How Will NonStop Fit Into the Internet of Things?

Justin Simonds >> Master Technologist >> Americas

Dean Malone >> NonStop Architect >> Caleb Enterprises Ltd.

PART II – New NonStop Architecture Fundamentals

In Part I we discussed the IoT (Internet of Things) market with some general examples in the automotive and energy markets. We discussed how this expanding market is similar to OLTP and how data stream processing requirements were a good fit for NonStop. In this section we'll discuss some ideas for enhancing NonStop based on the x86/InfiniBand announcement that was made during the 2013 Boot Camp.

What InfiniBand Brings to the Party

OLTP transactions typically have an all-or-none nature that is specifically tied to a database. IoT data will not involve a rigid definition of transactions, but will require fault-tolerance – the ability to survive any single point of failure. With storage systems implemented using flash memory (see www.hp.com/3PAR and NonStop's use of SSD) the simple truth is that accessing secondary storage across a fiber network implemented with InfiniBand (IB) will be no slower than connecting to RDMA (remote direct memory access) but there is definitely a difference in cost. Memory is cheaper and prices will continue to fall. Also, on any given server, the memory residing in local DMA is a few orders of magnitude faster to access than storage arrays because of context switching. How much faster? Latency for RDMA over IB is about 250 µs (billionths of a second) versus about 85 µs for local DMA or just a couple of µs (billionths of a second) for cache. Intel reports that a Linux context switch on an i5 core is about 3000 µs (see blog.tsunanet.net/2010/11/how-long-does-it-take-to-make-context.html); and so applications will be optimized to house the memory in the server that will access it most frequently.

The key area of opportunity that supports both MPP (Massively Parallel Processing) and fault tolerance that very few – if any - products have leveraged to date is the ability to leverage RDMA (Remote Direct Memory Access). Any server that implements an InfiniBand HCA can be engineered to access RDMA but only NonStop is engineered to provide a fault-tolerant framework for RDMA. But is there a need for such a capability? Let's dig deeper.

HP Nonstop has been the designated database and hub-system platform of choice for mission-critical core processing architectures in many Fortune 500 companies. The reasons customers select the HP NonStop platform are usually three-fold; mixed-workload processing, scalability and high availability concerns.

The basic premise of NonStop is continuous application availability. Availability, as defined by NonStop, is more than just up-time; it presumes applications performing at acceptable service levels with appropriate response times. NonStop systems are in environments where on-line transaction processing capabilities, response time, security and accuracy are paramount. The NonStop hub may provide stand-in processing at times when the back-end ERP and legacy systems may be experiencing an outage or planned downtime. HP's own IT call centers and websites (HP eats its own cooking) interrogate the NonStop enterprise data store (iHub) to determine order information, ship dates and whether or not a cross-sell opportunity exists for the current customer. These opportunities involve a different style of processing which requires a robust mixed-workload capability (i.e. consistent response times to users and applications while processing batch jobs and large queries), which is a known capability of HP NonStop. When future processing requirements are uncertain, the MPP design allows scalability in a graceful and predictable manner. Adding processors adds additional capability up to a theoretical limit of 4,080 logical processors. Currently with the Quad-core Integrity Blade (Itanium and x86) systems, this is a physical limit of 16,320 cores. These, of course, are known capabilities of NonStop and they fit extremely well with the requirements we are seeing with IoT and stream processing systems. Let's take a look at what's on the horizon.

Why Latency is Important

While NonStop has long been recognized as the database and hub-system platform of choice for mission-critical core processing architectures in many Fortune 500 companies, scalability and fault tolerance has come at the expense of price and performance compared to SMP architecture. There is something very intriguing about IB and how it relates to secondary storage, shared-memory and transaction coordination in a distributed computing environment that will have a profound impact on future application architectures. It all boils down to throughput capabilities and how they are expanding logarithmically; so as to actually invalidate past architectures. Competent systems architects are aware of the following basic rules of thumb about I/O latency:

Operation	Latency (µs)
Disk	7500000
Disk with RAID 10	15000000
XP 20000 Storage array (worse case)	9000
Fiber SAN	5000
Linux/Intel E5-2620 Context switch	3000
10 Gb Ethernet	1200
HP StorServ 7450 (3PAR flash storage)	700
InfiniBand	250
RAM	83
CPU Cache	3

