Memory Aggregation For KVM
Hecatonchire Project

Benoit Hudzia; Sr. Researcher; SAP Research Belfast
With the contribution of Aidan Shribman, Roei Tell, Steve Walsh, Peter Izsak

November 2012
Agenda

- Memory as a Utility
- Raw Performance
- First Use Case: Post Copy
- Second Use Case: Memory aggregation
- Lego Cloud
- Summary
Memory as a Utility

How we Liquefied Memory Resources
The Idea: Turning memory into a distributed memory service

Breaks memory from the bounds of the physical box

Transparent deployment with performance at scale and Reliability
High Level Principle

Memory Demander

Virtual Memory Address Space

Memory Demanding Process

Memory Sponsor A

Memory Sponsor B

Network
How does it work
(Simplified Version)

Virtual Address → MMU (+ TLB) → Physical Address

- Miss
- Update MMU

Page Table Entry → Coherency Engine

- Remote PTE
- (Custom Swap Entry)
- PTE write

Page request → Network Fabric → Page Response

RDMA Engine

- Extract Page
- Prepare Page for RDMA transfer

Physical Node A

Physical Node B

© 2012 SAP AG. All rights reserved.
Reducing Effects of Network Bound Page Faults

**Full Linux MMU integration (reducing the system-wide effects/cost of page fault)**
- Enabling to perform page fault transparency (only pausing the requesting thread)

**Low latency RDMA Engine and page transfer protocol (reducing latency/cost of page faults)**
- Implemented fully in kernel mode OFED VERBS
- Can use the fastest RDMA hardware available (IB, IWARP, RoCE)
- Tested with Software RDMA solution (Soft IWARP and SoftRoCE) (*NO SPECIAL HW REQUIRED*)

**Demand pre-paging (pre-fetching) mechanism (reducing the number of page faults)**
- Currently only a simple fetching of pages surrounding page on which fault occurred
Transparent Solution

**Minimal Modification of the kernel (simple and minimal intrusion)**

- 4 Hooks in the static kernel, virtually no overhead when enabled for normal operation

**Paging and memory Cgroup support (Transparent Tiered Memory)**

- Page are pushed back to their sponsor when paging occurs or if they are local they can be swapped out normally

**KVM Specific support (Virtualization Friendly)**

- Shadow Page table (EPT / NPT )
- KVM Asynchronous Page Fault
Scalable Active - Active Mode (Distributed Shared Memory)

- Shared Nothing with distributed index
- Write invalidate with distributed index (end of this year)

Library LibHeca (Ease of integration)

- Simple API bootstrapping and synching all participating nodes

- We also support:
  - KSM
  - Huge Page
  - Discontinuous Shared Memory Region
  - Multiple DSM / VM groups on the same physical node
Raw Performance

How fast can we move memory around?
Raw Bandwidth usage

HW: 4 core i5-2500 CPU @ 3.30GHz - SoftIwarp 10GbE - Iwarp Chelsio T422 10GbE - IB ConnectX2 QDR 40 Gbps

- Sequential Walk over 1GB of shared RAM
- Bin split Walk over 1GB of shared RAM
- Random Walk over 1GB of shared RAM

- Not enough core to saturate (?)
- No degradation under high load
- Maxing out Bandwidth
- Software RDMA has significant overhead

© 2012 SAP AG. All rights reserved.
# Hard Page Fault Resolution Performance

<table>
<thead>
<tr>
<th></th>
<th>Resolution time Average (μs)</th>
<th>Time spend over the wire one way Average (μs)</th>
<th>Resolution time Best (μs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SoftIwar p (10 GbE)</td>
<td>355</td>
<td>150 +</td>
<td>74</td>
</tr>
<tr>
<td>Iwarp (10 GbE)</td>
<td>48</td>
<td>4-6</td>
<td>28</td>
</tr>
<tr>
<td>Infiniband (40 Gbps)</td>
<td>29</td>
<td>2-4</td>
<td>16</td>
</tr>
</tbody>
</table>
Average Compounded Page Fault Resolution Time
(With Prefetch)

- IW 10GE Sequential
- IB 40 Gbps Sequential
- IW 10GE Binary split
- IB 40 Gbps Binary split
- IW 10GE Random Walk
- IB Random Walk

Micro-seconds vs. Threads

© 2012 SAP AG. All rights reserved.
Post-Copy Live Migration

Technology first Use Case
Post Copy - Pre Copy - Hybrid Comparison

- Pre-copy (Forced after 60s)
- Post-Copy
- Hybrid - 3 seconds
- Hybrid - 5 Seconds

Host: Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz, 4 cores, 16GB RAM

Network: 10 GB Eth - Chelsio T422-CR I Warp

Workload App Mem Bench (~80% of the VM RAM) Dirting Rate: 1GB/s (256k Page dirtied per seconds)
Post Copy vs Pre copy under load

Virtual Machine:
- 1 GB RAM - 1vCPU
- Workload: App Mem Bench

Hardware:
- Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz, 4 cores, 16GB RAM
- Network: 10 GB Eth Switch – NIC: Chelsio T422-CR (IWARP)
# Post Copy Migration of HANA DB

