

Design of a Portable Cluster Supercomputer for Particle Image Velocimetry Data Processing

Thomas Hauser* and Mark A. Perl†
Utah State University, Logan, UT 84322, 4130

DOI: 10.2514/1.35550

In this paper, we present the design of a portable cluster supercomputer created specifically for processing particle image velocimetry image data. To make this computer system portable for a laboratory environment, it is designed to run on minimal power, to be lightweight, and portable. The hardware configuration consists of 12 processing nodes with a total of 48 processor cores, one master node with five terabytes of disk storage, and a gigabit ethernet interconnect. The total configuration along with a rugged transportation case has an approximate gross weight of 390 lb and works on one standard 120 V, 20 A electric circuit. With this cluster computer, a speedup of 28 relative to standard serial processing of a large particle image velocimetry data set was achieved.

I. Introduction

THE use of optical measurement techniques increases in experimental fluid dynamics because of their nonintrusive nature [1]. Recent advances in digital photography and solid-state lasers make it possible to acquire image data sets at up to 12,000 frames per second [2]. However, as the ability to acquire large samples very quickly has been realized, processing technology has not kept pace. A typical data acquisition computer would require several hours to process the data that can be acquired in one second with a high-speed imaging system.

In this paper, we demonstrate how to improve the processing times for time-resolved stereo particle image velocimetry (TRSPIV) data sets using a custom-designed cluster computer. Recent results [3,4] show that TRSPIV can reveal the full three-dimensional (3-D) structure of certain types of flow, which was previously only possible numerically with the direct numerical simulation approach. However, the current state of the art for processing these data sets falls far short of the state of the art in high-speed data acquisition.

TRSPIV evolved from the original particle image velocimetry (PIV) approach. The PIV process generates a planar two-component velocity vector field at an instant in time. Stereo PIV is performed by adding a second camera to a standard PIV system. The two cameras observe the laser sheet from the sides at angles allowing them to sense flow perpendicular to the sheet in addition to flow in the plane of the laser sheet. As a result, all three components of velocity can be computed in the plane of the laser sheet. TRSPIV is realized, when the images can be acquired sufficiently fast, to resolve the time-scales of interest.

As the image pairs acquired can be processed independently from each other, parallel computing using a Beowulf cluster supercomputer [5] to decrease the postprocessing time is possible. Beowulf cluster supercomputers are built from commodity parts and provide low-cost parallel computing power. McCray et al. [6] have used cluster supercomputers at Wright-Patterson Air Force Base to cut the processing time for PIV data sets by a factor of 80 compared to a single processor. In their presentation, McCray and his coworkers stated that the main disadvantage

Received 7th November 2007; accepted for publication 3 May 2008; Copyright © 2008 by Thomas Hauser and Mark A. Perl. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 1542-9423/08 \$10.00 in correspondence with the CCC.

* Assistant Professor, Mechanical and Aerospace Engineering, *thomas.hauser@usu.edu*, Associate Fellow AIAA

† Graduate Student, Mechanical and Aerospace Engineering, Student Member AIAA

of their approach is the limited network bandwidth between the cluster supercomputer in the data center and the PIV system in their laboratory regarding the data transfer. Our approach is to integrate the TRSPIV system with a cluster supercomputer to process the data in the experimental laboratory. This cluster is specifically designed for the data processing of large experimental image data sets. By integrating the PIV system and the Beowulf cluster, communication time is minimized, thus speeding up the PIV image data processing. It also allows the use of this system in locations without a network connection.

II. Designing a Cluster Supercomputer for PIV Image Processing

A. The Integrated Data Acquisition and Parallel Processing System

The concept for the proposed system consists of several compute nodes and the cluster server node with direct attached storage as shown in Fig. 1. The data acquisition PC, the cluster server, and several compute nodes are connected through a high-speed network, for example Gigabit ethernet. The data acquisition system receives the data from the camera and stores it on the cluster server. The server distributes the image pairs to the compute nodes for processing. As the processing of one PIV image pair or two SPIV image pairs is independent from the processing of all others, no communication between the compute nodes is necessary. This mode of parallelism is called “embarrassing parallel” [7].