Figure 1 - Relative latency of computing infrastructure components

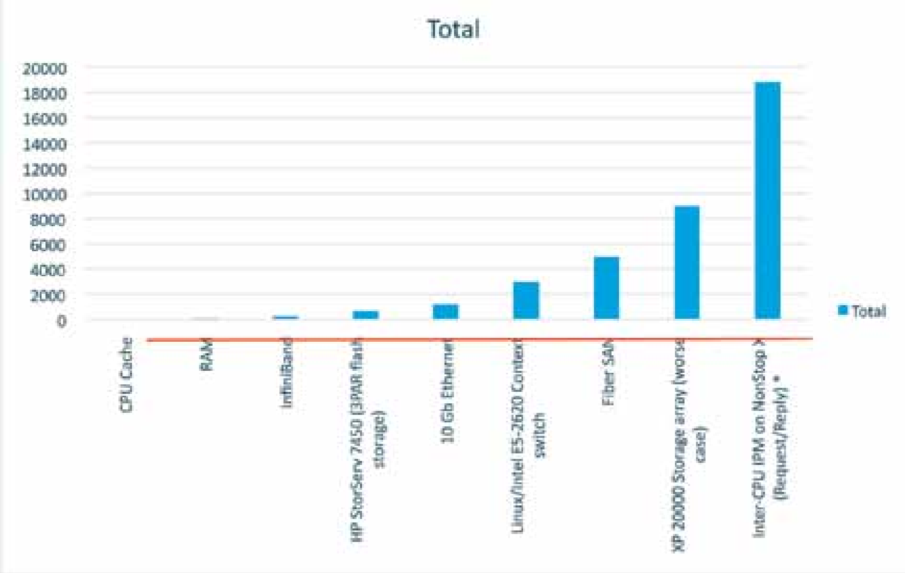

Figure 2 - Relative latencies (µs)

Latency is the critical metric to consider when it comes to performance because this is how long a task must wait before starting to process what has been sent or retrieved. The following graph puts all of this into perspective in nanoseconds with *typical* latencies for each operation of interest. We could find no metrics for the cost of a context switch on NonStop (although we did observe that a Guardian IPM request/reply across processors on a NonStop X server took 18,800 µs) so we included the cost of an Intel i5 on Linux as a reasonable approximation. All the values represented were found on various web sites and are only represented as reasonable approximations:

Platform	Writes/ Sec	Updates/ Sec	Reads/ Sec	Deletes/ Sec
HP Model 30 NIKE RAID 1 raw file access across a single mirrored drive with no indexing	246	159	116	159
Synchronics 1000 with RAID 1 Solid State	61	61	72	61
Synchronics 1000 with RAID 1 conventional disk drives	44	44	45	44
Informix database access on our K-460 with HP model 20 NIKE storage array.	16	50	200	113

Figure 3 - SQL I/O rates versus other file systems

Conventional disk I/O is such a huge drag that we had to factor it out to get a clear sense of the relative latency of the other elements.

Bandwidth and channel overhead become the next critical component to measure. We are ignoring this for now because ultimately, all devices can be engineered to transfer at the IB channel rate. With InfiniBand, all of these interfaces can be leveled to 250 µs latency (i.e. the red line of Figure 2) with a theoretical sustained aggregate transfer throughput of 100 Gb/sec on the latest (e.g. Mellanox SB7790 100 Gb FDR InfiniBand switches, 3PAR solid state storage arrays from HP that are connected as InfiniBand TCA devices, etc.) available technologies. This means that a process can directly access the memory of a remote server at this incredible I/O rate (i.e. 12.5 billion bytes per second) and it can retrieve data from secondary storage at this same rate. Computer processors typically can't keep up with sustained throughput like this today. Processors and their operating systems are the new bottleneck.

From Partitioned Disks to Partitioned Memory

How might new system architectures take advantage of this? If you are building a shared memory solution, you will want the shared memory to reside on the server that will access and update it most frequently to leverage extra 170 µs less latency (i.e. DMA versus RDMA) but you will still provide RDMA access at µs (billionths of a second) speeds in the same order of magnitude so that the data can now realistically remain at rest for remote accessing processes too. That is a key premise behind The Machine (www.hpl.hp.com/research/systems-research/themachine/). If you are doing a database access, the whole access path can be radically accelerated with the use of IB; but there is still considerable processing overhead involved in executing a SQL query. Here is an admittedly dated (i.e. over a decade old) comparison of SQL versus other forms of I/O that one of the authors actually measured to illustrate the point:

Ideally architectures should only do database operations when needed and should wherever possible, accelerate applications I/O through the use of shared memory. Shared memory can now be distributed across an IB fiber network.

