DISTRIBUTED PROCESSING OF HIGH-SPEED STEREO PARTICLE IMAGE VELOCIMETRY DATA WITH A LOW POWER BEOWULF CLUSTER

ABSTRACT

In this paper, we present our design of a portable cluster supercomputer created specifically for processing PIV image data. To make the system portable for a laboratory environment, the cluster computer is required to run on minimal power and be lightweight. The final cluster design was based on the Intel® Quad-Core Xeon® X3210 low-power processors. This hardware setup allows for a total system consisting of twelve processing nodes with a total of 48 CPU cores, one master node with 5 TB of storage, and a Gigabit Ethernet connection. The total configuration along with the case has an approximate gross weight of 390 lbs and works well on one standard 120-V, 20-Amp electric circuit. With this cluster computer, a speedup of about 28 relative to standard serial processing of a PIV data set could be achieved.

Introduction

Time-Resolved Stereo Particle Image Velocimetry (TRSPIV) is a revolutionary measurement technique with the potential to fundamentally alter the way we approach fluid measurements. Recent results [?] demonstrate that TRSPIV can reveal the full 3-D structure of a flow, which was previously only possible with Direct Numerical Simulation. 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 2-component velocity vector field at an instant in time. Stereo Particle Image Velocimetry 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 possible when the images can be acquired sufficiently-fast to resolve the
time-scales of interest. Recent advances in digital photography and solid-state lasers make it possible to acquire images at up to 3000 frames per second. However, as the ability to acquire large samples very quickly has been realized, processing speed has not kept pace. Our present (and typical) 2-D PIV acquisition computer at USU, purchased recently, would require over five hours to process the data that can be acquired in one second with the TRSPIV system.

To decrease the computational time, parallel processing has been implemented using a Beowulf cluster [?]. Beowulf cluster supercomputers are built from commodity parts and provide low cost parallel processing power. At USU we have developed a low-power Beowulf cluster integrated with the data acquisition system of a TRSPIV system with a low power cluster supercomputer specifically designed for the data processing of large experimental image data sets. This approach of integrating the PIV system and the Beowulf cluster eliminates the communication time, thus speeding up the process. In addition to improving the practicality of TRSPIV, this system will also be useful to researchers performing any PIV measurement where a large number of samples are required. This paper will describe the hardware and software implementation of our approach.

1 System Concept

The system was to consist of a number of computer nodes, the cluster server node and the data acquisition system as shown in Figure 1. In designing this cluster different criteria must be met. Since the system must be portable there are two major constraints: 1) power consumption, and 2) the physical size of the supercomputer system. This cluster system should work in a normal lab environment, so we limited the system to work on a single, 20-Amp circuit. Further, the physical size could not be larger than what two people could easily move. This criteria allows the computer and PIV system to travel together.

The software infrastructure of the cluster will be based on a master-slave configuration. This is implemented using the Warewulf clustering solution [?]. Warewulf is is a Linux cluster solution that is scalable, flexible and easy to use. It manages and distributes Linux systems to any number of nodes in the master-slave relationship. It does this by making the node file system manageable from the master node, and automates the distribution of the node file system during the node boot process. This system also allows the monitoring and controlling of the nodes from one system. The Warewulf system is not limited to just high-performance computing systems, and as such will allow the creation of a file system that will work with the LaVision PIV software (DaVis).

Use of the LaVision software, and most other commercial PIV products, requires a windows-based system to distribute the PIV images to each of the nodes. This requires that the Windows system be able to access the storage file system. This presents a communication problem between the Windows and Linux-based files systems. This same problem has been noticed extensively as the Internet grows larger; both Linux and Windows machines are required to communicate with each other more. To allow for this communication, we will use Samba [?]. Samba is a Linux-based program that can speak to a Windows-based machine like a native.

The parallelization is also based on a master-slave [?] concept. This concept is based on one machine (The Master) sending tasks to other machines (The Slaves) instead of processing the task locally. This process allows for multiple tasks to be performed simultaneously. This method is ideal for PIV processing, since each pair of images can be considered a separate task. To implement this, a program for distributing the images was constructed using the Python [?] programming language with the Ipython1 [?] module. This module facilitates the communication with instances of python on other machines.