B. Design Requirements

To design the cluster for the single task of processing PIV image data, the requirements are defined first. The main constraints for the cluster are:

- 1) The system needs to operate on one single 120V and 20A electric circuit. This circuit is normally available in a laboratory environment. Standard high-performance cluster computers available from most vendors run generally on 240V circuits.
- 2) No special air-conditioning is available for the operation of the cluster. This requirement is also different from regular cluster supercomputers, which are housed in data centers with specialized cooling systems.
- 3) The budget is limited to \$60,000.
- 4) The cluster needs to be portable, so that it can be transported between different wind tunnel sites.
- 5) The system needs to run the DaVis [8] imaging software from LaVision. This means that the processor architecture is restricted to the $\times 86$ architecture [9] as the code for the DaVis clients under Linux is binary only. The choice of the $\times 86$ architecture eliminates the Cell BE processor [10] or the MIPS [11] architecture.

An initial conceptual design revealed that the main design restriction for this cluster will be the electric power limitation. Therefore, we limited our processor choice in the design to low power variants.

C. Hardware Solutions Considered in the Design

The low power processors preselected to meet our electric power constraint are tabulated in Table 1.

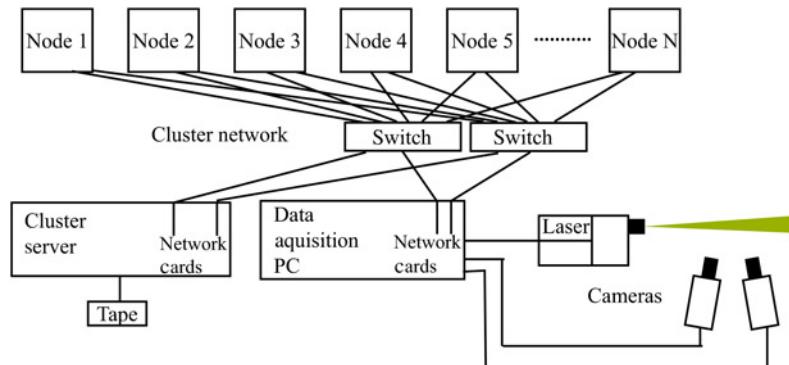


Fig. 1 The concept of the integrated TRSPIV and parallel processing system.

Table 1 The low power processors considered in the cluster design

Name	Speed/core (GFLOPS)	Number of cores	Power W	Cost, \$
Intel Quad Core Xeon X3210	2.33	4	105	785.00
Intel Dual Core Xeon 5148LV	4.66	2	40	670.75
Intel Core 2 Duo E6700	5.34	2	65	560.00
Intel Core 2 Duo E6600	4.8	2	65	375
AMD Opteron 275 HE	4.4	2	55	1019.00
AMD Opteron 260 HE	3.2	2	55	461.00
Intel Core 2 X6800	5.86	2	75	1075.00
AMD Opteron 240 EE	2.8	2	33	210

The Intel processor line with the characteristics needed for a low power cluster is the Dual-Core Intel Xeon processor 5100 series [12] and the Quad-Core Intel Xeon X3210. In general the 5100 series processors deliver two times the performance and over two times the performance/Watt of a previous-generation dual core Intel Xeon processors.

The AMD Opteron processors use energy-efficient DDR2 memory [13]. In addition, these processors are designed to maximize the performance to power ratio, giving the good performance at 68 W for 2.4 GHz [14].

D. Benchmarking Multicore Processors using a PIV Kernel

To determine the weaknesses and characteristics of the different processor architectures for the compute nodes of the cluster, a simple PIV code, named PPIV, was used. The PPIV program offers portability to all compute architectures while performing the same basic computational operations as more sophisticated commercial implementations. The main phases of a typical PIV code include processing the cross-correlation of the interrogation regions using fast Fourier transforms (FFTs), postprocessing of the displacement field, and adaptive local window shifting. Because of its performance and use in commercial PIV applications, we used the Fastest Fourier Transform in the West (FFTW) [15,16].

1. PIV Benchmark Algorithm

The PPIV program follows the procedures outlined in Bolinder [17] for the cross-correlation, the postprocessing, and the adaptive local window shifting. The cross-correlations are performed on interrogation windows of the image to determine the most likely velocity vector for that subregion. This cross-correlation is performed as

$$C = \text{Re}[\text{FFT}^{-1}\{\text{FFT}^*(IA_1) \cdot \text{FFT}(IA_2)\}] \quad (1)$$

where $*$ denotes the complex conjugate, and IA_1 and IA_2 are the interrogation windows. The interrogation window consist of 64 pixels in each direction which overlap one another by 50%. During the PIV process the computation of cross-correlations is, for the most part, an expensive process. The PPIV program further increases the accuracy by shifting the second window in the estimated direction of the velocity vector by a known amount; this is known as adaptive local window shifting. A first pass with no shift provides the estimate of how much to shift the window on the second pass. This allows the program to reduce the interrogation window size to 32 pixels, therefore doubling the spatial resolution.

