Digiliant and Open-E Solid State NAS Clusters Lee County Electric Cooperative

A Discourse on Performance Improvement and Enterprise Capability at an Affordable Price Point for Small to Medium Businesses

Zacchary Deems, LCEC Sr. System Engineer

Contents

SUMMARY

Background

Lee County Electric Cooperative (LCEC) is a nonprofit utility and one of the largest cooperatives in the United States with nearly 200,000 customers and more than 8,000 miles of energized line. LCEC is headquartered in North Fort Myers, Florida and responsible for distributing (not generating) reliable, cost-competitive electricity to customers throughout a five-county service territory in Southwest Florida, working varied schedules to ensure customers' energy needs are met around the clock. LCEC's values are deep-rooted in the community and focus on safety, quality customer service, integrity, diversity, respect, teamwork, accountability, and stewardship.

A little more than five years ago, the IT department implemented a new version of Oracle Utilities Customer Care & Billing, an ERP system designed specifically for utilities. The team took the opportunity to migrate from PA-RISC based HP servers running HP-UX to Xeon based HP ProLiant servers running Red Hat Linux. Our storage solution on the database tier at the time was an HP EVA 8100 SAN, connected via 4GB FC.

With this change, overall system performance was better than with the HP-UX platform, but there were still significant periods of poor performance when particular batch jobs kicked off, or poorly formed SQL requests were handed to the database from the applications tier. The issue presented itself as an extremely high process load on the operating system side, which at first look seemed to demand more processors. But under the hood, the team consistently found extremely high I/O wait times for database processes. Given budgetary resources at the time, LCEC managed with the infrastructure in place and made the best of the situation until it was time to upgrade once again.

Fast forward three years. Zacchary D. Deems, LCEC Sr. System Engineer, was sitting in the office of the Head of Infrastructure discussing the ongoing performance issues and brainstorming possible steps that could be taken to alleviate the constant performance issues.

Question

Given what we know about current storage technology, what is the fastest disk array configuration you can conceive of right now ignoring cost restrictions? Granted, it would have to be 'Enterprise Class', so it must be fully redundant, capable of withstanding the loss of one or more disks, and ideally it would it would meet all our Disaster Recovery (DR) requirements.

Answer

The solution would be an array comprised exclusively of Solid State disks (SSD) in a RAID 10 configuration with at least one hot spare. The standard connectivity for enterprise class applications and databases had been fiber channel for quite some time and iSCSI was only under investigation. But since this was brainstorming, they kept their options open and investigated both 10GB iSCSI and 8GB FC (8GB was the fastest fiber channel switch available at that time, and even then they did not yet have any FC switches capable of that speed).

The team was skeptical that such a configuration was within their budgetary limits, they moved forward with the hope of finding something to meet their performance and DR requirements.

Solution

SSDs were a pretty new concept at the time, so LCEC contacted Digiliant, a trusted system provider that specializes in Network Attached Storage (NAS) and iSCSI solutions. Digiliant is a Michigan-based company that was founded on more than 18 years of experience of computer hardware support and custom server designs to provide Network Storage Solutions so we called our representative in sales to determine our options and describe what we were trying to do. They are in the business of building systems and they will review specs and make sure they match the application and business needs and that it can perform the way we expect it to. Many other vendors will just give you what you asked for. They make sure we get what we need. Digiliant examined the prospective configuration the LCEC team put together, the connectivity options, and was able to satisfy all the requirements of our proposed new infrastructure with their systems. We knew we would want a pair of systems, for DR purposes, however weren't sure whether 10GB Ethernet, or 8GB FC would best suit our needs, so we reached out to Open-E who provide data storage software (DSS) used for building and managing centralized data storage servers - NAS and SAN - for expert advice.

Open-E's system experts were able to show the team that FC targets at that time could not fail over from one device to the next; however this was a feature of iSCSI targets. They also showed how Open-E DSS 7 storage software provides the option of using Active-Active clustering for High Availability (HA) and Disaster Recovery (DR) which would be critical given that this storage solution would provide disk storage to the most important database.

Result

The result of taking this step to "think big" and get expert advice was that LCEC tested and implemented a Digiliant/Open-E Active-Active SSD cluster over 10GB Ethernet.