Cluster Realization

After designing the cluster from available commodity hardware [?], we found that the Intel®Quad-Core Xeon® X3210 would give the greatest performance while keeping the power consumption and weight low. The final configuration for the computer nodes is described in table 1. This design incorporates two completely independent systems into a 1U1 chassis. Most systems would incorporate a single mother board with both processors. This duel processing system has the potential to reduce communication time between processors. However, with the master-slave model, each processor can run independently, and inter-node communication on one motherboard is not needed. With the DaVis program, it incorporates both models; first each system is run independently in a master-slave configuration, and second, the program runs parallel with the available cores on a single system. This parallel processing can consume valuable time for the inter-processor communication. To recover this time, each processor runs a separate instance of DaVis. However, LaVision software (DaVis)

Figure 1. The initial system design for the TRSPIV system. The TRSPIV system acquisition computer is linked directly to the the in-lab Cluster.

---

1The size of a rack unit with 1U being the smallest possible size.
ision limits only one instance per system. Thus by using separate mother boards, we can eliminate the communication overhead by having two completely separate systems. In addition, the double mother board configuration provides more operational computers without increasing the physical space or weight of the system.

In the original design [?] we estimated that a total of seven chassis can satisfy the power constraint, but the final configuration has only six chassis. This is due to the power needed for the master/storage node. A single chassis had to be removed to give the needed power for the master node. The configuration for the master/storage node is listed in table 2.

Beyond the nodes for the system, two additional hardware pieces were required. First, to connect all the computers together, a 24-port Gigabit Ethernet switch was used. Second, to increase the portability of the cluster, a shock-mounted traveling case was used in place of a standard computer rack case. This increased both the portability and durability of the system. Pictures of the final delivered system are shown in Figure 2.

### Speedup Benchmark

To test the cluster, two different benchmarking criteria were used. Each criteria used the same data set consisting of 1024 PIV image pairs. The difference in the criteria is the manner in how the images are processed. To test the full computational power of the system, a computationally-expensive processing job was used (see section 1 for details). Second, a job that more closely matched our own PIV implementation, called PPIV, was chosen to allow comparison of the commercial code produced by LaVision and PPIV. This job is called “Cheap” because it is computationally less expensive than the previous job. See section 1 for details.

### Expensive Benchmark

The computationally expensive setup consists of six different passes over the images sets. The first two passes use a $64 \times 64$ pixel oblong vertically stretched interrogation window with a ratio of $2:1$ and a 50% overlap. The stretching of the interrogation window is used to increase the accuracy of the PIV solution. 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 $32 \times 32$ pixels for the third and forth pass, and $16 \times 16$ 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.

Using one of the PIV processing computers at USU (sold as part of a LaVision system), this process would require over 31 hours to complete. This processing computer consists of the

![Figure 2. The fully developed Low Power PIV Cluster.](image)
following parts: One Intel Pentium 4 CPU running 3.00 GHz, and 1 GB RAM.

Cheap Benchmark

The computationally-cheap job consisted 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 x 64 interrogation-window size. The second and third pass use 32 x 32 pixels, and 16 x 16 pixels, respectively. This setup also uses the same median filter between each pass as was used in the expensive setup (section 1). With the current processing computer at USU, this process would require approximately 2.5 hours.

Benchmark Results

Running DaVis and PPIV on the cluster with the benchmark jobs documented in sections 1 and 1, we get the run times shown in table 3. We see the expected time difference between the Expensive job to the Cheap job. However, the time required for the Cheap job and PPIV was expected to be similar. We found that while DaVis ran the computationally cheap job, it did not utilize the full processing power available. Rather, it would use 40% to 50% of the available computational power. For the computationally expensive job, DaVis uses 100% of the processing power. The PPIV code, on the other hand, uses 100% of the available power running roughly the same setup as the computationally cheap job. In comparing the two programs, we see that PPIV uses each processor core as a separate system for processing images, while DaVis uses separate nodes. This leads us to believe that an excessive amount of time is wasted with the processor idle waiting for inner core communication. Running an instance of DaVis per core would theoretically solve this problem. Even with this problem, we still see a considerable speedup with the cluster.

