How will NonStop fit into the Internet of Things?

PART III – Leveraging Nonstop and Large-Scale Hybrid Solutions

Justine Simonds >> Master Technologist >> Americas

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

As we've shown in Parts I and II of this series, NonStop on x86 with InfiniBand (IB) has all the foundational underpinnings required for participation in the emerging Internet of Things (IoT) processing environment. Clearly NonStop's OLTP and message-switching history make it a strong contender for new applications. The special features of NonStop: availability, scalability, security, and parallelism are known requirements for many of the applications that will be developed in the coming years. In Part I and II we discussed the benefits of InfiniBand (IB) and some IoT use cases with connected cars, smart meters and the concept of informational messages versus critical messages - such as remaining oil life versus air bags deployed. Some use cases will require the fault tolerance of a NonStop. In Part III we want to discuss hybrid systems and how a combination of utility servers and NonStop would be the perfect architecture.

Why is this important? It is most definitely true that a large percentage of applications or portions of an application can tolerate a failure as long as a failing server can be quickly replaced. It is now a standard for virtually every operating system to do this. Microsoft has MSCS (Microsoft Cluster Solution). Linux systems rely on products like VMware. Web-based systems use DHCP server clusters and front-end processors. If a signal or data point being lost does not result in serious consequences, having it reside on conventional architectures is acceptable. If it does indicate something serious - your house is burning down, a car just crashed and the air bags deployed, or your boat is taking on water – losing the signal as a result of a single point of failure takes on much greater importance. This is the sweet spot for NonStop. And memory persistence is entirely possible because NonStop supports process pairs that can keep copies of memory in sync in two CPUs. In the event of a failure, the backup takes over and it can thus survive any single point of failure. With IB, this state checkpoint capability could even be extended to a process on a remote node or even a different data site without substantial latency. With some new thinking and products from NonStop partners it would be possible to integrate NonStop closely with these other systems to create a powerful high-performance hybrid platform.

EXTENDING NONSTOP FUNDAMENTALS TO CONVERGED INFRASTRUCTURE

At Discover 2011 and 2012 NonStop demonstrated an approach to cloud and commodity server processing known as Persistent Cloud. Please see Connect March/April 2012 Volume 33, No. 2 "Persistent Cloud Computing Architecture" for a refresher. The main idea was the extension of Pathway (TS/MP) off platform to Linux, Windows, UNIX and even cloud-based systems. Pathway has provided a number of transaction middleware features. The first is persistence, which allows programs to avoid complex NonStop process-pair programming and yet have their applications overseen by Pathmon and automatically restarted if there was a failure, retaining database consistency if TMF is used. Additionally, Pathmon provides some load-balancing of transactions between the various instances of the application, known as the serverclass members. It also provides elasticity based on response time. If response time slows down, Pathmon can automatically start additional application instances to maintain response time performance. Likewise once the additional load has dissipated, Pathmon can shut down the surplus copies, thus freeing up system resources. The idea behind persistent cloud was to extend these features off-platform and to create a composite application or service where those portions that needed fast and cheap processing were run on Linux, Windows or a cloud platform. The portions that could not afford any outage would run on NonStop.

The persistent cloud was a demonstration developed by the HP Enterprise, Solutions and Architecture organization of the Americas. Seeing a market for this service, Infrasoft licensed this demonstration code and has rewritten and productized this service; see:

http://www.infrasoft.com.au/maRunga.html

It has been released as maRunga and is sold through comForte; see:

https://www.comforte.com/products/modernize/marunga/

This solution was developed over standard TCP networks. It runs well, but what if the hybrid connections could use an InfiniBand fabric rather than a network? It would create an exponential increase in speed and performance between the hybrid components in cases where data transport costs were a significant contributor. The NonStop and Linux systems might seem to be a single system, based on performance. Imagine NonStop on x86 surrounded by a number of inexpensive Linux systems, interconnected by InfiniBand and voila! We have the architecture for a killer IoT appliance.

MaRunga serves the NonStop community well by enabling large scale-out of Pathway-based systems. Consider being able to add NonStop fundamentals of high availability and massive scalability to existing SMP-based applications that could be moved to NonStop with minimal porting effort. Is that possible? We think so, but not without constructing some new tools to leverage IB shared memory capabilities over MPP.