Here is another thought to ponder. We stated at the outset that IoT is "OLTP-like" but there are some interesting differences. When there is a deluge of digital and analog data, it is unlikely that businesses will wish to save all of it to permanent storage. It may well only be interested in aggregating 'normal' data or in the values that are outside the expected mean. That said, there will likely be a requirement to save the exception data and in fact, it may need to be fault tolerant. If this is the case, it will not be desirable to have a transaction monitor involved in the update because that will be a drag on performance. How will conventional architectures be able to ensure updates while surviving any single point of failure? Only NonStop is presently engineered to meet this need. How? By delegating such operations to a NonStop process pair whose primary holds the critical data and checkpoints changes to the backup process residing in another CPU. If the primary fails, the backup will elegantly and seamlessly provide the single-point-of-failure recovery that only a NonStop can achieve. That is entirely the point behind IDC's AL4 capabilities (IDC report on availability levels – 4 being the highest).

Are the Days of Map/Reduce Architectures Numbered?

Today cloud-based systems are utilizing various flavors of map/reduce technology (e.g. Hadoop, PIG, Simple Messaging Service, etc.) by replicating data across multiple servers and coordinating their updates with sophisticated monitoring frameworks. However there is very little to date that has been built which meets or exceeds IB aggregate throughput. If you consider, for example, the ZooKeeper framework for managing semaphores in a distributed computing environment, a given semaphore's performance deteriorates very rapidly if the rate of updates of all semaphores in aggregate exceeds 10% of the overall pool of resources (see zookeeper.apache.org/doc/current/zookeeperOver.html#Performance); not particularly scalable. If instead, that semaphore resides in a particular processor that can be accessed across an IB fabric, billions of operations per second are theoretically possible with no need to replicate (replication with map/reduce being the cloud methodology for achieving scalability and resilience).

New Core Capability for NonStop

What this really means is that shared-nothing multiprocessing – what is also generally known as MPP – architecture will soon eclipse symmetrical multiprocessor (SMP) architectures that are currently in vogue. Why? Because SMP is constrained by Von Neumann architectures (sequential processing see en.wikipedia.org/wiki/Von_Neumann_architecture) whereas MPP architectures deal with parallel processing naturally by providing the kinds of synchronization and failover mechanisms we take for granted on NonStop - but that are lacking in other operating systems. To date, NonStop has been at a disadvantage to the SMP shared-memory applications of competing platforms primarily owing to the high-latency IPM requirements mandated by the shared-nothing environment. With IB and the natural complimentary semantics it shares with WRITEREADX, READUPDATEX, AWAITIOX, etc. the disadvantage is about to be turned to advantage. The requirements of the new order include parallelism, scale and fault tolerance which will be combined with the speed advantages of IB. SMP-only systems are already hitting a wall. The increasing number of cores already have incredibly complex programming to optimize the use of threads. When core counts start reaching 64, 128, 256 and beyond; threading becomes untenable. MPP is a far more scalable architecture, as everyone will eventually come to appreciate. Emerging exascale standards such as MVAPICH2 (mvapich.cse.ohio-state.edu/overview/) are predicated on it.

NonStop has many of the new requirements embedded such as scalability, reliability, security and of course fault tolerance. It was mentioned earlier that not all elements of the new stream processing applications need to be fault tolerant. In this second instalment, we have demonstrated how NonStop is uniquely positioned to meet the fault tolerance challenge with the correct parallel processing architecture to meet the rigorous demands of the most demanding indestructible computing environments at the extreme velocities Web 3.0 is expected to bring – all over the radically increased velocity made possible by InfiniBand networks. What about hybrid, converged architectures? How can HP's NonStop with these new IB capabilities participate in a hybrid architecture to leverage the economies of lower-priced platforms? We'll explore that in Part III.

SIDEBAR FOR NONSTOP FUNDAMENTALS

Massively Parallel Processing versus Symmetrical Multi-Processor

Massively Parallel Processing (MPP) is a system architecture that presumes each processor has its own RAM and that other processors cannot access it. Each processor is presumed to operate as an autonomous system that has mechanisms for coordinating work with other processors – typically a message bus. Symmetrical Multi-processor (SMP) is a system architecture that presumes two or more processors access the same shared RAM. To do so, synchronization mechanisms (i.e. typically semaphores) are used to coordinate access between processes running in competing CPUs.

Single Point of Failure

A single point of failure is any hardware or software component that should it fail, will bring down the entire system. Such single points of failure typically include CPU, RAM, controller, bus, critical device driver, LAN, communication line, power supply, etc.

