target Indicator*) process; and N be the pulses per CPI. Table 1 summarizes the STAP processing complexities of three adaptive radar architectures [5]. Note that the adaptive beamforming is essentially driven by the number of weights (W) required by each beamformer. In general, $\boldsymbol{O}(W^3)$ operations are performed in the adaptive weight generation process. From Table 1, the full STAP over a CPI required the most processing power; even doppler filtering is totally eliminated.

Table 1: Three adaptive radar architectures.

Architecture	Adaptive Beamforming		Doppler Filtering	
	Number of Beam-formers	Number of Wts (W) per Beam-former	Number of Filter Banks	Number of Filters per Bank
Factored STAP Then Doppler Filtering	1	$M \times K$	1	N
Full STAP over a CPI	N	$M \times N$	-	-
Factored Doppler Filtering then STAP	N	$M \times K$	$M \times K$	N

The *total workload* is defined as the total flop counts in the sequential program. The actual flop counts in the parallel program may vary due to the communication and parallelization overhead. However, the workload from the sequential program is used in our experiment to calculate all the performance metrics. The workload of the STAP program depends on the size of its input data cubes, n. Hence the size of the problem can be easily defined as the size of the input data and is in the order of $\boldsymbol{O}(n)$. The *total message length* is the total number of bytes for all messages communicated in a parallel program. For fixed size problem, the total message length is also fixed, as in the STAP benchmark. However, this number varies if the problem size changes. The values of total workload and total message length for STAP benchmark used in our experiment are listed in Table 2.

Table 2: Total workload and total message length.

Benchmark Programs	APT	HO-PD	BM-Stag	EL-Stag	SORT	FFT	VEC	LIN
Mflop counts	936	10991	1927	1202	62	1790	369	1637
Message (MB)	18.4	61.4	81.9	98.3	-	98.3	98.3	-

Parallelization of STAP Programs

The STAP programs were parallelized using the *Phase Parallel* model [12] in SPMD mode. In the following text, we use the words node, processor, or PE interchangeably to mean a single processor, unless otherwise specified.

Fig. 1 gives a conceptual sketch of the dataflow and control flow in the parallelization of a typical STAP benchmark program. The machine size p corresponds to the user *partition size* preassigned to a given application. In the STAP Benchmark program, each of the p partitions is matched to a single processor or PE. The STAP workload is decided by the amount of reflected radar signals from the target. The input radar signals are structured as a 3-dimensional *data cube.*

Each data element in the signal cube is a complex number (two floating point numbers). Depending on the radar configuration, the three parameters PRI, RNG (*range*), and EL (*element*) of each STAP benchmark programs are listed in Table 3. The input data cube is first partitioned along the RNG dimension into p equal slices. Each processing node takes one slice as its input data. Each node program consists of three major steps: *doppler processing* (DP), *beamforming* (BF), and *target detection* (TD). In the DP step, each node executes FFT routine. All p nodes work on different data slices simultaneously.

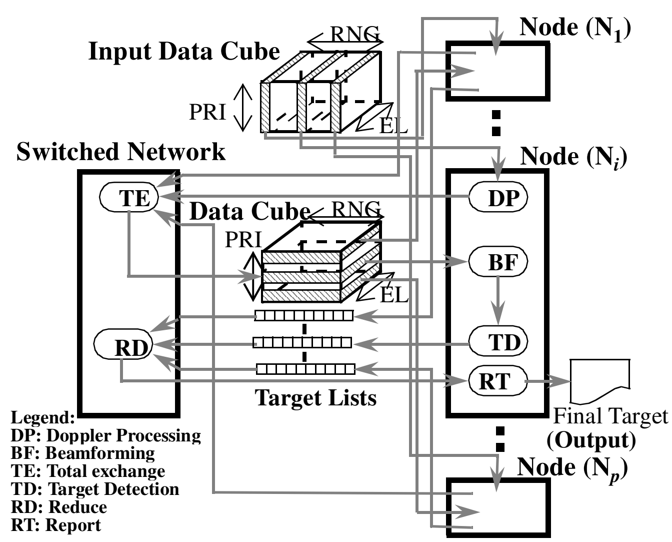

Figure 1. Data flow and control flow in each node program.

Table 3: The data cube dimensions and sizes.