NONSTOP AND HP'S BROADER MACHINE INITIATIVE

The question of greatest importance to NonStop customers and vendors is how will the new x86 NonStop fit into HP's stated Enterprise corporate strategy? Can it map to The Machine capabilities that Martin Fink presented in his keynote address at 2014 HP Discover?

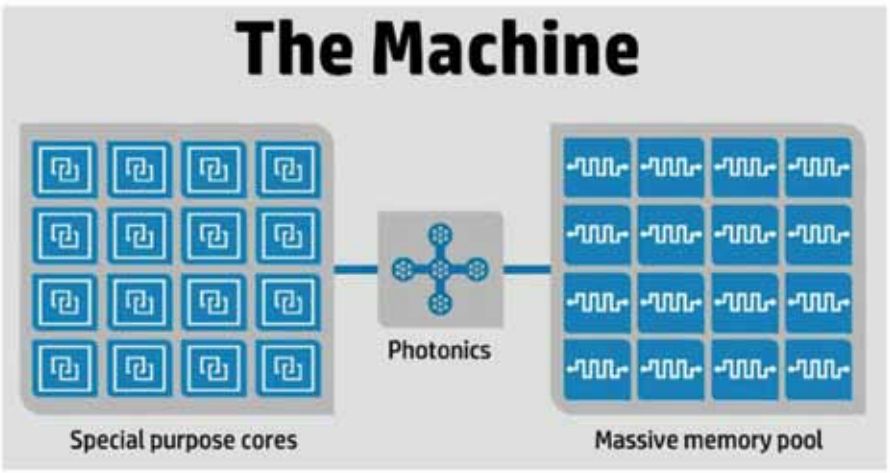

Figure 1 - HP's Ultimate Vision for its Products

This is a classic example of "a picture is worth a thousand words" but let's add a few more words anyway just to incorporate some NonStop perspective.

Electrons Compute: Let's start with ***special purpose cores.*** NonStop architecture supports from 2 to 16 logical processors – each comprised of multiple cores– to comprise a node. Each of these processors are directly connected to each other via IB. These nodes can be clustered together in an EXPAND network to support 255 inter-connected nodes, just like they always could. When IB clustering is announced a number of nodes will be inter-connected using IB so that Inter-Process Messaging (IPM) between nodes in the cluster could theoretically become just as fast as IPMs within a node.

Photons Transmit: How does IB fit into the picture? It is similar to the ***photonics*** piece in Martin's diagram and will be the fastest networking option until "The Machine" releases photonic networking. IB is the special sauce that makes all of this possible because IB is to our x86 processors and storage as photonics is to special cores and memory. NonStop uses IB as its interconnect "photon" bus.

Ions Store: Finally, how does the ***massive memory pool*** fit into the picture? Again IB provides the answer with two mechanisms; TCA (Target Channel Adaptor) for non-volatile storage and RDMA (remote direct memory access) for volatile storage whereby a process in one CPU can write to or read from the memory of a target CPU without interrupting processing on that target CPU. What this means is that context switching can be completely eliminated in the exchange of data between a process and either secondary storage or volatile memory. This means that any process residing in any one of the 4080 NonStop CPUs of a 255 node network could, in theory, scale vertically to access the memory of any other CPU. It also means, in theory, memory could be scaled horizontally by allocating memory across the RAM of up to 4080 processors to create a single universal resource pool. This does for memory what partitioning files across disks does for file I/O.

As Martin Fink said at HP Discover 2014, "But wait; it gets better!" NonStop processes can survive any single point of failure using our proven capability called NonStop process pairs, whereby a primary process residing in one CPU can checkpoint its state to a backup process residing in another CPU. This means that if memory access occurs with a remote process using IB Queue Pair (QP) messaging instead of RDMA, then that process pair can keep memory identical in two processors and the shared memory data can thus survive any single point of failure. NonStop can own a powerful and unique space within this marketplace – a massive fault-tolerant shared memory address space.

