CSC548, Fall 2006
Homework 4 – Project Proposal

Name: Arun Babu Nagarajan
Project: Enhanced Proactive Falut tolerance system for HPC with xen virtualization
Website: *http://www4.ncsu.edu/~abnagara/csc548_project/index.html*

Background:

The earlier work "Proactive Fault Tolerance for HPC with xen virtualization"[1] aims at providing a fault tolerance solution to the HPC clusters by automatically migrating the OS image from an 'unhealthy' node to a 'healthy' one. The health of a node is determined by parameters like CPU temperature, fan speed, etc,.

A daemon – Proactive Fault tolerance daemon (PFTd) runs and monitors the health of the node on a continuous basis by reading the hardware sensors using OpenIPMI. On detecting a deteriorating health, PFTd migrates whole of the OS image to a spare node and starts execution again. Xen virtualization is used to migrate the images from one node to another in a 'live' fashion with little overhead.

Problem Statement:

i) Improve the currently implemented FT system by including a functionality to proactively pre-deploy parts of the OS image to the spare node and help cutting down cost of migration of VM.

ii) Instrument the Xen tools to gather more details like pages sent in each iteration, dirtying rate etc, and perform a sensitivity study of the benchmarks on the exact time at which the migration is initiated.

Problem Description

The currently implemented FT system necessitates the health monitoring system to figure out node failure before 13 – 40 seconds (varies based on the application) so that the VM can be safely migrated to the destination. The way the migration happens is that during the initial iteration, the pages are sent over to the target. The next iteration sends the pages, which have been dirtied since the previous send, and so on. It has been observed with the NAS parallel benchmarks that a large chunk of the pages (in fact more than 90%) are sent during the initial iteration and the other pages are sent repeatedly during the following iterations, (depending on the working set at the time the migration command was initiated).

We would like to exploit this behavior by sending some part of the VM image earlier than required to the spare node so that we could significantly cut down on the transfer cost.

Proposed approach and issues:

The following list summarizes the tasks to be done and reason for doing.

- A study needs be done on how many pages of the initial iteration are that of the application and that of the OS. We could pre-deploy only the OS part so this study would help to roughly estimate the amount of savings.
    - To get an idea on common pages between OS images, two VMs can be booted up and page-wise walkthrough should be done to find differing pages
    - As an extension, an application can be run on the VM and again the differing pages should be identified to get much better idea.

- The structure of the xen OS image file needs to be studied
    - As already discussed, we can pre-deploy only OS and not application pages. But even with OS, we cannot deploy whole of the OS part because of the shared libraries. So we need to probably exclude the shared libraries while pre-deploying.
    - OS image would also include current register contents and other vital information. We would need a thorough understanding of the image file to properly setup the VM at the receiving end.

- Linux memory management needs to be studied to learn how to traverse the main memory and read list of loaded shared libraries in the memory. (This needs to be done from within the Xen VMM)

- After the above discussed information are available the plan is to pre-deploy (or even boot a VM at target node, rip the common parts of OS image and store it to a file) OS parts. When actual migration is to be done, the already available information (OS parts) and the newly available application pages need to be weaved together to form an active image file. When the transfer is complete, this image file should be ready to be booted up.

- For sensitivity study, the benchmarks need to be instrumented with interleaved sleeps and coordinated migrate commands during sleep to make sure we observe a less overhead (working set) during the migration command when the processes sleep.

References:

i) A. Nagarajan, F. Mueller, Proactive Fault Tolerance for HPC with Xen Virtualization, submitted to IPDPS 2007.

ii) P. Barham et al., Xen and the Art of Virtualization. SOSP 2003.

iii) C.Clark et al, Live migration of virtual machines, 2nd Symposium on Networked Systems Design and Implementation, May 2005.

iv) http://linux-mm.org/VirtualMemory

v) http://tldp.org/LDP/khg/HyperNews/get/memory/linuxmm.html