> *"The most amazing aspect of the solution was the price for all that performance with enterprise operations and DR features. Digiliant and Open-E made us reconsider our default storage platform for databases with excellent results. The difference in performance has resulted in end-users requesting the use of the Digiliant/Open-E environment to the exclusion of the others, stating that the other environments are much slower.*
> *We saw an almost 45x improvement on sequential block reads, a batch jobs that had to run during the day and took 4 ½ hours now runs in less than 20 minutes with no user impact, and another monthly batch job that ran for 18 hours over night now completes in 27 minutes.*
>
> *Open-E support has been excellent. They jumped in right away to work with us before there was ever a purchase order. Digiliant has also been a big help. They are in the business of building systems and they review specs and make sure they match the application and business needs and that the system can perform the way we expect it to. Many other vendors will just give you just what you asked for. They make sure we get what we need.*
> *I am extremely satisfied with both parties and their performance."*

Zacchary D. Deems, LCEC Sr. System Engineer, describes the implementation and testing:

ACTIVE-ACTIVE CLUSTERING IN THE ENTERPRISE

Given that we were still in the early stages of implementing an iSCSI network outside of development tier environments, we simply were not sure what to expect from clustering two Digiliant/Open-E NAS systems. We had previously done some non-clustered cross replication between two environments, but not on the level we anticipated here, and certainly not database type traffic.

The ideal configuration for an Active-Active cluster is for the NAS systems to sit next to each other and have the 'replication network' be a 10GB Ethernet crossover cable. That worked beautifully on our TEST cluster, but in production we wanted our second NAS to be geographically separated, for DR purposes. Also for DR purposes, we wanted our redundant network interfaces to be on separate switches, so we built out our switch infrastructure in both buildings to have fully redundant paths. The replication network has to live on one of the two switches, but is configured for both, so if one goes down; it will flip to the other. This does mean that we are consuming iSCSI bandwidth for replication traffic, but as we have since found using MRTG (Multi Router Traffic Grapher, the actual load is minimal.

As far as DR planning goes, the Open-E Active-Active design meets or exceeds all of our goals. In this design, each desired LUN is split in two, with half of the desired size on NAS1, the other half on NAS2.

Replication targets of exactly the same size are then created on each NAS. When the cluster is set up, the source LUN on NAS 1 is matched with its replication node on NAS2 by creating a replication task. This way, if NAS1 experiences a failure, NAS2 activates the local replication node, and the iSCSI client is none the wiser. The Open-E Data Storage Software sees the same SCSI ID and believes it to be the same exact disk.

iSCSI discovery requires scanning both NAS systems over both network interfaces. This results in the creation of local disk devices for each of the primary LUNs. These are then grouped together under a local volume group, effectively spanning the two NAS systems under one local device. This provides fault tolerance and grants the added benefit of spreading I/O across two physically separate data stores with physically separate bandwidth. This alone improves database I/O over our previous SAN configuration without factoring in the huge IOPs gain from using Solid State Disks.

The bottom line for our organization: the performance, breadth of features that come with the Open-E DSS operating environment, and the cost effectiveness of this Digiliant solution have changed the way we think about Enterprise storage.

OUR RESULTS

I touched on our previous infrastructure design, and in the nature of full disclosure will detail the differences between the old environment and the new one.

Our old server was a four (4) core server with 32GB of memory. Its SAN storage was formatted with XFS file systems under RHEL5. I/O scheduler was set to deadline. (ext4 was not mature enough under RHEL5 for our tastes, and XFS provided by far the best performance of the available RHEL5 file systems at that time).

Our SAN connectivity was 4GB FC, dual single port FC HBAs on the server with four controllers on the EVA, for a total of eight (8) paths to each LUN, managed by device-mapper-multipath.

Our new server is a dual quad (2x4) core ProLiant (eight (8) cores total) with 48GB RAM. It has two dual port 10GB Ethernet adapters with one (1) each of their ports connected to separate 10GB Ethernet switches. Both NAS systems are on both subnets, providing a total of two (2) paths to each LUN on each NAS. We performed quite a few tests and determined that ext4 gave the best overall performance, so all of the new file systems are formatted ext4. This host is also configured to use the deadline I/O scheduler.

There are likely more opportunities for us to tweak our performance, as we did not have much time to tweak before our project went live. Still, I will provide some bonnie++ results from one of the old servers and one of the new servers. The change in file system will be evident in some areas, as XFS still does some things better than ext4, but the real evidence is in how our production systems perform.