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Pre-Copy</th>
<th>Post-Copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downtime</td>
<td>N/A</td>
<td>7.47 s</td>
<td>675 ms</td>
</tr>
<tr>
<td>Benchmark</td>
<td>0%</td>
<td>Benchmark Failed</td>
<td>5%</td>
</tr>
<tr>
<td>Performance Degradation</td>
<td>0%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Virtual Machine:**
- 10 GB Ram, 4 vCPU
- Application: HANA (In memory Database)
- Workload: SAP H (TPC-H Variant)

**Hardware:**
- Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz, 4 cores, 16GB RAM
- Fabric: 10 GB Ethernet Switch
- NIC: Chelsio IWARP T422-CR
Memory Aggregation

Second use case: Scaling out Memory
Scaling Out Virtual Machine Memory

Business Problem
- Heavy swap usage slows execution time for data intensive applications

Hecatonchire/ RRAIM Solution
- Applications use memory mobility for high performance swap resource
  - Completely transparent
  - No integration required
  - Act on results sooner
  - High reliability built in
- Enables iteration or additional data to improve results

Solution

VM swaps to memory Cloud

Compression / Deduplication / N-tiers storage / HR-HA

© 2012 SAP AG. All rights reserved.
Redundant Array of Inexpensive RAM: RRAIM

1. Memory region backed by two remote nodes. Remote page faults and swap outs initiated simultaneously to all relevant nodes.

2. No immediate effect on computation node upon failure of node.

3. When a new remote enters the cluster, it...
QuickSort Benchmark with Memory Constraint

<table>
<thead>
<tr>
<th>Memory Ratio (constraint using cgroup)</th>
<th>DSM Overhead</th>
<th>RRAIM Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>3:4</td>
<td>2.08%</td>
<td>5.21%</td>
</tr>
<tr>
<td>1:2</td>
<td>2.62%</td>
<td>6.15%</td>
</tr>
<tr>
<td>1:3</td>
<td>3.35%</td>
<td>9.21%</td>
</tr>
<tr>
<td>1:4</td>
<td>4.15%</td>
<td>8.68%</td>
</tr>
<tr>
<td>1:5</td>
<td>4.71%</td>
<td>9.28%</td>
</tr>
</tbody>
</table>
Scaling out HANA

Virtual Machine:
- 18 GB Ram, 4 vCPU
- Application: HANA (In memory Database)
- Workload: SAP-H (TPC-H Variant)

Hardware:
- Memory Host: Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz, 4 cores, 16GB RAM
- Compute Host: Intel(R) Xeon(R) CPU X5650 @ 2.56GHz, 8 cores, 96GB RAM
- Fabric: Infiniband QDR 40Gbps Switch + Mellanox ConnectX2

<table>
<thead>
<tr>
<th>Memory Ratio</th>
<th>DSM Overhead</th>
<th>RRAIM Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:2</td>
<td>1%</td>
<td>0.887%</td>
</tr>
<tr>
<td>1:3</td>
<td>1.6%</td>
<td>1.548%</td>
</tr>
<tr>
<td>2:1:1</td>
<td>0.1%</td>
<td>-</td>
</tr>
<tr>
<td>1:1:1</td>
<td>1.5%</td>
<td>-</td>
</tr>
</tbody>
</table>
Transitioning to a Memory Cloud (Ongoing work)

Memory VM
Memory Sponsor

Compute VM
Memory Demander

Combination VM
Memory Sponsor & Demander

Many Physical Nodes Hosting a variety of VMs

Memory Cloud Management Services (OpenStack)
Lego Cloud
Going beyond Memory
Virtual Distributed Shared Memory System

(Compute Cloud)

**Compute aggregation**
- Idea: Virtual Machine compute and memory span Multiple physical Nodes

**Challenges**
- Coherency Protocol
- Granularity (False sharing)

**Hecatonchire Value Proposition**
- Optimal price / performance by using commodity hardware
- Operational flexibility: node downtime without downing the cluster
- Seamless deployment within existing cloud

Fast RDMA Communication
Disaggregation of datacentre (and cloud) resources

(Our Aim)

Breaking out the functions of Memory, Compute, I/O, and optimizing the delivery of each.

Disaggregation provides three primary benefits:

- **Better Performance:**
  - Each function is isolated => limiting the scope of what each box must do
  - We can leverage dedicated hardware and software => increases performance.

- **Superior Scalability:**
  - Functions are isolated from each other => alter one function without impacting the others.