Programs	APT	HO-PD	BM-Stag	EL-Stag	GEN
Dimension 1	256	128	64	16	64
Dimension 2	280	1250	1000	16000	128
Dimension 3	32	48	80	16	1500
Size of cube(MB)	18.35	61.44	40.96	32.77	98.30
Number of cubes	1	1	2	3	1

Following the DP step, every node must exchange their results with other nodes via collective communication, *total exchange*, through the switched network. The data distribution after *total exchange* is represented by the middle data cube in Fig. 1, and the data cube is partitioned along the *PRI* dimension into *p* equal slices. Next in phase BF, each node needs to perform the beamforming operations on the new slice of data. In the final TD phase, each node generates a partial *target list*, based on its own data slice. The last step in the node program is to merge the *p* target lists into a single target list. This merge operation demands the *reduce* collective communication. After the *reduce* operation, only one node is responsible for reporting the final target list as the output.new data slice.

3. Testing Environments

All three MPP systems have distributed-memory architecture using microprocessor-based nodes interconnected by switched network. Major architectural characteristics and testing environments are summarized in Table 4. We reserve the parameter *p* for the *machine size* or the number of *processing elements* of an MPP. Each processing node may consist of one or more processors or *processing elements* (PEs), depending on the machine architecture. For simplicity, we will use the terms, *node*, *processor*, and *PE*, interchangeably.

Table 4: Architectural and testing environments.

Machine Model	IBM SP2	Cray T3D	Intel Paragon
CPU	POWER2, 67 MHz, 266 Mflop/s	Alpha 21064, 150 MHz, 150 Mflop/s	Intel i860, 50 MHz, 100 Mflop/s
Node Architecture	1 Processor per Node, 64 MB to 2 GB Memory, 1 to 4.5 GB local disk or 50 GB Shared Disks	Dual Processors per Node, 64 MB Local Memory, 50 GB Shared Disk in Gateway Nodes	1 or 2 Processors, 16 to 128 MB Memory, 48 GB Shared Disk in I/O Nodes
Interconnect	Omega Multistage Switches	3-D Switched Torus	2-D Switched Mesh
Memory Model	NORMA	DSM/NUMA	NORMA
Node Operating System	IBM AIX with Parallel Programming Environment	UNICOS-MAX, a Mach-based Microkernel	Paragon OSF/1, a Mach-based Microkernel
MPI[9] Implementation	MPICH	CRI/EPCC MPI, MPICH	MPICH

4. Benchmark Performance Results

This section reports the measured performance of the STAP benchmark programs on three MPPs. All the data points for T3D up to 64 nodes configuration are results from running on machine installed CRI/EPCC MPI [4]. The data points on 128 nodes configuration were obtained from T3D with MPICH.

4.1 Sustained Speed Performance

The sustained speed is the ratio of the total workload divided by the total execution time. It varies with machine platforms and node numbers allocated. Among all three machines, we observed that no machine has exceeded 10 Gflop/s speed in executing any of the STAP programs up to 128 nodes. In general, the SP2 achieved the highest sustained speed on all machine sizes, followed by T3D, and Paragon.

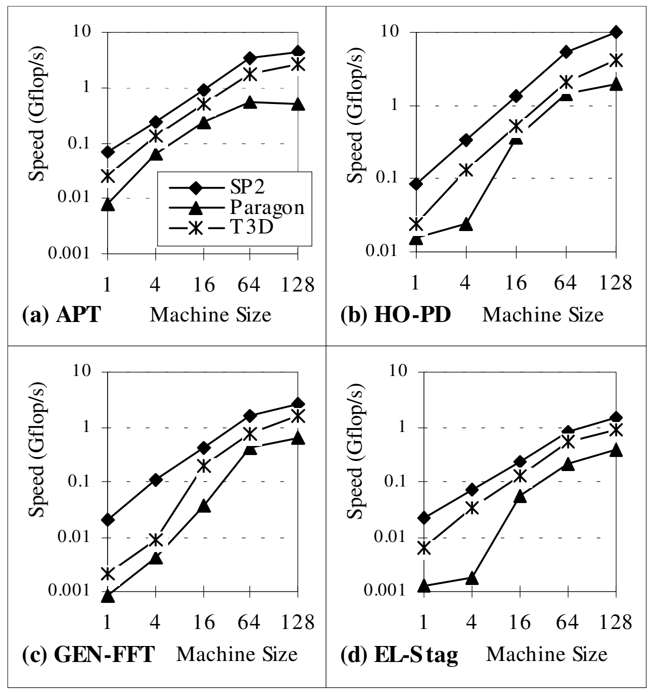

Figure 2. Sustained speed on various machine sizes.

In Fig. 2(a), the speed results of the APT program are shown. The APT program scored 66 Mflop/s on one SP2 node and 4.5 Gflop/s on 128 SP2 nodes. The same APT achieved 8 Mflop/s and 490 Mflop/s on 1 and 128 Paragon nodes, respectively. The T3D speed falls in the middle, ranging from 26 Mflop/s on 1 PE to 2.6 Gflop/s on 128 PEs. Notice that the communication time on 128 PEs is 0.44 sec using MPICH, which is higher than the projected 0.12 sec using CRI/EPCC MPI.

