SUSE Linux Enterprise Real Time Low Latency Fabric Roadmap

Moiz Kohari – Vice President Engineering Open Platform Solutions





OpenFabrics Enterprise Distribution

- Support and enhance OFED for SLERT
 - Fix existing drivers for SLERT environment
 - Not all OFED drivers have been vetted in RT environment(Chelsio)
 - Enhancements to eliminate long or unbounded code paths
 - QOS
 - Leverage recent OFED QOS implementation to provide differentiated services for fabric services(IP, iSER, RDMA, etc.)
 - > Provide configuration tools and procedures for fabric manager
 - > Performance manager monitoring of fabric
 - NFSoRDMA
 - > Provide configuration tools

Fabric QOS

N

End to end Prioritization

Scheduler priority

Packet priority

Reserved/prioritized hardware queues on fabric

Event/flow agility and minimization of contention

Direct classified flows as discrete events to individual CPUs without impact on rest of the system

Isolated "shielded lanes" built from queues along with the associated interrupts, CPU cores and memory nodes



Memory Channel

- A distributed shared memory system based on DEC's Memory Channel architechure.
- Similar to SysV Shared Memory Segments, but adds a 'wire' between clustered nodes.
- > Shared mem is "many processes, one node"
- > Memory Channel is "many processes, many nodes"
- Built on SLERT SP2 for low latency.
- Bidirectional Transports include Ethernet, iWARP, and Infiniband
- Requested by Credit Suisse.

Low Latency Stack

- Netchannels Stack
 - Eliminates bottlenecks encountered in the current stack.
 Targets these bottlenecks:
 - > Packet copies
 - > Context switches
 - > Syscall overhead
 - > Lock contention
 - Lockless queuing based on IP flow
 - Protocol processing performed in the context of the consuming user space application
 - > Shared library implementation
 - > Eliminates contention for common data structures



Low Latency Stack Cont.

- Scalable protocol processing model
 - > Ready for the next round of multicore machines
- Anticipates the introduction of adapters capable of queuing based on classification of traffic
 - > Network processing functions on 10G adapters

N

KVM-rt Low Latency Fabric

- Fabric interconnect extends to guest
 - Low overhead guest data path operations
 - > Virtualized IB and iWarp adapter
 - > Hypervisor bypass
 - > RDMA support for guest
 - Input-Output Queues (IOQ)
 - > Generalized para-virt "netchannel" style communication channel for virtualization.
 - > Has native guest IO to kernel subsystem potential
 - > Integrates with low latency netchannels stack



Networking Enhancements

 General enhancements to current stack to support low latency environments:

Support prioritization end-to-end through the entire stack

Extended NAPI to allow queue prioritization

Converted net-rx softirq into workqueue

Extended the workqueue API to incorporate priority

Allows HW classifiers to deliver true QOS to an application

Reduce context switches for SLERT

Collapse NAPI bottom-half processing into the (threaded) IRQ

Software interrupt coalescing

Can schedule the workitem in the future independent of hardware support for coalescing

Reduce unnecessary task wakeups caused by use of single waitqueue for different events(read,write,error)



Networking Enhancements Cont.

- Netchannel style interfaces for tx operations
 - > Eliminates lock contention against TX resources
 - > Allows for greater concurrency while using existing APIs for applications

Novell®

Unpublished Work of Novell, Inc. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary, and trade secret information of Novell, Inc. Access to this work is restricted to Novell employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of Novell, Inc. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. Novell, Inc. makes no representations or warranties with respect to the contents

of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for Novell products remains at the sole discretion of Novell. Further, Novell, Inc. reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All Novell marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