Checkpoint

A checkpoint is a NonStop-specific term relevant only to NonStop Process Pairs. The purpose of a checkpoint is to ensure that the two processes have identical process state so that if there is a single point of failure, the backup process can take over the completion of processing without the requesting process needing to do any exception handling. The purpose of checkpoints are to make primary process failures transparent. Checkpoints occur on critical region boundaries to ensure all-or-nothing processing process semantics. The primary process sends checkpoints on critical region boundaries to the backup and waits for acknowledgement before proceeding to the next critical region step. It is up to application architects to determine what the critical regions of a process are.

Multi-core Processors

Multi-core processors are a more recent evolution of CPU architectures whereby sophisticated chip logic and compilers allow processing to be broken into threads of critical regions and submitted to multiple cores simultaneously. Instead of a processor needing to run at ever faster clock speeds, cross-section computing power can be aggregated across multiple processors to achieve the same effect.

NonStop Process Pair

A NonStop process pair refers to a primary process residing in a particular CPU and a designated backup process residing in another CPU. They share the same process name but each have a different CPU:PIN (i.e. process id) pair. They can be configured to be amnesia backup processes (i.e. they know nothing about the state of the other process) or they communicate with each other to preserve state using checkpoints.

IPM

An IPM (Inter-process Message) is a type of message that is sent across a message bus to tie the processors of an MPP system together. In the context of NonStop, this is a message that is sent between any two processes within a given NonStop node or across an EXPAND network of NonStop nodes. These messages are unsolicited requests that are sent to the $RECEIVE message queue of a specified process by another NonStop process. On NonStop X, InfiniBand is the bus fabric.

Transaction Monitor (TMF)

A Transaction Monitor is any daemon or mechanism that can ensure ACID properties of a given database update or transactional all-or-none execution consistency respectively. On NonStop systems, the transaction monitoring is achieved with the Transaction Monitoring Facility (TMF) subsystem. It is fully integrated with the file system and IPM.

RDMA

RDMA (Remote Direct Memory Access) is a capability specific to ServerNet and InfiniBand architectures whereby the memory of a given CPU can be directly referenced by a process in another CPU without the CPU that owns the memory being involved in the I/O operation. There is no context switching, interrupt or trap handling needed to service the I/O. Everything happens in user mode during the process's execution time slice for maximum efficiency.

Justin is a Master Technologist for the Americans Enterprise Solutions and Architecture group (ESA), a member of the HP IT Transformation SWAT team, and a member of the Mainframe Modernization SWAT team. His focus is on real-time, event-driven architectures, business intelligence for major accounts and business development. Most recently he has been involved with modernization efforts, Data Center management and a real-time hub/Data Warehouse system for advanced customer analytics. He is currently involved with HP Labs on several pilot projects. He is currently working on cloud initiatives and integration architectures for improving the reliability of cloud offerings. He has written articles and whitepapers for internal publication on adaptive enterprise, TCO/ROI, availability, business intelligence, and the Converged Infrastructure. He is a featured speaker at HP's Technology Forum and at HP's Executive Briefing Center. Justin joined HP in 1982 and has been in the IT industry over 34 years.

Dean is one of the pioneers of Message Oriented Middleware (MOM), having chaired three panels on MOM in '93, '94 and '95 at COMDEX. He developed the world's first fault-tolerant shared memory (XIPC on NonStop in 1995) deployed that product as the first customer implementation of active NonStop process pair (four programs implemented) and also ported Seer HPS/NetEssential 4GL-middleware to the NonStop. His biggest middleware achievement was the porting of IBM MQ-Series to NonStop as Chief Architect in 1998. He was the infrastructure architect for the Province of Ontario responsible for implementing the world's first wireless WAN-based mobile workstations for OPP, regional police, carrier enforcement and ambulance services. His customers include banks, brokerages, retail, EFT/POS switches, funds wire, vendor products, airlines, reservation systems, industrial automation and more. He has built systems on NonStop, VMS, Stratus, Unix and PDP-11 and has played roles as architect, technical lead and hands-on technical problem solver as a consultant for over 30 years. He is presently completing an RDMA Middleware product that will implement distributed shared memory, semaphores and queue-based messaging between NonStop, Linux and Windows servers over InfiniBand.

YOUR NONSTOP AUDIENCE JUST GOT BIGGER!
Read The Digital Version of The Connection Now
The Connection
Where In The World Is My Credit Card Data?
The Connection
Managing the Media Life Cycle with HP StoreOnce & ETI-NET BackBox
http://www.connect-community.org/general/custom.asp?page=theConnectionArchive