With any parallel code, there is always an optimal number of nodes on which the program should be run. This optimal point becomes visible when running the program on an increased number of nodes does not significantly reduce the processing time. It is also possible for the running time to increase with increasing number of nodes. The simplest method to find this optimal point is plotting the speedup from the benchmarks, and looking for the point at which the curves to flatten out. For this cluster, using the

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
<th># of Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case</td>
<td>1U Chassis</td>
<td>6</td>
</tr>
<tr>
<td>Motherboard</td>
<td>Intel Server Board S3000PT</td>
<td>12</td>
</tr>
<tr>
<td>Processors</td>
<td>Intel® Quad-Core Xeon® X3210 2.13 GHz - LGA775 Socket - L2 8 MB</td>
<td>12</td>
</tr>
<tr>
<td>RAM</td>
<td>1 GB ECC DDR2-667 Memory Module (Low Power) (1 GB per core)</td>
<td>48</td>
</tr>
<tr>
<td>Hard Drive</td>
<td>None</td>
<td>-</td>
</tr>
<tr>
<td>Network Card</td>
<td>Integrated 10/100/1000 Ethernet</td>
<td>24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
<th># of Parts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case</td>
<td>3U Chassis with 650w Power Supply</td>
<td>1</td>
</tr>
<tr>
<td>Motherboard</td>
<td>Intel Server Board S3000PT</td>
<td>1</td>
</tr>
<tr>
<td>Processors</td>
<td>Intel® Quad-Core Xeon® X3210 / 2.13 GHz - LGA775 Socket - L2 8 MB</td>
<td>1</td>
</tr>
<tr>
<td>RAM</td>
<td>1 GB ECC DDR2-667 Memory Modules (Low Power)</td>
<td>4</td>
</tr>
<tr>
<td>Hard Drives</td>
<td>500 GB Raid Edition SATA Hard Drive (5.0Tb Usable Storage in RAID-6)</td>
<td>12</td>
</tr>
<tr>
<td>RAID</td>
<td>3Ware 9650 16-Port SATA Raid Controller</td>
<td>1</td>
</tr>
<tr>
<td>Optical</td>
<td>Slim-line DVD-RW Drive</td>
<td>1</td>
</tr>
<tr>
<td>Network Card</td>
<td>Integrated 10/100/1000 Ethernet</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1. The Final cluster design for the Compute Nodes. Approximate Configured Gross Weight is 30 lb.

Table 2. The Final cluster design for the Master / Storage Node. Approximate Configured Gross Weight is 80 lb.
Table 3. The time values from running 1024 images with varying node numbers using DaVis and PPIV.

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>LaVision w/ Computationally Expensive Setup</th>
<th>LaVision w/ Computationally Cheap Setup</th>
<th>PPIV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12h 50m 4s</td>
<td>54m 45s</td>
<td>8m 9s</td>
</tr>
<tr>
<td>2</td>
<td>6h 8m 58s</td>
<td>27m 8s</td>
<td>5m 6s</td>
</tr>
<tr>
<td>4</td>
<td>3h 13m 35s</td>
<td>13m 24s</td>
<td>3m 44s</td>
</tr>
<tr>
<td>6</td>
<td>2h 27m 58s</td>
<td>9m 11s</td>
<td>3m 26s</td>
</tr>
<tr>
<td>8</td>
<td>1h 36m 52s</td>
<td>6m 53s</td>
<td>3m 1s</td>
</tr>
<tr>
<td>10</td>
<td>1h 19m 41s</td>
<td>6m 7s</td>
<td>2m 35s</td>
</tr>
<tr>
<td>12</td>
<td>1h 4m 53s</td>
<td>5m 12s</td>
<td>2m 24s</td>
</tr>
</tbody>
</table>

For the DaVis Cheap job, the speedup curve starts to flatten out around 10 nodes, but does still produce a sizable speedup with 12 nodes. This leads us to believe that 12 nodes is an acceptable solution. Running the code on more nodes might push the optimal solution to 13 or 14 nodes, but the speedup curve is expected to continue flattening, leaving the optimal solution is close to 12 nodes. The optimal set of 12 nodes gives a maximum speedup of 10.5 on the cluster, and a speedup of 28.7 in comparison to the current PIV processing computer at USU.