There are several mechanisms required for hybrid computing. The first is the ability to do RDMA between IB hybrid host end points. The second is a high-performance synchronization mechanism (i.e. semaphores) to coordinate shared resource access and enable light-weight publish/subscribe group notifications. A third is queue-based messaging. This is fundamentally what the IPC services API provide on UNIX platforms in support of SMP architectures – but only on a single node. In the case of OSS, this is presently only possible in a single CPU. Ideally, these resources should be accessible from any program – compiled image or containers (i.e. JVM.) What about DNS-like capability for discovery and late binding to these resources, or metrics that are distributed across the entire fabric, or fine-grained security authorization and authentication? There remains much to be built but the foundational capabilities are all there. This has not even been possible until recently.

So let's put it all together, NonStop has the potential to inter-communicate with Linux and potentially Windows servers in a hybrid computing environment at blazing speed. This brings to market a concrete product that realizes what Martin Fink said in his June 11 blog titled Accelerating The Machine when he said "Remember I said we want working prototypes as soon as possible? We can get there sooner if we use plain old DRAM as a stand-in for the perfect memory technology. No, DRAM isn't persistent, but we can emulate persistence." Now if only the capabilities of the IPC subsystem could be implemented in this hybrid computing environment; the picture would be complete! NonStop would be at the pinnacle of reliability as the only platform that could provide these resources persistently and survive any single point of failure.

Speculation and conformance to Martin Fink's Machine vision is a good thing but what sorts of things can NonStop do in the near term that are a little less visionary and more practical and actionable?

NONSTOP AS INTEGRATION HUB

NonStop is particularly good at OLTP. It is also particularly adept at managing workflow integration. A decade ago, this capability was marketed as Zero-Latency Enterprise (ZLE). Indeed, this framework was a cornerstone of HP's own business systems where it managed and integrated all of HP's product ordering and fulfillment across its many product lines. An integration hub is more than ZLE. It is workflow orchestration and transaction integration in an explicitly hybrid computing environment. It meets the high bar of "the right compute for the right workload" by being able to scale massively and reach deeply into your enterprise's data lakes – wherever and whatever they may be.

What will it be able to do? Here are a few specific examples. Assuming that a "transaction" will originate from a "client" and that it must be processed on an all-or-none basis with fault tolerance built in so that the client is isolated from failures as much as possible:

1. The client application submits its cookie information about the customer engaging with us via their web browser. If we can identify the customer, we can initiate information lookup on several fronts.
2. If we can't identify the customer, we can ask them for basic information (name, address and email) in return for a one-time discount coupon.
3. Armed with customer-specific information we can:
   a. Look them up on our customer database to see if they have ever purchased from us before.
   b. Look through the click base to assemble a list of items they have shown a past interest in but did not purchase.
   c. Assess if any past purchases have follow-up sales opportunities (e.g. vacuum bags for a vacuum cleaner, v-belts for a lawn-mower, a replacement for something that has reached expected end-of-life, etc.)
   d. Perform a lookup in our data lake for any information about them that could help us personalize the interaction. For example if they recently moved, won an award, had a birthday etc.)
   e. Perform the same lookup on the web.
   f. Assess them for potential fraud.
   g. Invite them to the nearest bricks-and-mortar store with an incentive offer if they are accessing via a mobile device.
   h. Construct a special offer that can be dynamically offered on the browser based on the past-purchases and past-clicks analysis.
   i. Look them up in our credit-bureau information database to see if they qualify for our credit card.
   j. Evaluate complementary accessories and make a one-time special offer before committing the transaction if they make a decision to buy, as a final step in the shopping experience.
   k. Store everything in the corporate data lake about the interaction for future use in the above-described work flow.
4. Much of the workflow activities identified above will be analyzed on hybrid servers that the workflow engine will initiate. The work queues should reside on NonStop where they can live through any single point of failure by using process pair checkpoints. If any downstream servers fail, and they are supported by MaRunga the workflow stream