In Fig. 2(b), the speed results of the HO-PD program are shown. The HO-PD program is a computational intensive program with a workload of 10.99 Gflop. Its communication requirement is 61.44 MB. Hence its communication cost is relatively small, with a ratio of 0.0046 bytes communicated per flop. With 128 nodes, the SP2 achieved the highest speed of 9.8 Gflop/s. A 128-node Paragon achieved 2.0 Gflop/s speed in executing the HO-PD program. Note there is a dip in the Paragon with 4 nodes. This phenomenon is due to the *disk swap delay* (DSD) effect to be described in Section 5.3.

The speed results of the GEN-FFT program is shown in Fig. 2(c). The SP2 reached 2.5 Gflop/s and the Paragon reached 640 Mflop/s for a machine size of 128 nodes. The FFT program processes three large size data cubes with total size of 295 MB. Both the Paragon and T3D experienced a speed dip on a small configuration of 16 or less nodes due to small local memory and the DSD effect. The speed results of the EL-Stag program are shown in Fig. 2(d). This program needs to handle total messages of 98.3 MB all in one phase. Thus the communication time in the EL-Stag program occupies a large percentage in the total execution time. This degrades the speed of the EL-Stag program by at least an order of magnitude, compared with that of the HO-PD program.

4.2 Aggregate Bandwidth

The aggregate bandwidth is defined as the ratio of the total communication volume to the total communication time. The aggregate bandwidth of the T3D for all STAP programs are shown in Fig. 3(a). The aggregate bandwidth for FFT program are plotted in Fig. 3(b) for the three MPPs. The peak bandwidth at machine size 128 of SP2, Paragon, and T3D are 578, 1084, and 2870 MB/s, respectively. For all 3 MPPs, the aggregate bandwidth increases with machine size. The Paragon suffers from the DSD effect such the aggregate bandwidth on small machine size is degraded significantly due to limited local memory to contain relatively large message. The T3D has better aggregate bandwidth than the other two MPPs.

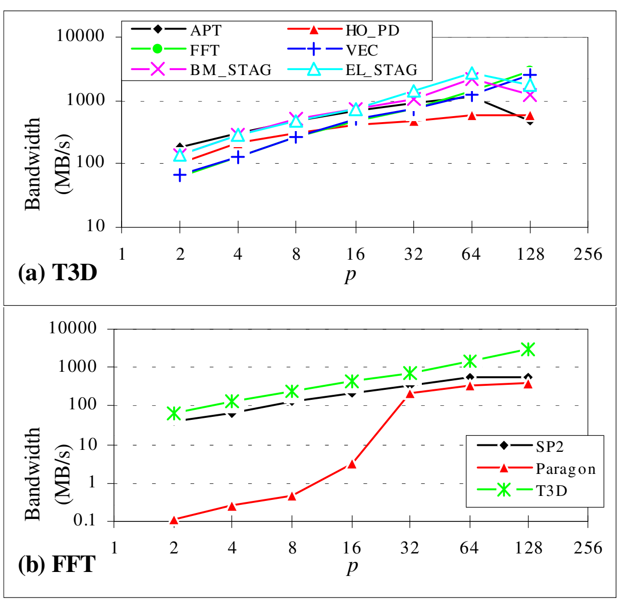

Figure 3. The aggregate bandwidth obtained for (a) T3D for six programs, (b) FFT program on three MPPs.

4.3 Breakdown of Parallel Execution Time

The timing breakdown of all STAP programs is summarized in Table 5 for two machine sizes, one is a small machine size of 8 nodes and another corresponds to a medium size of 64 nodes. The computation time is divided into two components: *doppler processing* (DP) and *beamforming* (BF). All the communication overhead (Com) is lumped in a single column in Table 5. The Disk time accounts for all the time to access the disks during the DP, BF, and Com operations. The effects of these excessive disk events will be discussed in Section 5.3.

Table 5: Breakdown of parallel execution time in seconds.