2. Speedup of the PIV Code on Dual-Core Systems

All of the processors in Table 1 are dual-core processors. To verify that dual- or quad-core processors are appropriate choices for the design of the cluster, two, three, and four instances of the PPIV benchmark program were run on test compute nodes. These compute nodes consisted of two dual-core processors. The test determines how efficient the dual-core processor architecture can run several instances of the PIV code. Ideally, the wall clock times for the runs should be the same, regardless of the number of program instances, as long as there are not more program copies running than cores. Slowdown can occur if all cores access the main memory and memory bandwidth becomes a performance-limiting factor. In Table 2 the run time in seconds of the AMD Opteron 265 node and of the Intel Dual-Core Xeon 5148LV with two dual-core processors are compared.

Table 2 Comparison of run times of the PPIV code for the AMD dual-core Opteron 265 and the Intel dual-core Xeon 5148LV

Number of processes	AMD dual-core Opteron 265	Intel dual-core Xeon 5148LV
1	81.46 [s]	56.91 [s]
2	83.30 [s]	56.74 [s]
3	82.37 [s]	57.02 [s]
4	80.34 [s]	58.98 [s]

The AMD Opteron 265 and the Intel Xeon 5148LV operate at a clock speed of 1.8 GHz and 2.33 GHz respectively. Table 2 shows that the Intel processor is faster than the AMD processor for this task. This speed advantage of the Intel processor can be explained through the faster clock speed. The table also shows that there is virtually no overhead for running a copy of the PIV program on each core of a node. This means the dual-core or even quad-core architecture is a good choice for running multiple instances of a PIV program on a multiprocessor/multicore node.

III. Design the Cluster for PIV Data Processing

Finding the right cluster architecture for the PIV processing application is a difficult problem, even for someone trained in building computer systems. A wide variety of components and network topologies is available, but it is not clear which combination will support the application best. To ensure a good selection of components for the cluster, we have used the cluster design rules (CDR) expert system [18]. We populated our own database with processor choices from Table 1. The expert system requires input about the application requirements and the resource available for the cluster deployment. We used the following cluster requirements as inputs for the CDR expert system:

- 1) **Memory requirements:** In this section of the CDR expert system, the memory requirements for data, code, and operating system for the PIV application is provided.
 - a) **Data in main memory for the whole cluster:** 512 MB to 1 G; this provides enough memory to hold the four images in RAM for the stereo cross-correlation, the operating system and cache for I/O operations. This amount of memory is more than enough for the image size, but it allows for caching the network file system traffic in memory.
 - b) **Compiled program size:** 64 MB to 128 MB; this value is based on the DaVis program, the commercial program used, on Linux. The programs size was not based on the size of PPIV because of its smaller size.
 - c) **Memory bandwidth:** min 0.5 GB/s; preliminary benchmarks indicated that a higher bandwidth increases the processing speed.
- 2) **Hard disk drive requirements:** The disk requirements for the compute nodes are specified here.
 - a) **Swap space:** none, as our data sets fit in main memory.
 - b) **Storage within the cluster:** none; to increase the processing speed none of the images will be stored on the hard disk while they are being processed.
- 3) **Networking requirements:** The communication pattern; in our case there is no communication between compute nodes, define the networking requirement for the cluster.
 - a) **Network latency:** 50 to 100 μ s.
 - b) **Bisection bandwidth:** 400 MB/s to 500 MB/s; the maximum bandwidth from the camera to the computer is only 400 MB/s the communication within the cluster should be of the same order of magnitude.
 - c) **Communication with other nodes:** each node processes its own a set of images, therefore each node only needs to talk to the master node.
- 4) **Physical parameters:** The power, cooling, and space constraints for the cluster are specified in this input section of the expert system.
 - a) **Rack space:** the cluster needs to be mobile, so only one single rack is allowed.
 - b) **Power:** 15 to 20A; the cluster needs to be able to function in a laboratory environment, with only standard 120V 15–20A circuits available. Such a standard power outlet has only 20A peak and 18A continuous.

