

788.08P Winter 2008

Programming Models (SDSM)

Xiangyong Ouyang



Backgrounds of Distributed Shared Memory (DSM)

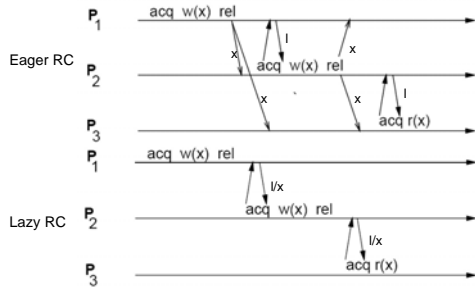


- DSM provides a simple and effective programming model
- Memory consistency model
 - Sequential consistency
 - Processor consistency
 - Weak consistency
 - Release consistency (RC)
 - Eager RC
 - Lazy RC (LRC)
 - Homeless (TreadMark)
 - Home-based (HLRC)

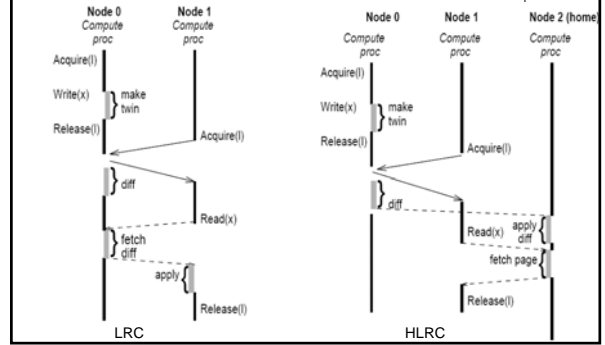
RC vs LRC

LRC reduces message numbers and data amount by

- Not propagate modifications globally at time of release
- Piggyback data on lock transfer messages



LRC vs Home-based LRC (HLRC)



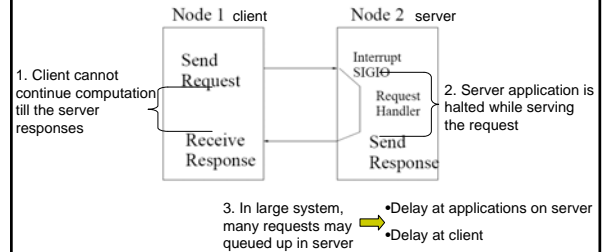
- R. Noronha and D. K. Panda, "Designing High Performance DSM Systems using InfiniBand Features"



Motivations and problems: DSM communication characteristic



- Client-server request-response model
- Problem: inefficient communication



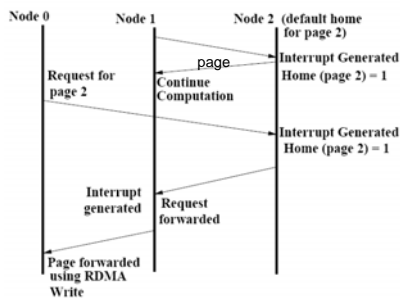
Introduction to InfiniBand

- A framework for system area network
- Switched, channel-based interconnection
- 2 communication semantics
 - Send/Receive
 - RDMA
 - Read/write remote process's user space
 - Remote atomic operations
 - These operations don't require intervention of receiver

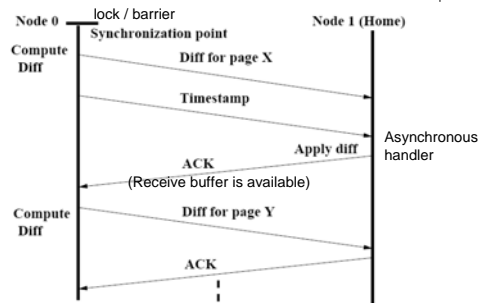
Base protocol--ASYNC

- Home-based Lazy Release Consistency(HLRC)
- Every page/lock is assigned a home
- All requests for a page/lock go to home node.
- All update to a page/lock go to home node
- Updates/Diffs of a page propagate to home node at synchronization points (lock release/barrier)

Base ASYNC protocol (page fetch)



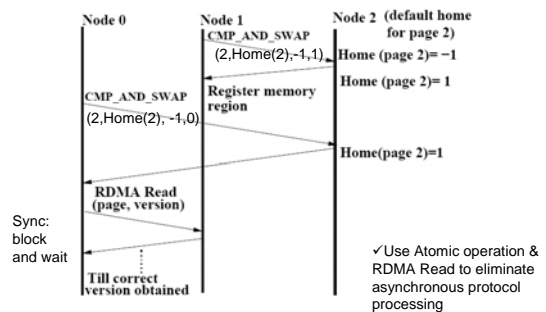
Base ASYNC protocol (diff)