MPP	SP2				T3D				Paragon			
Code	DP	BF	Com	Disk	DP	BF	Com	Disk	DP	BF	Com	Disk
A Small Machine of 8 Processing Nodes												
APT	0.49	1.29	0.13	NA	0.62	2.80	0.07	NA	1.87	5.61	0.25	NA
HOPD	1.38	14.8	0.41	NA	1.75	39.2	0.26	NA	5.03	55.1	0.86	118
BM	2.85	3.73	0.95	NA	4.46	7.78	0.93	NA	NA	NA	NA	NA
EL	3.68	2.62	2.58	NA	5.35	10.6	2.03	NA	16.9	22.8	4.29	NA
SORT	0.18	NA	0.00	NA	0.37	NA	0.00	NA	0.73	NA	0.00	NA
FFT	7.44	NA	0.78	NA	18.3	NA	0.39	NA	34.0	NA	1.03	209
VEC	1.52	NA	0.55	NA	3.07	NA	0.38	NA	7.20	NA	1.47	NA
LIN	2.43	NA	0.00	NA	8.41	NA	0.07	NA	12.3	NA	0.02	NA
A Medium-Size Machine of 64 Processing Nodes												
APT	0.04	0.19	0.06	NA	0.07	0.44	0.01	NA	0.26	0.88	0.57	NA
HOPD	0.10	1.86	0.15	NA	0.20	4.90	0.10	NA	0.59	6.91	0.31	NA
BM	0.28	0.50	0.20	NA	0.60	0.96	0.14	NA	NA	NA	NA	NA
EL	0.47	0.44	0.63	NA	0.70	1.37	0.27	NA	2.01	2.98	0.62	NA
SORT	0.03	NA	0.00	NA	0.05	NA	0.00	NA	0.09	NA	0.00	NA
FFT	0.97	NA	0.18	NA	2.25	NA	0.07	NA	4.14	NA	0.30	NA
VEC	0.19	NA	0.21	NA	0.38	NA	0.08	NA	0.79	NA	0.31	NA
LIN	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
NA: Not-Applicable												

For communication intensive program EL-Stag, the total communication time is in the range of 10 to 12% of the total execution time for T3D and Paragon in both small and medium machine sizes. For SP2, the communication time is 29% of the total execution time for small machine size, and 41% for medium machine size. Although the message processors within the nodes of SP2 and Paragon are both i860, the SP2 has a relatively slow multistage network (40 MB/s per link in SP2 vs. 175 MB/s per link in Paragon).

The disk time for executing the HO-PD and FFT on the small Paragon configuration are higher than the sum of other three time components. This is due to the Paragon compute nodes are not attached with local disks. The paging faults cause the disk access which must traverse through the 2-D mesh network to the pager node using message passing. The long disk access latency slows down the Paragon performance.

The disk access is the major bottleneck in performance. For 4 node Paragon running HO-PD program, the disk access time is 331 sec, about 73% of the overall execution time. This is attributed to two reasons: 1) Small local memory can not contain large data set, hence paging faults occur more often, and 2) No local disk attached to the node can further increase the disk access latency, since all disk access must transverse the interconnect network, as in the case of Paragon. Accessing remote disk through the interconnect network further increases the network traffic and this in turn decreases the communication performance. Apart from the disk access time, the beamforming consumes the most time in running all STAP programs on all MPPs. The computing power of the node processors determines the time to perform the beamforming operations.

4.4 Effects of Communication-to-Computation Ratio

The *communication-to-computation ratio* (CCR) is defined by dividing the total communication time by the total computation time measured in the STAP Benchmark experiments. The CCR equals zero for an one-node application partition, where there is no need of communication.

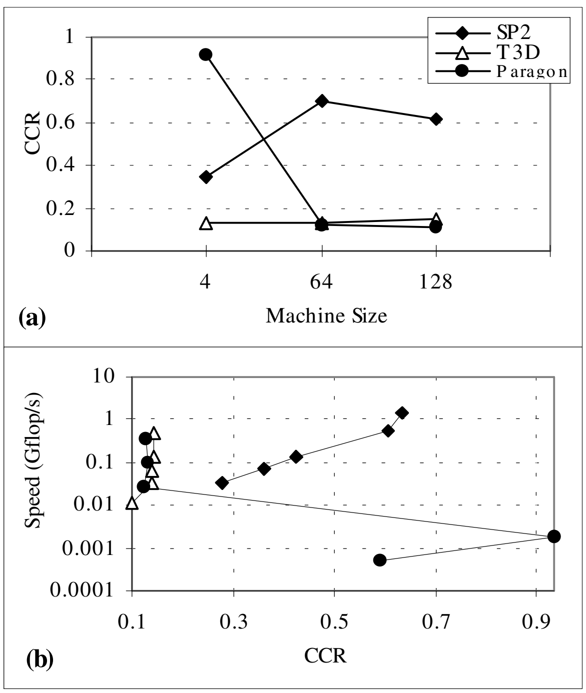

Figure 4. (a) Variation of CCR with the machine size, (b) Speed variation with the CCR value.