Table 3 Cluster design for each of the selected processors

Processor name	Powered 1U nodes possible on 20A	Processors/1U	Processor cores/processor	Total amps/1U	Watts/1U
Intel Quad-Core Xeon X3210	7	2	4	2.37	284.705
Intel Dual-Core Xeon LV 5148	13	2	2	1.31	157.69
Intel Core 2 Duo E6700	23	1	2	0.78	93.70
Intel Core 2 Duo E6600	23	1	2	0.78	93.70
AMD Opteron 260 HE	12	2	2	1.458	174.995
AMD Opteron 275 HE	12	2	2	1.458	174.995

Table 4 The total number of cores and peak performance for the given processors running on a 120 V 20A electric circuit

Processor name	Total number of cores	Processor speed, GHz	Usable GFLOPS
Intel Quad Core Xeon X3210	56	2.13	119.28
Intel Dual Core Xeon LV 5148	52	2.33	121.16
Intel Core 2 Duo E6700	46	2.66	122.36
Intel Core 2 Duo E6600	46	2.4	110.4
AMD Opteron 260 HE	48	1.6	76.8
AMD Opteron 275 HE	48	2.2	105.6

c) **Air conditioning:** 0.25 tons to 0.5 tons; as the cluster can be plugged in anywhere, it is assumed that only minimal cooling will be available. The low level of cooling specified can be accomplished by standard cooling in buildings.

When running the CDR expert system with the input specified above, it produces 25–50 different designs per processor. These are then narrowed down to the best design with three different criteria. First, any design that did not use a Gigabit network was dropped. Second, to ensure enough memory each design has 1 GB of RAM per core. Third, the system with the least power usage was then selected.

To find the power needed per 1U in each case, we started by subtracting the power needed for the switches (27.5 W per switch) from the total power. This number was then divided by the total number of powered nodes in the cluster. This approach is slightly modified for any solution that has the Intel S3000PT motherboard. This board is designed to have two motherboards boards per a single 1U case. So the number of powered nodes is cut in half and the number of processors per node is raised to two. Using these steps the results for each processor in Table 1 is tabulated in Table 3.

The total number of cores and the maximum processing speed of the cluster as given by the CDR expert system is given in Table 4. This table shows that the Intel Quad-Core processor provides the most cores for the given power constraints and all Intel processors are very similar in the usable GFLOPS metric given by the CDR expert system. Therefore we have selected the Intel Quad-Core Xeon X3210 for our PIV processing cluster.

The graphical output of the CDR expert system of one design using the Intel Quad-Core Xeon X3210 is shown in Fig. 2. The output of the CDR expert system also provides the user with a detailed cluster configuration and pricing, as well as operating costs which can be an important factor in the purchase decision of a cluster supercomputer.

IV. Cluster Implementation

The designed cluster was not available as a standard configuration in the low power and portable implementation from any cluster vendor. The configuration for the compute nodes is listed in Table 5. This design incorporates two completely independent systems into one rack unit (1U), which is 44.45 mm.

In the design [19], seven chassis satisfied the power constraint. To accommodate the storage requirements, one chassis was replaced by a master/storage node. The configuration for this master/storage node is listed in Table 6. This master/storage node provides five terabytes of disk storage for PIV data to the data acquisition system and the cluster, and serves as the management node for the compute nodes.

Powered nodes	14	14	14
Cold spares	1	1	1
NICs	30	30	30
Cables	30	30	30
Switches	2	2	2
Processors	15	15	15
Motherboards	15	15	15
Disk drives	0	0	0
Cases	15	15	15
Racks	1	1	1
NICs/node	0	0	1
Processors/node	1	1	1
Processor Cores/Chip	4	4	4
Usable GFL OPS	119.28	59.64	119.28
Usable memory (MB)	55328.00	55328.00	112672.00
GB/s per (GFL OPS)	2.54	2.54	5.07
Usable disk (GB)	0.00	0.00	0.00
Total Amps	34.23	31.68	39.74
Total Watts	2053.67	1900.51	2384.50
Air Conditioning (tons)	0.59	0.54	0.68
Network latency (us)	38.00	38.00	38.00
Network bandwidth (Mb/s)	28000.00	14000.00	42000.00
Total cost	28541.94	21859.39	38144.79
Metric Value	1.386	0.1928	1.386

The components used, and their costs, are summarized here:

Quantity	Component Description	Part Price	Extended Price
30	Cat5 cable for Gigabit Ethernet	\$2.25	\$67.50
2	D-Link DGS-1024D 24-Port Gigabit Switch	\$226.75	\$453.50
15	Intel quad-core X3210	\$762.00	\$11430.00
15	Intel S3000PT	\$235.97	\$3539.55
60	Kingston (KVR667D2E5K2/1G)	\$85.49	\$5129.40
15	Ever Case R193-PTS	\$428.17	\$6422.55
1	APC NetShelter 47U Rack Enclosure AR2104BLK	\$1499.99	\$1499.99
Total			\$28541.94

Estimated annual operating costs assuming the cluster operates 24 hours a day, 7 days per week:

Cost item	Amount	Unit Cost	Total Cost
Electrical power	18002.47	KWh	\$0.053270 /KWh + \$10.00/mo.
Electrical power for air conditioning	2571.78	KWh	\$0.053270 /KWh
Space cost (including 2' between racks in 8' rows)	9.00	ft. ²	\$15.60 /(ft. ² · mo.)
Total			\$2900.79

Fig. 2 The selected design result for the Intel Quad-Core Xeon X3210 as given by the CDR expert system.

