**Building TAU with ADIOS**

1. Build ADIOS 1.12 as usual (source available at [https://github.com/ornladios/ADIOS](https://mail.bnl.gov/owa/redir.aspx?C=QukvnXz8jWDXDYNy-c09lbAFrd46BH9DLmwAnarmtXJTTAYikL3UCA..&URL=https%3a%2f%2fgithub.com%2fornladios%2fADIOS))
2. Configure and build TAU 2.26.2 as usual, adding the -adios=/path/to/adios/installation flag at the configure step (source available at [http://tau.uoregon.edu/tau.tgz](https://mail.bnl.gov/owa/redir.aspx?C=ricMk2osIVHn4MetaYZJyTc1OnjlhNvMEJ7jFvHKBxdTTAYikL3UCA..&URL=http%3a%2f%2ftau.uoregon.edu%2ftau.tgz))
3. Add /path/to/tau/$arch/bin to your PATH/path environment variable (where “/path/to/tau” is your TAU installation location)
4. set the TAU\_MAKEFILE to the Makefile that matches your TAU configuration, located in /path/to/tau/$arch/lib/Makefile.tau-\*

**Instrumenting using TAU**

When linking your executable(s), I assume you are using the recommended ADIOS technique of using the adios\_config program to get link flags. If that is the case, TAU has something similar that works as well. Replace "${ADIOSDIR}/bin/adios\_config -l -f” with "tau\_cc.sh -tau:showlibs ${ADIOSDIR}/bin/adios\_config -l -f” to get both the TAU link flags/libraries as well as the ADIOS flags and libraries. If replacing it in a Makefile, it might look something like this: "$(shell tau\_cc.sh -tau:showlibs) $(shell ${ADIOSDIR}/bin/adios\_config -l -f)”. Make sure the TAU libraries are before the ADIOS libraries in the ordering.

**Setting up Directories for Dumping Information**

***Changes in workflow.swift***

import io;

import launch;

import string;

**string envs[] = [ "TAU\_PROFILE=1", "PROFILEDIR=/path to the directory/" ];**

program1 = "./heat\_transfer\_adios1";

arguments1 = split("heat 4 3 40 50 6 500", " ");

printf("swift: launching: %s", program1);

**exit\_code1 = @par=12 launch\_envs(program1, arguments1,envs);**

printf("swift: received exit code: %d", exit\_code1);

if (exit\_code1 != 0)

{

printf("swift: The launched application did not succeed.");

}

else

{

printf("Entering the second Stage----");

}

**string envs1[]= [ "TAU\_PROFILE=1", "PROFILEDIR=/path to the directory/" ];**

program2 = "stage\_write/stage\_write";

arguments2 = split("heat.bp staged.bp FLEXPATH \"\" MPI \"\"", " ");

printf("size: %i", size(arguments2));

printf("swift: launching: %s", program2);

**exit\_code2 = @par=2 launch\_envs(program2, arguments2, envs1);**

printf("swift: received exit code: %d", exit\_code2);

if (exit\_code2 != 0)

{

printf("swift: The launched application did not succeed.");

}

else

{

printf(" The workflow is done!!!! ");

}