This ratio is affected by the software overhead in initiating the message passing functions, the processing speed of the message processor to assemble and disassemble the messages, and the routing algorithm used in the network, *etc*. In Fig. 4(a), the variation of CCR with the machine size are plotted for the EL-Stag program. The T3D has the most stable CCR for various machine sizes among the three MPPs. The absolute communication time for T3D is also the lowest among the three MPPs. We can say that the T3D has equipped with the most efficient communication mechanisms among the three machines. However, the T3D performs lower in overall *sustained speed* than SP2, because the POWER2 processor is much more powerful in floating-point speed, compared with the Alpha 21064 microprocessors.

When the total message length is large such that memory can not accommodate the whole message, the Paragon swaps the message to disk. This DSD degrades the communication performance. Furthermore, since there is no local disk on the Paragon node, the swapping activity must pass through the network to access the remote system disk. The network traffic is further increased due to swapping. This is reflected as the very high CCR as shown in Fig. 4(a) for Paragon on machine size 4. Without DSD effect, the Paragon shows similar CCR values as T3D, as shown in Fig. 4(a) on machine size 64 or greater. This indicates that the 2-D mesh network on Paragon is capable of catching up with the processor requests for data.

Irrespective of the DSD effect, the SP2 has higher CCR than either Paragon and T3D. Both SP2 and T3D have comparable communication time, whereas the computation time on SP2 is less than half of that on T3D. The high CCR on SP2 (Fig. 4(a)) is mainly attributed to the very fast processors in SP2 node. We can conclude that the performance bottleneck on SP2 is on its multistage network. Although the overall performance of SP2 is the best among the three MPPs, a better network for SP2 is more important than a faster processor.

In Fig. 4(b), the speed variation with different CCR values is shown for the EL-Stag program. For fixed size problem, the speed increases with increasing machine size. The speed begins to drop when the machine size is so large such that the communication shadows the benefit obtained from increasing processors. When the speed is plotted against CCR, we should see a curve starting from the lower left corner and increasing to the upper right direction. A peak speed is then reached. Further increase in CCR will results in speed reduction. The SP2 shows an increasing speed with increasing CCR. The performance peak of SP2 was not reached up to 128 nodes.

On the right corner of Fig. 4(b), the Paragon shows a lower speed for a very high CCR. This is attributed to the communication using disk as a swap space. For a fixed total message length in collective communication, the time usually reduces by half, when the machine size doubles. The 308 sec communication time apparently includes the DSD effect. Disregarding these two points, both Paragon and T3D show a rather stable CCR for fixed workload problems.

In general, if the CCR value is small, higher speedup can be achieved with some performance tuning. With large CCR, the parallel algorithm should be redesigned. To improve the algorithm design, we should allow more computation to be done on each processor and reduce the interprocessor communication. Duplicating the input data can cut down the number of interactions and thus reduce the overhead. Eliminating data dependencies among subtasks can reduce the synchronization overhead. Various message passing modes (such as *blocking*, *synchronous*, and *buffered* modes in MPI) can also avoid the communication overhead.

4.5 System Utilization

The ranges of the system utilization of all three MPPs are shown in Fig. 5(a) for all STAP programs. The SP2 achieved the highest overall utilization of around 31.8%, 30.6%, and 28.6% at machine sizes 4, 64, and 128, respectively. The Paragon has the lowest utilization range among three machines. It only achieved 16.6%, 22.0%, and 15.6% utilization at machine sizes 4, 64, and 128, respectively. The T3D has very stable utilization of 22% for all machine sizes. Generally speaking, the system utilization rate decreases with increasing machine size in most parallel applications.

In Fig. 5(b), we show the variation of utilization with machine size for the EL-Stag program. The Paragon observed the DSD effect on small machines. The overall utilization for all three machines is rather low in this program. This is attributed to the fact that EL-Stag has very large communication overhead. The computational granularity assigned to each node program should increase accordingly to lower the CCR value.

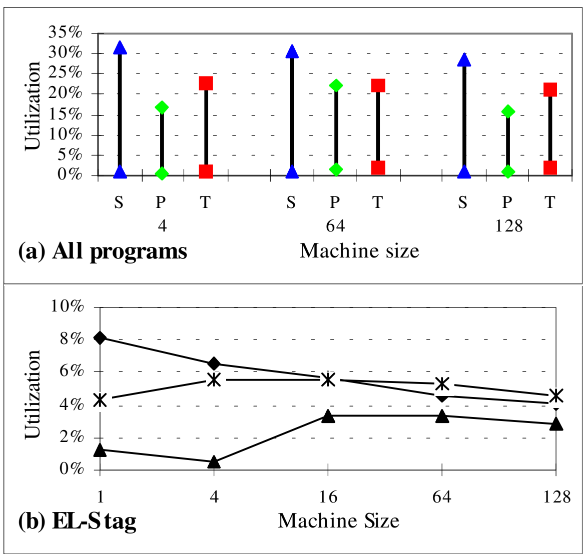

Figure 5. Utilization for (a) All STAP programs, (b) EL-Stag program. (S: SP2, P: Paragon, T: T3D)

5. Architectural Implication of Benchmark Results

The performance of MPPs can be attributed to many hardware mechanisms and software support systems. We identify below the architectural strength and weakness of each MPP with respect to STAP applications.