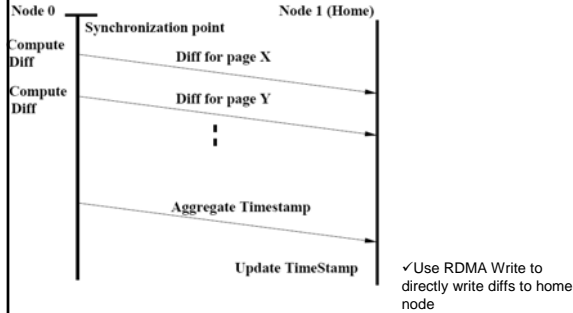
Proposed design methodology

- Utilize One-sided operation to move the DSM protocol-processing into client side.
 - RDMA
 - Remote atomic operation (compare_and_swap)
- Benefits
 - Offload the server -> better scalability
 - Removing asynchronous protocol processing-> better performance
- NEWGENDSM
 - ARDMAR: Atomic operation & RDMA Read (for page fetching)
 - DRAW: RDMA Write (for Diff propagation)

ARDMAR (page fetching)



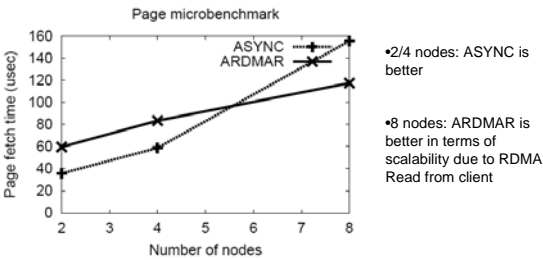
DRAW (diff)



Performance evaluation

- Test bed
 - 8 nodes cluster, connected through MT43132 Eight port switch(4x port)
 - Each node
 - Dual Xeon 2.4G processors, 512MB memory
 - Mellanox MT23108 DualPort 4x HCA

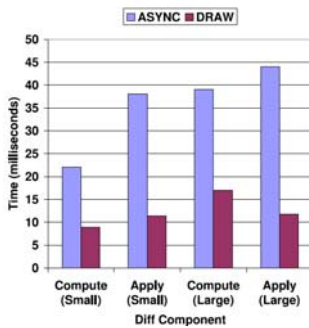
Page fetch micro-benchmark



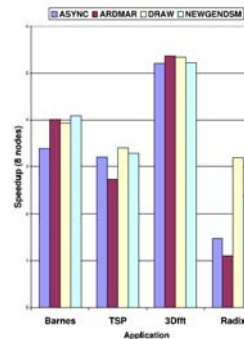
Diff-time micro-benchmark

- Diff-time comprises of 2 components
 - Diff creation time(at client)
 - Compare a dirty page with twin to create run-length encoding + post a descriptor to send the diff to home node
 - Diff application time
 - Receiver applies the differences to the page
- Methodology
 - Node 0 is home of all 1024 pages
 - Node 1 modifies 1 word/all words in every page.
 - At barrier, diffs are created at node 1 (creation time) and applied at node 0(apply time)

Diff-time micro-benchmark

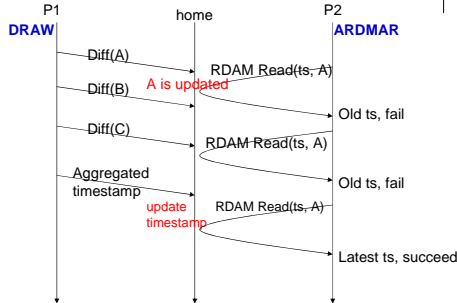


Application level evaluations: speedups



- DRAW is better than ASYNC in all cases
- NEWGENDSM better than ASYNC
- ARDMAR performs worse than ASYNC in TSP/Radix
- NEWGENDSM performs worse than DRAW in 3 cases
 - Combined side-effect of ARDMAR&DRAW

