# CSC 2231

# Project: Application Level Speculative Processing through VM Forking.

## Progress report 1

Francescio Iorio

Alex Tessier

Our current status is that we now have 2 machines setup and configured with Xen and SnowFlock. Various API schemes have been designed, discussed and are currently being investigated.

We started our installation odyssey by installing Debian 5.0.8 on the Dell Optiplex 795 according to the instructions on the SnowFlock Wiki. After a successful install, we then tried build and install Xen + SF on top of the 2.6.23 kernel (Debian 5.0.8). Xen + SF built and installed, but on the final reboot, it wouldn't boot -- the CPUs, Intel SandyBridge CPUs, would not boot. We attempted to fix this by disabling multiprocessor/multicore support in the EFI settings and were able to boot. Upon reboot, the kernel could not mount the SATA disks because the SATA controller kernel module/driver didn't support SandyBridge processors.

Our next attempt was to try and install the mainline version of SnowFlock, which is based on the 2.6.26 kernel. We reformatted the volumes and reinstalled Debian 5.0.8 for a clean install. All the build attempts from the mainline failed. Kernel.org was down at that time (it had been hacked apparently) and all attempts to get patches failed. We then tried to download a snapshot of a vanilla 2.6.26 and setup a local Mercurial repository to store it. We then tried to apply the various Xen and SF patches from the mainline. We then contacted Andres Lagar-Cavilla for advice and support for building the mainline snow flock from scratch. We tried to follow the instructions for almost an entire week unsuccessfully. Build errors, missing dependencies, patch ordering and conflicts between Xen and SF patches caused a lot of problems. We managed to get a few builds, but most of these were half-baked and missing critical pieces. Xen appeared to be building Ok, the kernel would boot, but we could never get built image for dom0 to startup.

So we abandoned the Optiplex 795 and sought help from Eric to locate a dual core Dell Optiplex 745 with older processors and chipset. We reformatted, installed Debian from scratch, pulled the version 2.6.18 of SF and built it successfully. The OS would boot, but onboard NIC was not recognized (no drivers successfully loaded ). We couldn't start the daemons at all without a working NIC/ethernet stack.

We then, with Eric's help, tried to locate an older ethernet card (intel e100 or compatible card). After reformatting and reinstalling Debian with the new card (and the onboard disabled), we then re-attempted to install and configure Xen/SF. This time, the install was successful. We then pulled the SF control stack from mercurial and built it. We encountered a few problems initially -- the scripts mandated the presence of eth0 AND eth1 or the daemons would not start. As we had only 1 NIC. After consulting with Nilton, we were able to re-configure the scripts to start up the SF daemons with 1 NIC.

The virtual machine image in the documentation for SF was downloaded and we tried to get Xen+SF to start it up. Neither Xen nor Xen+SF could start the image. We sought out Nilton's help again and were unable to determine the nature of the failure with the preconfigured images. We were able to acquire some working VM images directly from Nilton and these finally work.

We then acquired a second machine, a Dell Optiplex 780 with 4-cores, and repeated the (now) working installation and configuration process. We took care to document the procedure as well for future reference. We still had problems with the NIC in this new machine and had to find a working NIC, which required physical modification to be able to insert it into the half height case. After a few more days of installing and configuring, we now had 2 working SnowFlock machines.

This consumed considerably more time than originally anticipated, delaying our current progress. We are now playing catch up and are working on creating a standalone application for the purposes of experimentation. The requirements for this GUI driven application are simple -- expose some user facing choices that are reasonably computing intensive enough to warrant speculative execution. We are currently evaluating some image processing algorithms and are also considering some real-time stock market based algorithms as well but have not finalized on a design. The GUI will most likely be implemented on top of Qt in C++ which runs on gnome (this still requires some testing which is currently underway).

We are currently working on API design and application selection and development. We are discussing and analyzing 3 API variants, which can be summarized as follows:

V1: A SF\_fork inspired API. A special execution context is constructed and passed to each clone to set initial and end points of speculative execution.

V2: A parent-child leading approach where the main VM reaches the point of speculation and bails just after forking a single clone.

V3: A custom event-loop application framework with custom messages and message handlers for setting up and managing application and clone states and managing joins.