5.1 Node Architecture

The node architecture greatly affects the total system speed. Among the three MPPs, the SP2 performs the best in speed and utilization for almost all STAP benchmark programs. This is primarily attributed to the peak 266 Mflop/s speed and a good optimizing compiler designed by IBM POWER2 microprocessor group. Even the Alpha 21064 has higher clock rate, it is outperformed by the POWER2 for its lower instruction-level parallelism exploited. For 64 nodes, the relative speed ratio of SP2, T3D, and Paragon are 5: 2: 1, respectively. Another strength in SP2 lies in large node memory allowed, compared those allowed in each Alpha or i860 node. For example, many of the Paragon nodes have only 16 MB of local memory. Loading a large data set into this node requires virtual memory which causes many disk swaps, and it significantly degrades the system performance.

5.2 Switched Interconnection Networks

Communication is expensive on the MPPs. The communication performance is heavily tied to the switched network latency and bandwidth, as well as to the CCR, which depends on both network traffic and node granularity. Our measured latencies are rather close to the manufacturer supplied values. In case of collective communication, the aggregate bandwidth depends on network bisection bandwidth, the message length, and operation type, as seen in those for the *total exchange* operation.

The lower CCR is attributed to the lower software overhead in initiating a message passing function, the speed of a message coprocessor in assembling and disassembling the messages, the routing algorithm used in the interconnection network, *etc*. Large-scale signal processing applications, such as STAP, can only exploit coarse-grain parallelism for much higher communication delay than the computation delay in each node. The lower CCR often leads to a better speed performance as explained in Section 4.5.

The 3-D torus topology in T3D network ensures short connection paths as well as a high bisection bandwidth. Three latency hiding mechanisms in T3D, *prefetch queue* (PFQ), *read ahead, and Block Transfer Engine* (BLT), help improving the overall T3D performance. These features are designed to hide latencies in passing a few words among the nodes, as well as in transferring thousands of words effectively.

5.3 Effects of Excessive Memory/Disk Events

A typical STAP program requires to process more than 100 MB of radar signal data cube within each pulse repetition interval. It is impossible to load the entire data cube into the local memory of a single node. Among the three MPPs we have used, a single node of T3D has 64 MB of local memory, and the Paragon has only 16 MB per node. Excessive disk accesses are required to support large data set.

For example, the Paragon supports microkernel operating system on each compute node. The kernel and server will occupy 6.5 MB of memory. The NX message buffer occupies another 1 MB of memory. This leaves less than 8 MB for data memory, beside the space needed for the node programs. In this section, we address several issues related to unwanted memory overhead and disk-access penalties in using MPPs with limited local memory resources.

Disk Swap Delay

The DSD occurs often in MPP with limited local memory to solve a large-scale problem. We refer DSD to systems with virtual memory support. The problem comes from the fact that the local physical memory is not sufficient to hold the entirety of the node program and the associated data allocated to a given node, therefore paging faults occur frequently. This DSD may severely degrade the MPP performance. Fig. 6(a) illustrates the DSD effects in using 16 MB in each Paragon node. The DSD effect may degrade the performance of the Paragon up to 7.6 times in the STAP benchmark.

Large local memory is required in large-scale or real-time applications, such as the STAP radar signal processing, or when the exploitable parallelism is limited. Avoiding unnecessary DSD is of crucial importance in these applications. One may also consider to trade off DSD with internode communication to improve the performance.

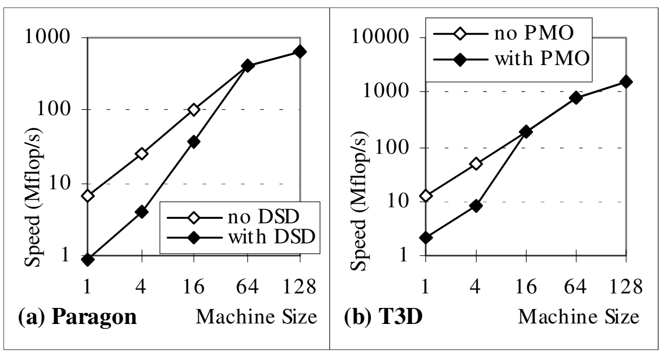

Figure 6. Speed variations on (a) Paragon, and (b) T3D.