Looking at the speedup curve for the PPIV program we see the curve quickly flattening out showing the code optimized around 4 to 5 nodes. This is expected since the total run time for one node is small. With any parallel process, there is overhead that cannot be parallelized, and as such, this time will remain constant with any number of nodes. When this overhead accounts for a significant part of the total process time, it limits the speedup. We see this happening with the two DaVis sets. If we assume that both sets have the same amount of overhead, this overhead will have a greater effect on the cheap job, because the overall time is a larger part of the total. Increasing the size of the data set would most likely show a better speedup of the PPIV program. In comparing this to the current PIV system at USU, we get a total speedup of 62.5.

Power Budget

By fire code, any electrical system must not run continuously using more than 80% and should not peak above 100% of the available power. In the USU Experimental Fluid Dynamics Laboratory (EFDL), we have have several 120-V circuits with 20 amps available. This means that the continuous power consumption of the cluster cannot be more than 16 amps (1920 Watts), and that the system cannot peak above 20 amps (2400 Watts). To insure that the cluster operates within those limits, it was tested under realistic operating conditions. Single node and entire system tests were performed using the benchmark data set described in section 1 running the computationally expensive setup.

Compute Node Load Test

Since there are two nodes in each chassis, it was not possible to test just one node. Therefore, the whole chassis was tested, and we assumed that half the power is drawn by each node. To test one chassis, it was plugged into a separate circuit and the power was measured using a power analyzer. The system tested consisted of:

- 2 Quad-Core Intel® Xeon® X3210 processors
- 8 Gigabytes RAM
- 4 Gigabit ports
- No Hard Drive

The system was tested from power up through the PIV process. The results showed that a single node has the following power consumption:
131 Watts peak power consumption during boot up.
75.5 Watts power consumption while idle.
143 Watts peak power consumption during the benchmark run.

Cluster Load Test
Since the cluster has two power strips, the meter could not be used to measure the power. Instead 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 120 volts. Just like the single node load test, the cluster was tested from power up through the PIV process and yielded the following results:

16.0 amps (1920 Watts) peak during boot up
  The boot up process consisted of booting up the master node first. Then each node was powered up in 2 second intervals
14.5 amps (1740 Watts) while idle
16.9 amps (2028 Watts) peak during benchmark run.
  during the benchmark run the current fluctuated between 15.4 amps and 16.9 amps, with an average current below 16 amps.

We conclude based on these results that the cluster will run on a single, 20-amp circuit.
For the occasion when the cluster is to be used with only a 15 amp circuit is available, we continued the test by shutting down one node at a time until the system was at 12 amp (80% of a 15 amp circuit) continuous. This was achieved after 3 nodes were shut down. For a 15 amp circuit, the cluster should be run with 9 nodes in addition to the master node.

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 light weight for portability. To minimize the cluster’s power and cooling requirements, we limited the power to one, 120 volts circuit running at 16 amps continuous and 20 amps peak. This is based on fire code standards for a 120 volt and 20 amp circuit. The final cluster design was based on the Intel® Quad-Core Xeon® X3210 low-power processor. This hardware setup allows for a total system consisting of twelve processing nodes with a total of 48 CPU cores, one master node with 5 TB of storage, and a Gigabit Ethernet connection. The total configuration along with the case has an
approximate gross weight of 390 lbs.

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 load test was performed. The load test shows it would run at the required power levels with a maximum power of 16.9 amps (2028 watts).

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 max speedup of 12 on the cluster and a speedup of 28.7 compared to the previous PIV system at USU. 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.

Acknowledgment

This work was supported in part by the National Science Foundation under award CBET-0521621. Any Opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

References