Our batch processes, in general, ran fairly well on our old system. Most ran in 15 minutes or less, with our so-called 'background processes' running every 15 or 30 minutes. However, we had various jobs which always ran in excess of two hours, and often our nightly batch would run into the next day.

One job in particular was responsible for much of our daily poor performance, a job called 'MUP2'. On our old system, this would typically run for four and a half hours, and it had to run during the day. This job now runs in less than 20 minutes, and there is absolutely no user impact due to the job.

> We run one job called "CMDEPEXT" once a month.
> Prior to the upgrade, this would run for over **18 hours**. This job now finishes in **27 minutes**.
> Our nightly billing run previously ran for seven and a half hours.
> This has been cut in half, finishing in two and a half hours now.

System configuration details were provided in an effort to demonstrate that multiple changes contributed to our performance improvements. Doubling CPU cores and connection speeds definitely contribute to improved system performance. Upgrading to RHEL6 played a part. Not sharing the storage with any other apps or databases played a part. The big difference is that those four new cores are no longer waiting on I/O.

We have two of these Digiliant/Open-E Active-Active clusters. One is in our UAT environment, and one is in production. We also have three TEST instances of this database and application. Only one of them is using the Digiliant/Open-E NAS. The others are connected via 4GB FC to an EVA 4400. The difference in performance has resulted in end-users requesting the use of the UAT environment to the exclusion of the others, stating that the non-UAT environments are much slower. The databases live on exactly the same hardware configuration. The applications are VMWare clones of each other. The database servers are configured at the OS level in exactly the same way. The only difference is the storage system.

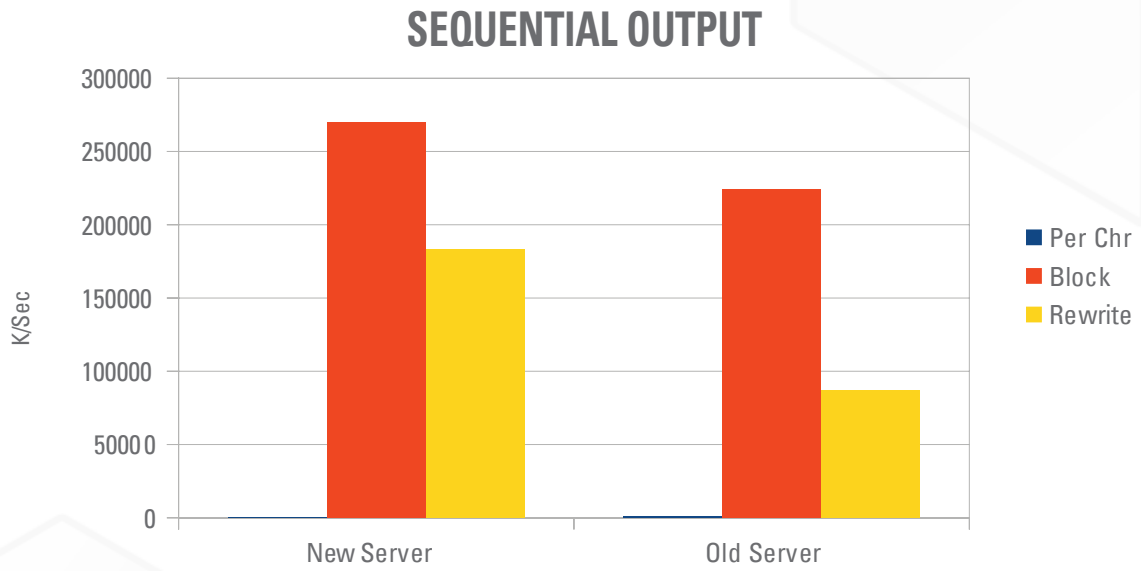
BENCHMARKS

All of the graphs which follow reference two servers, 'New Server' and 'Old Server'.
'Old Server' is the one described above, connected via 4GB Fiber Channel, using XFS for its file system.
'New Server' is the one described above, connected via iSCSI over dual 10GB Ethernet, using ext4 as its file system.