Physical Memory Overlay in Program Fitting

The *physical memory overlay* (PMO) occurs when the node program and data are too large to fit the local physical memory and the virtual memory is not supported by the OS. In this case, the programmer may divide the program and data into overlay pieces, which will be moved in and out of local memory dynamically under program control during execution time. The unloaded portion is stored on the disk.

This problem happened in using the T3D machine. Because there is no virtual memory on the T3D machine, the physical memory is all that is available. The PMO effect may degrade the performance of the T3D up to 5.7 times in the STAP benchmark (Fig. 6(b)). The PMO problem may destroy the portability of the application codes. The solution is to increase the address space using virtual memory at each node. This approach requires a major change in the OS support. Other approach includes the compiler support for out-of-core arrays [3], where the programmer's effort in modifying the code will be replaced by the compiler. The programmer is alleviated from the burden in recoding the program and only compiler directives need to be inserted by programmers. This compiler approach must employ some form of runtime support for data parallel programming.

Memory Copy Time in Collective Communications

In the collective communication, usually the data in the user space must be copied to the MPI system buffer before it can be transmitted. The speed degradation from the effect of MCT is illustrated in Fig. 7. As the value of MCT increases, the speed continues dropping. The speed drop is inversely proportional to the increase of MCT. However, different programs experienced different amounts of MCT even with the same speed performance.

Based on our measurement, the MCT from the STAP experiments contribute as high as 50%, 40%, and 90% of the total communication overhead on the SP2, Paragon, and T3D, respectively. Software and hardware approaches to reducing the MCT in collective communication operations were proposed previously. The *Active Messages* [11] offers a latency hiding means to cope with this MCT problem. *Copyless communication* was proposed in [10] where data are copied from the sender space to that of the receiver node based on a virtual memory concept.

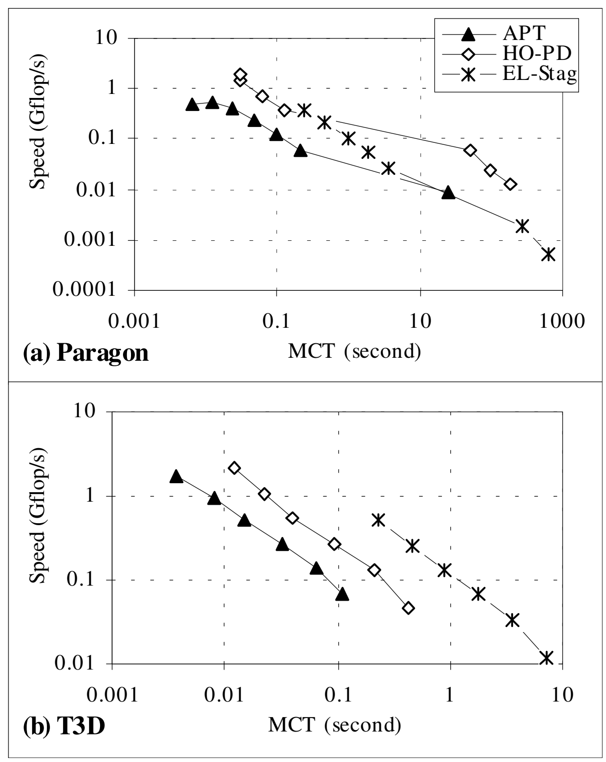

Figure 7. Speed variation with MCT on Paragon and T3D.

6. Conclusions

Among all three machines, we observed that no machine has exceeded 10 Gflop/s speed in executing any of the STAP programs up to 128 node. In general, the SP2 achieved the highest sustained speed on all machine sizes, followed by the T3D, and then trailed by the Paragon.

The disk access is the major bottleneck in performance. No local disk attached to the node can further increase the disk access latency, since all disk access must transverse the interconnect network, as in the case of Paragon. We have seen that the disk access can be as high as 73% of the overall execution time of a program.

The performance is suffered if the accompanied interconnect network is relatively slow, as in the case of SP2. The performance bottleneck on SP2 is on its multistage network. A better network for SP2 is more important than a faster processor.

The Paragon suffers from the DSD effect such the aggregate bandwidth is degraded significantly due to limited local memory to contain relatively large message. The T3D has better aggregate bandwidth than the other two MPPs.

Communication is rather expensive on the MPPs. The communication performance is heavily tied to the switched network latency and bandwidth, as well as to the CCR, which depends on both network traffic and node granularity. The aggregate bandwidth depends on both network bisection bandwidth, the message length, and operation type.