Table 5 The final design for one compute node with an approximate configured gross weight of 30 lb

Part	Description	No. of parts
Case	1U chassis	6
Motherboard	Intel server board S3000PT	12
Processors	Intel Quad-Core Xeon X3210 2.13 GHz - LGA775 Socket -L2 8 MB	12
RAM	1 GB ECC DDR2-667 memory module (low power) (1 GB per core)	48
Hard drive	None	—
Network card	Integrated 10/100/1000 ethernet	24

Table 6 The final design for the master/storage node with an approximate configured gross weight of 80 lb

Part	Description	No of parts
Case	3U chassis with 650w power supply	1
Motherboard	Intel server board S3000PT	1
Processors	Intel Quad-Core Xeon X3210 / 2.13 GHz - LGA775 Socket - L2 8 MB	1
RAM	1 GB ECC DDR2-667 memory modules (low power)	4
Hard drives	500 GB raid edition SATA hard drive (5.0Tb usable storage in RAID-6)	12
RAID	3Ware 9650 16-port SATA raid controller	1
Optical	Slim-line DVD-RW drive	1
Network card	Integrated 10/100/1000 ethernet	1

To finalize the cluster, two additional hardware components were required. First, a 24-port Gigabit ethernet switch was selected to provide connectivity between all nodes. Second, a shock-mounted traveling case/rack replaces the standard rack to enable portability and durability of the cluster in a laboratory environment. A picture of the final delivered system is shown in Fig. 3.

V. Benchmarking the Implemented Cluster Supercomputer

To test the performance of the portable cluster supercomputer, two different benchmarks were used. These benchmarks were chosen from typical setups used in experiments and were used to benchmark the cluster. Each benchmark used the same data set consisting of 1024 PIV image files containing PIV image data in the LaVision IM7 file format. Each file contains one image pair from each camera, that means four images per file. Each image in the file has a resolution of 1024 by 1024 pixels at 10 bits black and white. The difference between the benchmarks is the manner in how the images are processed. First, to test the full computational power of the system, a computationally-expensive processing job using the commercial code produced by LaVision, called DaVis, as described in section A, was performed. Second, a benchmark that more closely matches our PPIV kernel was chosen to enable a comparison between the different codes. This benchmark is called “cheap” because it is computationally less expensive than the previous one and is described in section B.

A. “Expensive” Benchmark

The computationally “expensive” benchmarks consists of six different passes over the images sets. The first two passes use a 64×64 pixel oblong vertically stretched interrogation window with a ratio of 2:1 and a 50% overlap. However, this stretching greatly increases the amount of processing time needed for each pass over the images. The third, fourth, fifth, and sixth pass use the same interrogation window except the interrogation window size is reduced to 32×32 pixels for the third and forth pass, and 16×16 pixels for the fifth and sixth pass. In addition a smoothing/post processing filter was used between each image pass. This filter consisted of a median filter where the vectors are removed and replaced if the vector is greater than two standard deviations from its surrounding vectors.

B. “Cheap” Benchmark

The computationally “cheap” benchmark consists of three passes over the image sets. All three passes used a square interrogation window and a 50% overlap. The first pass uses a 64×64 interrogation-window size. The second and



Fig. 3 The implementation of the low-power PIV cluster supercomputer.

third pass use 32×32 pixels, and 16×16 pixels, respectively. This setup also uses the same median filter between each pass as was used in the expensive setup (section A).