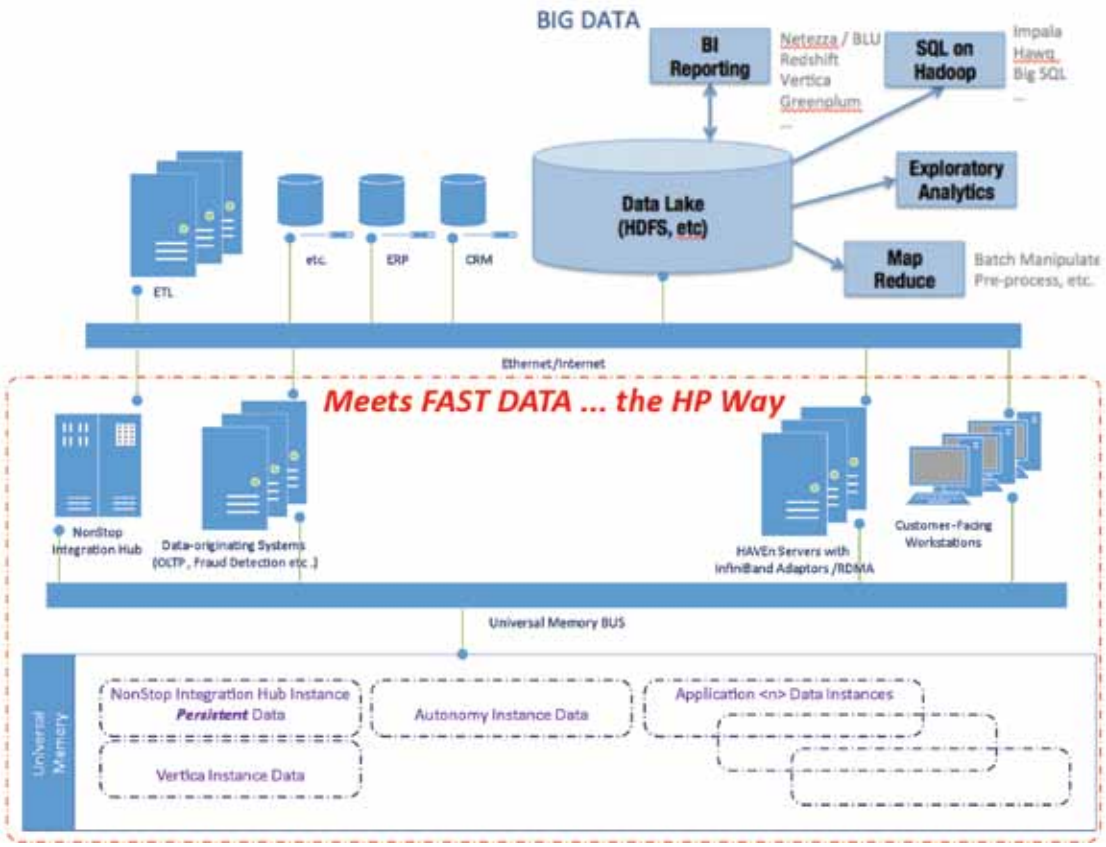

Figure 2 - Big Data meets Fast Data

will be restarted by the integration hub.

5. If the customer navigates away from our web site, we will make them a special offer to keep shopping.
6. If the customer leaves our site, we will construct a special offer to email them.
7. If the customer leaves our site, we will clean up all the dynamic memory queues related to downstream workflow items and send downstream cancellation notifications for any work that is pending.

The main driver behind all of this is the notion that data should remain at rest and access to data by workflow participant processes should be direct across a fiber-optic network. This was not even possible until the advent of InfiniBand RDMA and OFED (see www.openfabrics.org to learn more) drivers. In this new world, it becomes important to make sure memory-based operations – queue-based messages, semaphores and direct memory updates - can survive any single point of failure. These may be things that NonStop can bring to the party in the future.

What we are describing here is a specific framework for big data to intersect meaningfully with fast data using HP's HAVEn framework to yield increased revenues. The following figure illustrates what that looks like:

PUTTING IT ALL TOGETHER

In conclusion, the data deluge that the IoTs will bring provides some unique challenges and opportunities that NonStop, with its fault tolerant, MPP architecture, can take a leadership role in addressing. Key to meeting these challenges will be to incorporate fault-tolerant versions of shared memory distributed across a hybrid computing environment on an IB Fiber Network. This new hybrid computing environment will offer the massive scalability, openness, performance, and reliability that will be required in the Internet of Things era.

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.