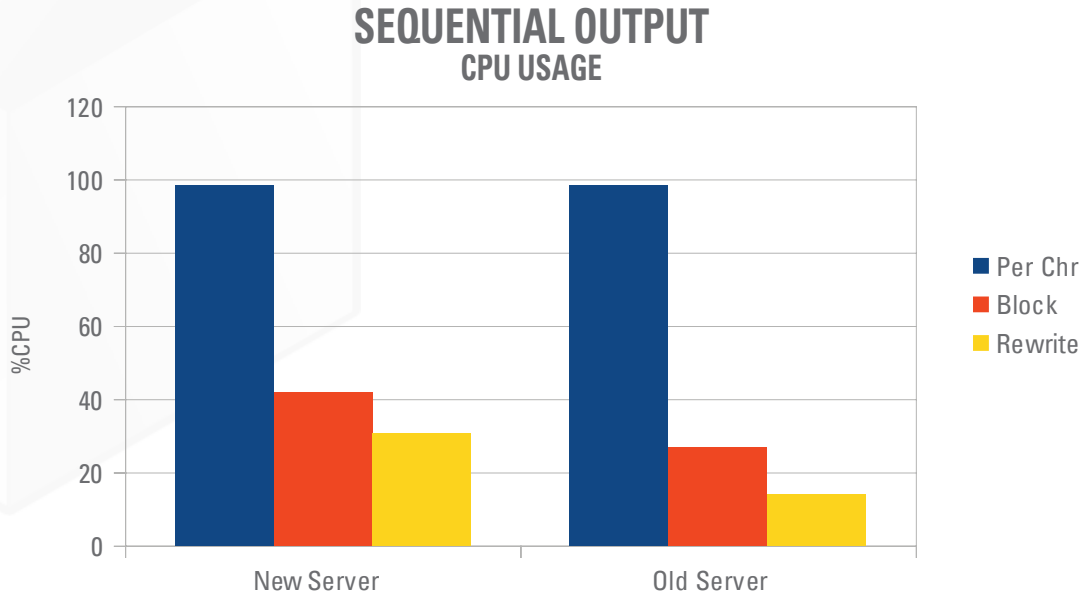
At the time these benchmarks were taken, 'Old Server' was idle and 'New Server' had a single Oracle database running, but was not under load.

SEQUENTIAL OUTPUT

As the graph below shows, 'New Server' performed better on block writes, and significantly better on rewrites. It actually performed half as well on per character writes, scoring 650 K write per second compared to 1217 for 'Old Server'.

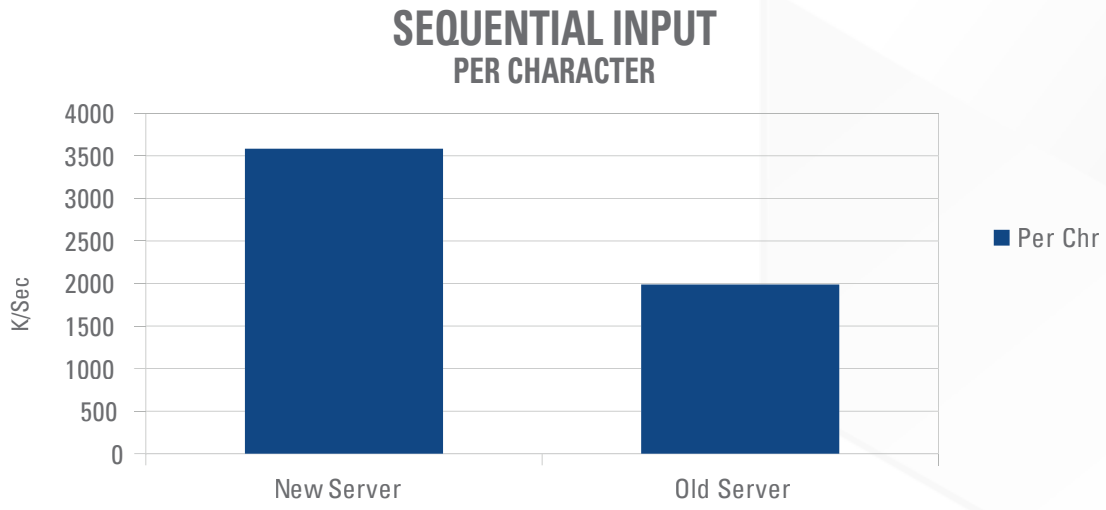


This next graph demonstrates the costs associated with the Sequential Output tests. Both servers experienced a 99% CPU usage total while performing the 'per character' output test. The other two show increases for 'New Server' somewhat relative to the increase in performance. Again, some portion of this statistic is related to the file system change, but the exact amount is unknown, as is the actual effect.