The T3D has the most stable and low CCR among the three MPPs. The low CCR leads to a better speed performance. The 3-D torus topology in T3D network ensures short connection paths as well as a high bisection bandwidth. However, T3D requires a more powerful processor to further increase the performance.

The CCR for Paragon is very high when DSD is in effect. Otherwise, the 2-D mesh network on Paragon is capable of catching up with the processor requests for data.

Generally speaking, for fixed size problem, the speed increases with increasing machine size. The speed begins to drop when the machine size is so large such that the communication shadows the benefit obtained from increasing processors. Disregarding the DSD effect, both Paragon and T3D show a rather stable CCR for fixed workload problems. In general, if the CCR value is small, higher speedup can be achieved with some performance tuning. With large CCR, the parallel algorithm should be redesigned.

In our experiments, the SP2 achieved the highest overall utilization rate of around 30%. The Paragon has the lowest utilization range among three machines. It only achieved 15.6% utilization at machine size 128. The T3D has very stable utilization rate of 22% for all machine sizes.

The DSD effect may severely degrade the MPP performance. It may degrade the performance up to 6 times slower in our experiment. The PMO problem may destroy the portability of the application codes. Based on our measurement, the MCT from the STAP experiments contribute as high as 50%, 40%, and 90% of the total communication overhead on the SP2, Paragon, and T3D, respectively.

We conclude that the computation power is the most important factor in the STAP application. The 3D-torus network of T3D shows the best communication performance. It outperforms others timing-wise due to its three dimensional topology. The Paragon 2D-mesh network suffers from the four-link-per-link topology and hence the congestion in the network is more severe. This is further worsened by the paging structure of the system. The SP2 architecture is scalable, but a much higher bandwidth network is required to further improve the performance.

References

[1] Bhat, P., Y. Lim, and V. Prasanna; "Scalable Portable Parallel Algorithms for STAP," *Proc. of the Adaptive Sensor Array Processing Workshop,* MIT Lincoln Lab.,March 1996.

[2] Bond, R.; "Measuring Performance and Scalability Using Extended Versions of the STAP Processor Benchmarks," Technical report, MIT Lincoln Laboratory, December 1994.

[3] Bordawekar, R., A. Choudhary, K. Kennedy, C. Koelbel, and M. Paleczny; "A Model and Compilation Strategy for Out-of-Core Data Parallel Programs," *Proceedings of the Fifth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, *SIGPLAN NOTICES*, August 1995, pp. 1-17.

[4] Cameron, K., L.J. Clarke, and A.G. Smith; "CRI/EPCC MPI for Cray T3D," http://www.epcc.ed.ac.uk/t3dmpi/Product/, August 1996.

[5] Day, J.; "Tutorial on Digital Adaptive Beamforming," IEEE National Radar Conference, Martin Marietta Corp., Utica, NY, April 22, 1993.

[6] Hwang, K., C. Wang, and C.- L. Wang; "Evaluating MPI Collective Communication on the SP2, T3D, and Paragon Multicomputers," *Third Int'l Symposium on High-Performance Computer Architecture* (HPCA-3), San Antonio, Texas, Feb. 1-5, 1997.

[7] Hwang, K., and Z. Xu; "Scalable Parallel Computers for Real-Time Signal Processing," *IEEE Signal Processing*, Vol. 13, No. 4, July 1996, pp. 50-66.

[8] Hwang, K., Z. Xu, and M. Arakawa; "Benchmark Evaluation of the IBM SP2 for Parallel Signal Processing," *IEEE Trans. Parallel and Distributed Systems*, May 1996, pp. 522-536.

[9] Message Passing Interface Forum; *MPI: A Message-Passing Interface Standard*, version 1.1, June 12, 1995, ftp://ftp.mcs.anl.gov/pub/mpi/mpi-1.jun95/mpi-report.ps, August 1996

[10] Murayama, H., S. Yoshizawa, Aimoto, H. Inouchi, S. Murase, T. Hayashi, and H. Iwamoto; "A Study of High-Performance Communication Mechanism for Multicomputer Systems," *10th International Parallel Processing Symposium*, Honolulu, Hawaii, April 15-19, 1996, pp. 76-83.

[11] von Eicken, T., D.E. Culler, S.C. Goldstein, and K E. Schauser; "Active Messages: A Mechanism for Integrated Communication and Computation," *Proceedings of the 19th International Symposium on Computer Architecture*, Gold Coast, Australia, May 1992, pp. 256-266.

[12] Xu, Z., and K. Hwang; "Early Prediction of MPP Performance: SP2, T3D, and Paragon Experiences," *Parallel Computing*, October 1996.