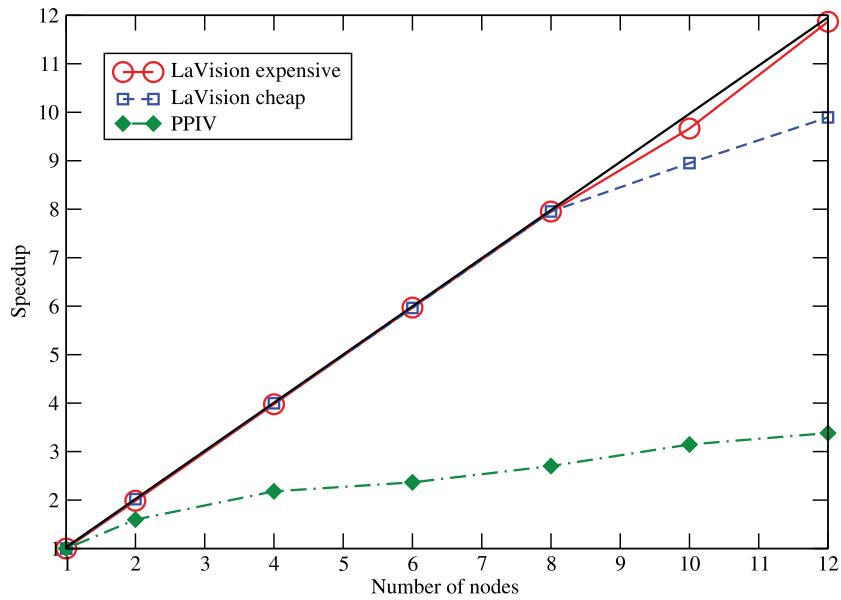
C. Benchmark Results

Running DaVis and PPIV on the cluster with the benchmark setups documented in Secs. A and B, we obtained the run times shown in Table 7. Each node contains one four core processor. We see the expected time difference between the “expensive” and the “cheap” benchmark.

As DaVis and PPIV use a very similar algorithm for the “cheap” benchmark, we expected the wall clock times to be similar. As Table 7 shows there is a large difference of processing time between the two codes. We found by observing the CPU load that the DaVis code uses the four core processor only fully in the expensive setup. During the “cheap” benchmarks runs only a 40% to 50% load was achieved. In comparison, the PPIV code uses the CPU fully, because each core runs one instance of the code. The DaVis approach of running only one instance of the code

Table 7 The time values from running 1024 images with varying node numbers using DaVis and PPIV

No. of nodes	LaVision: “expensive” benchmark	LaVision: “cheap” benchmark	PPIV
1	12 h 50 m 4 s	54 m 45 s	8 m 9 s
2	6 h 8 m 58 s	27 m 8 s	5 m 6 s
4	3 h 13 m 35 s	13 m 24 s	3 m 44 s
6	2 h 27 m 58 s	9 m 11 s	3 m 26 s
8	1 h 36 m 52 s	6 m 53 s	3 m 1 s
10	1 h 19 m 41 s	6 m 7 s	2 m 35 s
12	1 h 4 m 53 s	5 m 12 s	2 m 24 s

**Fig. 4** Speedup for DaVis and PPIV for an image set of 1024 images.

on the processor and then parallelize over the cores is not able to use a four core processor efficiently for the “cheap” benchmark. The main reason for the inefficiency is probably communication overhead between the cores. Running an instance of DaVis per core would theoretically solve this problem and we are currently working on a solution. Even with this not optimal implementation of the commercial code, we still see a considerable reduction in run time when using the cluster to process PIV data.

Another performance measure for a parallel system is speedup. Speedup is defined as

$$\text{speedup} = \frac{\text{Sequential execution time}}{\text{Parallel execution time}}, \quad (2)$$

where the parallel execution time is listed in Table 7 for compute node numbers from two to twelve. The sequential execution time is the time for a single node. These speedup results for each benchmark case are plotted in Fig. 4. For comparison the solid black line shows the ideal speedup.

Looking at the speedup curve for the PPIV program, we see the curve quickly flattens out. As the run time for the PPIV code are so small the overhead in distributing the images dominates in this case. The cluster is basically spending most of its time in distributing the images to the compute nodes and receiving the results. The performance of PPIV could be improved by improving the network performance.

In comparison to the PPIV code, the processing of image data with DaVis performs much better. The DaVis “expensive” benchmark scales very well with the number of compute nodes because it shows nearly ideal speedup behavior for all number of nodes. This gives us a total speedup on the cluster of 11.9 and a speedup of 28.7 in comparison to a single Intel Pentium 4 CPU running 3.00 GHz, and 1 GB RAM. For the DaVis “cheap” run, the speedup curve starts to flatten out at around 10 nodes, but does still produce a speedup of 10.5 with 12 nodes. These results demonstrate that the cluster computer is well designed for processing PIV image data sets using the DaVis program.