Potential performance loss

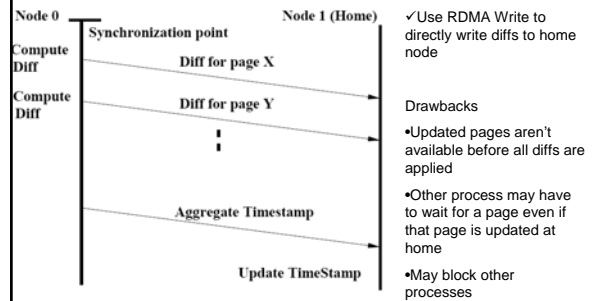


- R. Noronha and D. K. Panda, "Reducing Diff Overhead in Software DSM Systems using RDMA Operations in InfiniBand"

Motivation

- Diff: An import protocol activity in SDSM
 - Compare a modified page to its clean copy(twin), create run-length encoding of difference, send the encoding back to home node
 - Home node apply the diff to original page to bring it updated
 - Home ACKs to sender
- Constitutes great overhead
 - Becomes worse if there is large amount of page sharing

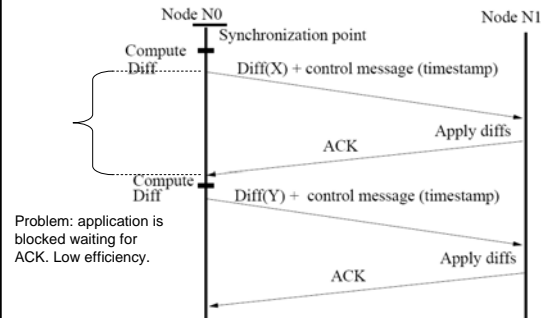
Problem of previous solution: DRAW

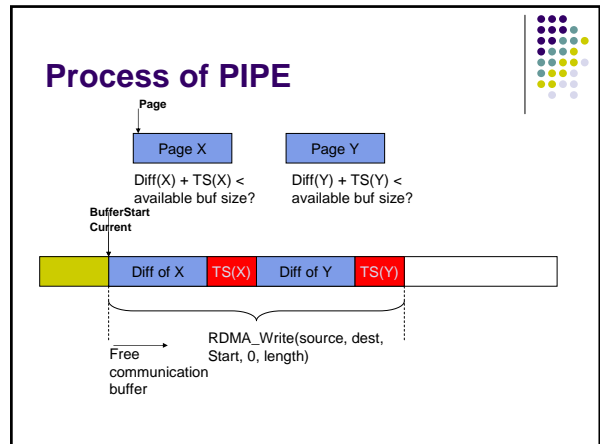
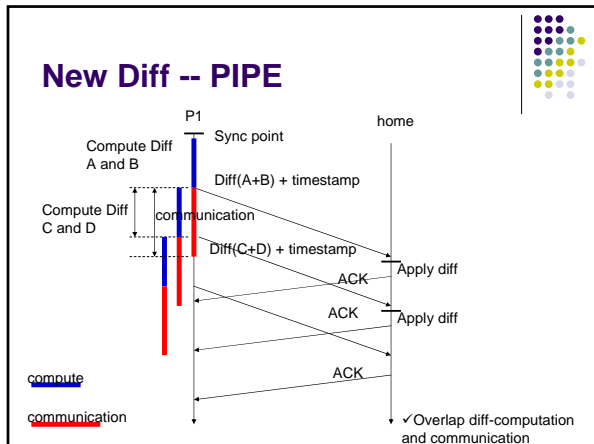


Base protocol--ASYNC

- Home-based Lazy Release Consistency(HLRC)
- Every page/lock is assigned a home
- All requests for a page/lock go to home node.
- All update to a page/lock go to home node
- Updates/Diffs of a page propagate to home node at synchronization points (lock release/barrier)

Diff in base protocol -- ORIG





- ### Performance evaluation
- Test bed
 - 16 nodes cluster, connected through MTS-2400 24 port IB switch(4x port)
 - Each node
 - Mellanox MT23108 DualPort 4x HCA
 - 133MHz PCI-X bus
 - 8 nodes have dual Xeon 2.4G processors, 512MB memory
 - 8 nodes have dal Xeon 3.0G processors, 1GB memory

Application characteristics

Application	Barnes	IS	LU	Ocean
Average Diff Traffic (MegaBytes)	1.83	29.55	10.9	9.16
Average Number of Diffs	6060	7680	15114	14327.56
Average Diff Size (bytes)	317	4034	756.21	670.38
Average Number of Intervals	13	17	129	937
Average Number of Diffs Per Interval	466.15	451.764	117.16	15.29