- **Improved Economics:**
  - Cost-effective deployment of resource => improved provisioning and consolidation of disparate equipment
Summary
Hecatonchire Project

- **Features:**
  - Distributed Shared Memory
  - Memory extension via Memory Servers
  - HA features
  - Future: Distributed Workload executions
    - *Use standard Cloud interface*
    - *Optimise Cloud infrastructure*
    - *Support COTS HW*
Key takeaways

- Hecatonchire project aim at disaggregating datacentre resources
- Hecatonchire Project currently deliver memory cloud capabilities
- Enhancements to be released as open source under GPLv2 and LGPL licenses by the end of November 2012
- Hosted on GitHub, check: www.hecatonchire.com
- Developed by SAP Research Technology Infrastructure (TI) Programme
Thank you

Benoit Hudzia; Sr. Researcher;
SAP Research Belfast
benoit.hudzia@sap.com
Backup Slide
Instant Flash Cloning On-Demand

Business Problem
- Burst load / service usage that cannot be satisfied in time

Existing solutions
- Vendors: Amazon / VMWare/ rightscale
- Startup VM from disk image
- Requires full VM OS startup sequence

Hecatonchire Solution
- Go live after VM-state (MBs) and hot memory (<5%) cloning
- Use post-copy live-migration schema in background
- Complete background transfer and disconnect from source

Hecatonchire Value Proposition
- Just in time (sub-second) provisioning
DRAM Latency Has Remained Constant

CPU clock speed and memory bandwidth increased steadily (at least until 2000)

But memory latency remained constant – so local memory has gotten slower from the CPU perspective

Source: J. Karstens: In-Memory Technology at SAP. DKOM 2010
CPUs Stopped Getting Faster

Moore’s law prevailed until 2005 when core’s speed hit a practical limit of about 3.4 GHz

Since 2005 you do get more cores but the “single threaded free lunch” is over

Effectively arbitrary sequential algorithms have not gotten faster since

Source: “The Free Lunch Is Over..” by Herb Sutter

Source: http://www.intel.com/pressroom/kits/quickrefyr.htm
While ... Interconnect Link Speed has Kept Growing

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet (1979 - )</td>
<td>10 Mbit/sec</td>
</tr>
<tr>
<td>Fast Ethernet (1993 - )</td>
<td>100 Mbit/sec</td>
</tr>
<tr>
<td>Gigabit Ethernet (1995 - )</td>
<td>1000 Mbit/sec</td>
</tr>
<tr>
<td>ATM (1995 - )</td>
<td>155/622/1024 Mbit/sec</td>
</tr>
<tr>
<td>Myrinet (1993 - )</td>
<td>1 Gbit/sec</td>
</tr>
<tr>
<td>Fibre Channel (1994 - )</td>
<td>1 Gbit/sec</td>
</tr>
<tr>
<td>InfiniBand (2001 - )</td>
<td>2 Gbit/sec (1X SDR)</td>
</tr>
<tr>
<td>10-Gigabit Ethernet (2001 - )</td>
<td>10 Gbit/sec</td>
</tr>
<tr>
<td>InfiniBand (2003 )</td>
<td>8 Gbit/sec (4X SDR)</td>
</tr>
<tr>
<td>InfiniBand (2005 - )</td>
<td>16 Gbit/sec (4X DDR)</td>
</tr>
<tr>
<td>InfiniBand (2007 - )</td>
<td>24 Gbit/sec (12X SDR)</td>
</tr>
<tr>
<td>InfiniBand (2011 - )</td>
<td>32 Gbit/sec (4X QDR)</td>
</tr>
<tr>
<td>40-Gigabit Ethernet (2010 - )</td>
<td>40 Gbit/sec</td>
</tr>
<tr>
<td>InfiniBand (2012 - )</td>
<td>56 Gbit/sec (4X FDR)</td>
</tr>
<tr>
<td>InfiniBand (2012 - )</td>
<td>100 Gbit/sec (4X EDR)</td>
</tr>
</tbody>
</table>

Panda et al. Supercomputing 2009
Result: Remote Nodes Have Gotten Closer

Accessing DRAM on a remote host via IB interconnects is only 20x slower than local DRAM.

And remote DRAM has far better performance than paging in from an SSD or HDD device.

Fast interconnects have become a commodity - moving out of the High Performance Computing (HPC) niche.

HANA Performance Analysis, Chaim Bendelac, 2011
Post-Copy Live Migration (pre-migration)

Pre-migrate → Reservation → Stop and Copy → Page Pushing 1 Round → Commit

Live on A → Downtime → Degraded on B → Total Migration Time → Live on B

Guest VM

Host A → Host B
Post-Copy Live Migration (reservation)

Host A

Guest VM

Stop and Copy

Page Pushing 1 Round

Commit

Reservation

Pre-migrate

Downtime

Degraded on B

Total Migration Time

Live on A

Live on B

Host B
Post-Copy Live Migration (stop and copy)
Post-Copy Live Migration (post-copy)

Host A -> Guest VM -> Page fault -> Page push -> Guest VM -> Host B

- Pre-migrate
- Reservation
- Stop and Copy
- Page Pushing 1 Round
- Commit

- Live on A
- Downtime
- Degraded on B
- Total Migration Time
- Live on B

© 2012 SAP AG. All rights reserved.
Post-Copy Live Migration (commit)

Host A

Pre-migrate
Reservation
Stop and Copy
Page Pushing 1 Round
Commit

Live on A
Downtime
Degraded on B
Total Migration Time
Live on B

Host B

Guest VM

Reservation
Stop and Copy
Page Pushing 1 Round
Commit

Live on A
Downtime
Degraded on B
Total Migration Time
Live on B