D. Electric Power Usage

By fire code, any electrical system must not run continuously using more than 80% and should not peak above 100% of the available power. This means that the continuous power consumption of the cluster cannot be more than 16A (1920W), and that the system cannot peak above 20A (2400W). To ensure that the cluster operates within those limits, it was tested under realistic operating conditions. Node and entire system tests were performed using the benchmark data set described in Sec. V running the computationally expensive setup.

1. Compute Node Power Usage

As there are two nodes in each 1U chassis, it was not possible to measure the power one node uses. We tested two nodes (one 1U chassis), and assumed that each node draws half of the measured power. For the test we plugged one 1U chassis in a separate electric circuit and measured the power with a power analyzer. The system tested consisted of:

- 1) 2 quad-core Intel Xeon X3210 processors;
- 2) 8 Gigabytes RAM;
- 3) 4 Gigabit ports;
- 4) no hard drive.

The power usage of two nodes was monitored from power up through the PIV process. The results showed that a single node has the following power consumption:

- 1) 131 W peak power consumption during boot up;
- 2) 75.5 W power consumption while idle;
- 3) 143 W peak power consumption during the benchmark run.

This would allow for 13 compute nodes, but we must consider the master/storage node.

2. Complete Cluster Power Usage

The cluster was plugged into a single circuit and an amp meter measured the current. The power was then calculated by multiplying by an assumed 120V. The boot process consisted of booting the master node first, and then each node was powered up in 2 s intervals. Just like the single node load test, the power usage was monitored from power up through the PIV process and yielded the following results:

- 1) 16.0A (1920W) peak during boot up;
- 2) 14.5A (1740W) while idle;
- 3) 16.9A (2028W) peak during benchmark run.

During the benchmark run the current fluctuated between 15.4A and 16.9A, with an average current below 16A. We conclude based on these results that the cluster will run on a single, 20A circuit. For the occasion when the cluster is to be used with only a 15A circuit is available, we continued the test by shutting down one node at a time, until the system was at 12A (80% of a 15A circuit) continuous. This was achieved after three nodes were shut down. For a 15A circuit, the cluster should be run with nine compute nodes and the master node.

VI. Conclusion

We have designed a portable cluster supercomputer created specifically for processing PIV data. The design requires the system to run on minimal power and be lightweight for portability. To minimize the cluster’s power and cooling requirements, we limited the power to one 120V circuit running at 16A continuous and 20A peak. This is based on fire code standards for a 120V and 20A circuit. The final cluster design uses the Intel Quad-Core Xeon X3210 low-power processor. This hardware setup allows for a total system consisting of 12 processing nodes with a

total of 48 CPU cores, one master node with 5 TB of storage, and a Gigabit ethernet connection between the nodes. The total configuration along with the case has an approximate gross weight of 390 lb.

To maintain the portability of the system, a basic computer rack case was replaced with a traveling case. This case system can be completely closed and has removable wheels, making it instantly ready for shipping. This system can operate in various locations.

To ensure the system will work in realistic conditions, a PIV power load test was performed. The load test shows that the cluster runs at the required power levels with a maximum power of 16.9 A (2028 W) and an average power of below 16 A.

To determine the performance increase, the cluster was benchmarked with a large data set consisting of 1024 image pairs. Using the LaVision DaVis code, we found a speedup of 12 on the cluster and a speedup of 28.7 compared to an existing system. This does not keep pace with the current acquisition hardware that can take 3000 images a second, but it is a good step in this direction. Further, by connecting the PIV system directly to the cluster, the communication time lag between the two systems has been reduced dramatically.

Acknowledgments

This work was supported in part by the National Science Foundation under CTS-0521621. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation. The authors would like to thank B. Smith and Z. Humes from the experimental fluid dynamics laboratory at Utah State University for the benchmark data set, and their suggestions for the cluster design.