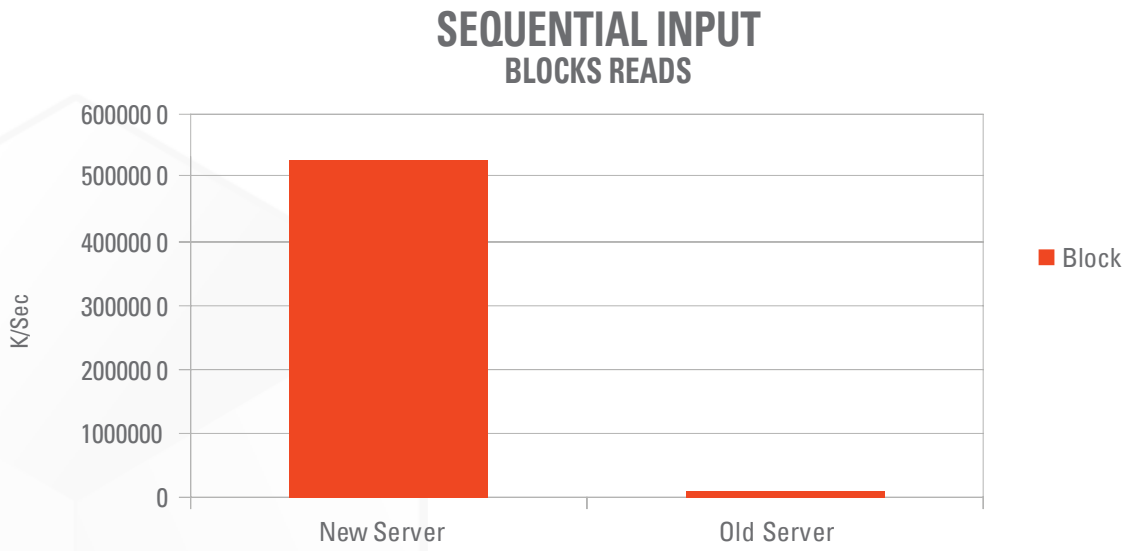


SEQUENTIAL INPUT

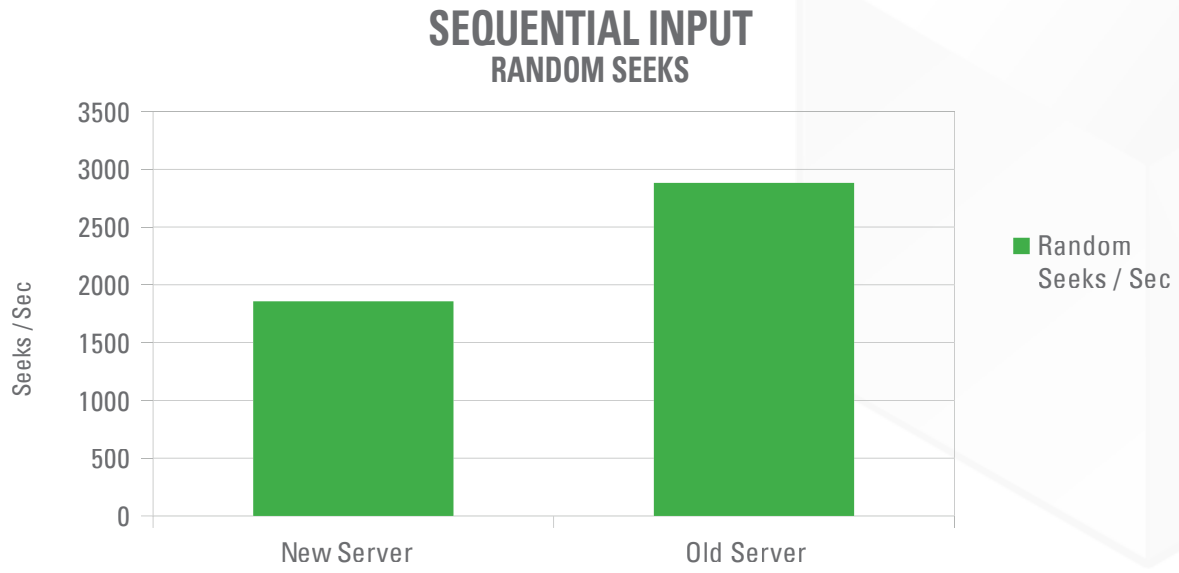
The next few metrics will be shown individually in order to properly reflect the differences.