Table 2. Per node application characteristics for a 16 node run



- ### More considerations
- Pipeline depth
 - The number of messages (packed diffs) PIPE can send before waiting for an ACK form Home node
 - Depends on
 - amount of local buffer, and amount of remote receive buffer
 - Longer pipeline vs shorter pipeline
 - Packed diff size
 - Diff-time = diff-compute time + time to post a send descriptor
 - Large pack size leads to better network bandwidth utilization, and dominates the cost to post a send descriptor
 - But large pack size can make it longer for a update become available at home

- R. Noronha and D. K. Panda, “Can High Performance DSM Systems Designed With InfiniBand Features Benefit from PCI-Express?”

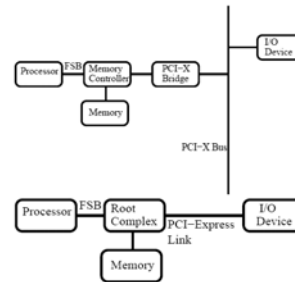
Problems

- InfiniBand can provide high bandwidth
 - 4x IB DDR card provides 20Gb/s
- PCI-X architecture cannot sustain such high bandwidth.
 - Throughput is limited by PCI-X bus
 - PCI-X is shared bus

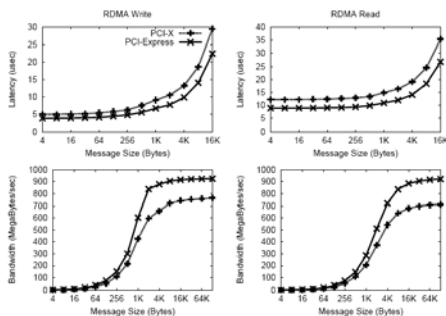
PCI-Express

- Serial point-to-point links
- Each link is made of x1, x2, x4, x8, x16, x32 lanes.
- Each lane operates at 2.5Gb/s
 - use 8/10 encodes
- A hub on mainboard acts as crossbar switch
 - Allows parallelism
 - Multiple pair of devices to communicate at same time

PCI-X vs PCI-Express



Performance comparison of PCI-X and PCI-Express



Motivations

- Use PCI-Express to replace PCI-X
- Measure the performance of NEWGENDSM on PCI-Express, against that of ASYNC protocol

Performance evaluation

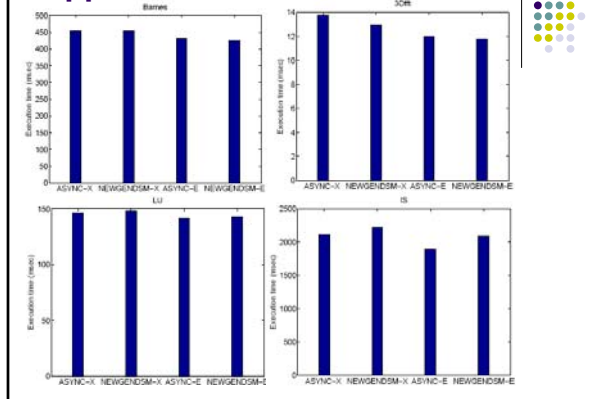
- Test bed
 - 4 nodes cluster, connected through MT43132 switch
 - Each node
 - Dual 3.4G Xeon processor, 512 MB memory
 - 64bit, 133MHz PCI-X and PCI-Express x8
 - MT23102(PCI-X) IB HCA, MT25208 (PCI-Express) IB HCA

Micro-benchmark (page-fetch)

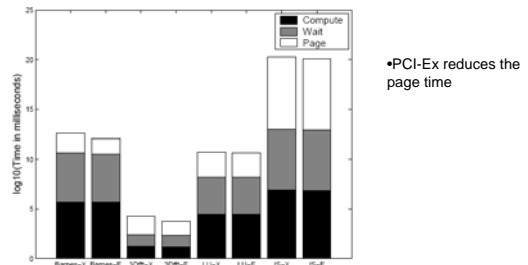
Table 2. Time for the page microbenchmark in μs

Interface	ASYNC (two nodes)	ASYNC (four nodes)	NEWGENDSM (two nodes)	NEWGENDSM (four nodes)
PCI-X	36	51	33	50
PCI-Express	33	40	27	41

Application level evaluation



Breakdown of exec time



The end

- Questions and comments?