References

- [1] Feldmann, O., and Mayinger, F., *Optical Measurements Techniques and Applications*, 2nd corrected ed., Springer-Verlag, Berlin/New York/Heidelberg, 2001.
- [2] Towers, D., and Towers, C., *Particle Image Velocimetry*, Vol. 112 of *Topics in Applied Physics*, Chap. High-Speed PIV: Applications in Engines and Future Prospects, Springer-Verlag, Berlin/New York/Heidelberg, 2008, pp. 345–361.
[doi: 10.1007/978-3-540-73528-1_18](https://doi.org/10.1007/978-3-540-73528-1_18)
- [3] van Doorne, C. W. H., and Westerweel, J., “Measurement of Laminar, Transitional and Turbulent Pipe Flow using Stereoscopic-PIV,” *Experiments in Fluids*, Vol. 42, No. 2, 2007, pp. 259–279.
- [4] Sung, J., and Yoo, J. Y., “Three-dimensional Phase Averaging of Time-resolved PIV Measurement Data,” *Measurement Science Technology*, Vol. 12, No. 6, 2001, pp. 655–662.
[doi: 10.1088/0957-0233/12/6/301](https://doi.org/10.1088/0957-0233/12/6/301)
- [5] Becker, D. J., Sterling, T., Savarese, D., Dorband, J. E., Ranawak, U. A., and Packer, C. V., “BEOWULF: A Parallel Workstation for Scientific Computation,” *International Conference on Parallel Processing, ICPP Workshop on Challenges for Parallel Processing*, CRC Press, 1995.
- [6] McCray, T. W., Estevadeordal, J., and Puterbaugh, S., (eds.), “Parallel Computing for Linux Clusters Application to Particle Image Velocimetry,” *Proceedings of the 43rd Aerospace Sciences Meeting and Exhibit*, AIAA, Reston, VA, 2005.
- [7] Breen, B., Weidert, C., Lindner, J., Walker, L., Kelly, K., and Heidtmann, E., “Invitation to Embarrassingly Parallel Computing,” *American Journal of Physics*, Vol. 76, No. 4, 2008, pp. 347–352.
[doi: 10.1119/1.2834738](https://doi.org/10.1119/1.2834738)
- [8] “DaVis product description,” <http://www.lavision.de/products/davis.php> [retrieved 21 April 2008].
- [9] Alpert, D., and Avnon, D., “Architecture of the Pentium Microprocessor,” *Micro, IEEE*, Vol. 13, No. 3, 1993, pp. 11–21.
[doi: 10.1109/40.216745](https://doi.org/10.1109/40.216745)
- [10] “IBM, Cell Broadband Engine processor-based systems White Paper,” IBM, September 2006.
- [11] Hennessy, J., Jouppi, N., Przybylski, S., Rowen, C., Gross, T., Baskett, F., and Gill, J., “MIPS: A Microprocessor Architecture,” *SIGMICRO Newsletter*, Vol. 13, No. 4, 1982, pp. 17–22.
[doi: 10.1145/1014194.800930](https://doi.org/10.1145/1014194.800930)
- [12] “Dual-Core Intel Xeon Processor 5100 Series,” <ftp://download.intel.com/products/processor/xeon/dc51kprodbrief.pdf>, Intel Corporation [retrieved 5 November 2007].
- [13] “Product Brief: Next-Generation AMD Opteron Processor with DDR2 and AMD Virtualization,” http://www.amd.com/us-en/Processors/ProductInformation/0,30_118_8796_14309,00.html, Advanced Microsystems, Inc. [retrieved 5 November 2007].

- [14] "Understanding Next-Generation AMD Opteron Processors Model Numbers," http://www.amd.com/us-en/Processors/ProductInformation/0,30_118_8796_14266,00.html, Advanced Microsystems, Inc. [retrieved 5 November 2007].
- [15] Frigo, M., and Johnson, S. G., "FFTW: An Adaptive Software Architecture for the FFT," *Proceedings of 1998 IEEE International Conference on Acoustics Speech and Signal Processing*, Vol. 3, Institute of Electrical and Electronics Engineers, New York, NY, 1998, pp. 1381–1384.
- [16] Frigo, M., and Johnson, S. G., "The Design and Implementation of FFTW3," *Proceedings of the IEEE*, Vol. 93, No. 2, 2005, pp. 216–231, special issue on "Program Generation, Optimization, and Platform Adaptation." [doi: 10.1109/JPROC.2004.840301](https://doi.org/10.1109/JPROC.2004.840301)
- [17] Bolinder, J., "On the Accuracy of a Digital Particle Image Velocimetry System," Technical Report ISSN 0282-1990, Department of Heat and Power Engineering Division of Fluid Mechanics, Lund Institute of Technology, Box 118, S-221 00 Lund, June 1999.
- [18] Dieter, W. R., and Dietz, H. G., "Designing a Cluster for Your Application," *Computing in Science and Engineering*, Vol. 9, No. 4, 2007, pp. 72–79.
- [19] Perl, M., and Hauser, T., "Processing High-Speed Stereo Particle Image Velocimetry Data with an Integrated Cluster Supercomputer," *45th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Reston, VA, 2007, AIAA Paper 2007-51.

J. A. Mulder
Associate Editor