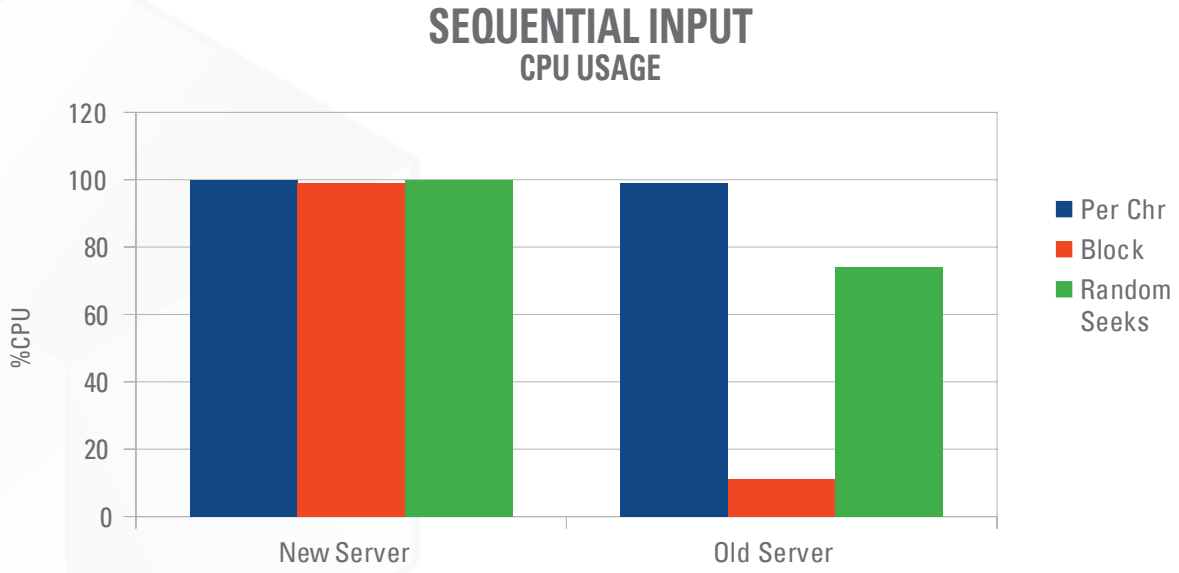
The real 'wins' with this storage appear on the 'read' side. As the Per Character graph shows, the new server better than doubled the performance of the old server. The next graph is even more meaningful:



The performance improvement on block reads is through the roof.
The SAN based server scored 118,934 K/Sec, compared to the Active-Active cluster's stellar 5,269,393 K/Sec.
The Random Seeks test actually proved out in favor of the old server, as shown below:



We see significant gains in some areas and moderate reductions in others.
This graph shows what the gains cost us in CPU utilization.



So we either have some very expensive input, or the server was extremely idle while the test was being performed on 'New Server'. If it indeed is that expensive, we may well have noticed the difference on a similarly configured 4 core machine, but the new 8 core server barely notices it. The fact remains, though, that we did see almost a 45x improvement on sequential block reads.
Latency values are not supplied due to the fact it would be impossible to determine within our current infrastructure whether the resulting latencies are intrinsic to the storage solution, the associated network fabric, or pressure caused by other systems sharing the fabric.

(All benchmarks were performed using bonnie++ 1.96 and 1.97 under 64bit Red Hat Linux.)

What Open-E DSS V7 with Active-Active iSCSI Failover has to offer?

- It is simple to use – while Active-Active set up is considered as complex to configure - with Open-E DSS V7 it is no longer an issue.
- Eliminates wasting of hardware, all resources are in use - working towards better system performance.
- Provides self-validation of the system. When starting a cluster, Open-E DSS V7 checks all the critical settings on each node. This way, clusters will not be started if they were configured wrong.
- Increases sensibility for network failures, thanks to the possibility of configuring Ping Nodes.
- Speeds up networking connectivity, since I/O traffic is equally balanced on two nodes.
- Fully utilizes all processing power on both cluster nodes.