1 What is PAPI? / Where is PAPI? / Getting PAPI

PAPI is the Performance Application Programming Interface. The software is available at `http://icl.cs.utk.edu/`. The latest release is version 3.6.2 and is available at `http://icl.cs.utk.edu/projects/papi/downloads/papi-3.6.2.tar.gz`
Under Linux, in order to use PAPI, we need to patch the kernel. The goal of this lab is to learn how to patch our kernel in order to use PAPI in order to access the hardware counters available on our processor.
Download PAPI and unpack the tarball:
tar –zxvf papi-3.6.2.tar.gz
The (detailed!) installation instructions are in `papi-3.6.2/INSTALL.txt` for PAPI itself and in `papi-3.6.2/src/perfctr-2.6.x/INSTALL`.

2 Getting the Kernel

Your kernel can come from either:

- `http://www.kernel.org`
- Your Linux distribution

For this lab, we will be using the latest PAPI supported kernel from `http://www.kernel.org`. It is the *2.6.25* kernel and is available at `http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.25.tar.bz2`.
Unpack the Linux Kernel tarball:
tar –jxvf linux-2.6.25.tar.bz2

3 Saving *important* stuff

We want to make sure we keep a backup of our old kernel configuration file! If we have already built Linux from this source before, we will want to save the `/usr/src/linux-version/.config` file so we don't mess it up. If not, we will want to save the configuration file from `/proc/config.gz` if your kernel was built with support for saving its configuration (usually a very good idea!).

4 Patching the Kernel

PAPI provides two different interfaces to the hardware counters. Historically, PAPI only supported the Perfmon interface: Now it supports both the Perfmon and Perfmon2 interfaces. The Perfmon interface is more mature so we will be using it for this lab. *YMMV*!
In order to patch the kernel, PAPI has a simple script that will take care of the patching work for you!

From the top-level of the Kernel source directory (/usr/src/linux-2.6.25 or wherever you unpacked it), run: `/path/to/papi-3.6.2/src/perfctr-2.6.x/update-kernel`.

5 Verify APIC support

In order to get proper access to the hardware counters, such as things like detecting overflows, etc. we will need to turn on "APIC" support in our kernel. If we are building an SMP kernel, this will be enabled by default. However, if we are building a kernel for a uniprocessor machine, we will have to explicitly enable it.
APIC support is located in the "Processor type and features" section of the Kernel configuration.
Confirm APIC support:

```
grep PIC .config
CONFIG_X86_LOCAL_APIC=y
CONFIG_X86_IO_APIC=y
```

6 Compile and Install the Kernel:

In order to compile the kernel, in the kernel's top-level source directory, run:

```
make
```

Then we will want to install the kernel. One plausable way to do this is to run: `make install`. Do whatever suits your taste.

7 Reboot (and cross your fingers!)

Now we will reboot our system into our new kernel.
After rebooting, the first thing we will want to do is to create a device file to access the performance counters. In order to do this, execute the following commands:

```
mknod perfctr c 10 182
chmod 644 perfctr
```

Then, we will want to build the PAPI distribution. In the PAPI source directory, we will run: `make`. This will build the PAPI package.
Now, we will want to verify that we got APIC support working correctly. The simplest way to do this is to run the `perfex` binary located in `perfctr/examples/perfex` with the `--i` option. The output should look something like:

```
PerfCtr Info:
abi_version 0x05010501
driver_version 2.6.13 DEBUG
cpu_type 5 (Intel Pentium III)
cpu_features 0x7 (rdpmc,rdtsc,pcint)
cpu_khz 731060
tsc_to_cpu_mult 1
```

```
cpu_nrctrs 2
cpus [0,1], total:  2
```

*NOTE: The above output is from an *OLD* version of Perfctr*

8 Installing PAPI

Finally, we are ready to install the PAPI Software. In the PAPI source directory, we will want to run: `make PREFIX=/usr/papi install`. Replace `/usr/papi` with wherever you would like PAPI to be installed.
WE ARE DONE!

9 The *EASY* Way

Traditionally, the method that we've just used was the *ONLY* way to get performance counter support in our kernel. Now, there is support in the kernel configuration itself!
Under the "Processor type and features", there is a "Performance Counters" section where we can simply select that we would like Perfctr support!.
In the *"Bad Old Days"* HPC kernels required *a lot* of specialized patches. You would need to patch in order to support:

- the larger amounts of memory that we typically see on HPC systems
- specialized interconnect networks
- performance counters
- patches for various other things

...but, those were the "Bad Old Days"...