VMware, Inc. Invention Disclosure Form

Overview

File #:

Title*

A METHOD TO OPTIMIZE MEMORY PAGE TRANSPORTATION IN VMOTION

Technology Group:

Technology Group Category*

SDDC Compute

Product or Technology*

SDDC Compute - Memory

Received Date

vmware

VMware, Inc. Invention Disclosure Form

Disclosure

Previous Public Disclosure:

Has this invention been made known to anyone outside of VMware and not subject to an NDA?*

No

When?

How?

Anticipated Public Disclosure:

Outside of a non-disclosure agreement (NDA), is there any planned disclosure of this invention or release of a product incorporating this invention to anyone outside of VMware?

No

When?

How?

Supporting Documents:

Documents and Attachments:	
Document Name	**Subject**
vmotionzerostripped.pdf	VMotionSteps

vmware

VMware, Inc. Invention Disclosure Form

Third Party Interest

Development Funding:

Is the development of this invention being funded by an agency of the U.S. Government?*

No

Agency

Contract Number

Special Contract Limitations:

Are you under a duty to maintain any aspect of this invention in confidence or to assign all or any part of it to any other individual (for example, business partner), company (for example, previous employer), organization, or educational institution?*

No

Describe

Have you developed all or any part of this invention using the equipment, consulting services, or facilities of any other individual, company, organization or educational institution?*

No

Describe

Do you know of any other potential limitations to VMware's exclusive right to own or develop this invention?*

No

Describe

VMware Project:

Is the development of this invention related to a current VMware Project?*

Yes

Enter Name (or Codename) of Project:*

vsphrere

Is this a project that VMware is collaborating on with a third party (or parties)?*

No

Enter Name of Third Parties:

Supporting Documents:

Documents and Attachments	
Document Name	**Subject**

vmware

VMware, Inc. Invention Disclosure Form

Invention Description

Problem to be Solved:

What problem does the invention solve?*

During VMotion, ESXi source host needs to send the non-zero page to ESXi destination host when detecting zero page configuration is enabled, for the present when ESXi source detects a non-zero page, it will copy the full 4096 bytes page to socket buffer then sends the page, at destination, a new page is allocated and copy data from buffer. for virtual machine with large memory, vmotion may take very long time due to network bandwidth bottleneck, lots of non-zero pages need to be transferred by network.

Summary of the Invention:

Briefly, summarize the invention and how it solves the stated problem.*

During vmotion, virtual machine will be transferred from source ESXi host to destination ESXi host via network, the current method of regular page transmision is to copy the whole zon-zero page into socket buffer then send it out, the destination host receives this buffer, then allocates a new page and copies data from socket buffer to this page.

We did an investigation of memory page inside layout of Windows 7 and RHEL6, found that the continuous 0 bits at the start and the end of page may be up to 6~7% of a page on Windows 7 and 8~9% of a page on RHEL6, this value can be higher if we run certain applications on operating system.

Our method is an improvement based on the investigation result above, the virtual machine page of 4096 bytes size is divided 64 blocks, each block has 64 bytes, we strip zero bits at the begin and the end of this page, we only transfer the non-zero block data in the middle of this page.

Detailed steps are as following.

ESXi Source Host:

1. Check if the 64 byte block is a zero filled block, if yes, redo the step 1 with the next block until the ending block of this page, if the block is no-zero, go to step 2.

2. Record the start position of non-zero block, go to step 3.

3. Go to the ending block of this page and checking if it's zero filled, if yes, redo the steps 3 with previous block until reach the start position recorded in step 2, if no go to step 4.

4. Record the end position of non-zero block with in this page.

5. Copy the start poistion and end position data to meta-data of this page.

6. Copy the data between start and end position in page into socket buffer.

7. ESXi source host send the buffer data to destination host.

ESXi Destination Host:

1. ESXi destination received buffer data from source host.

2. Extract the start and end position from page-meta data.

3. Alloc a new page for destination virtual machine.

4. Set the memory from 0 to start position in this page with zero.

5. Set the memory from end position to PAGE_SIZE(4096) in this page with zero.

6. Use memory from start to end position in this page to receive data from socket buffer, the received data length is (end position - start position).

By applying the method above we will strip the zero bytes at the start and the end of page without introducing any overhead and this can save 6~9% bandwidth for the cases that we running original Windows 7 or RHEL 6 without application.

Prior Art:

How have others tried to solve the problem and in what ways have their solutions been inadequate?

Supporting Documents:

Documents and Attachments	
Document Name	**Subject**
vmotionzerostripped.pdf	VMotionSteps

vmware

VMware, Inc. Invention Disclosure Form

Inventors

Inventor [1]	
Full Name:	Liu, GengSheng
Home Address:	
Work Email:	gengshengl@vmware.com
Citizenship:	N/A
Office Location:	3401 Hillview Avenue Palo Alto, CA 94304
Work Phone Number:	650-427-5000