PTF Fixed and Adaptive sequence Plugins User's Guide

PTF Version: 1.1
Fixed sequence Plugin Version: 1.0
Adaptive sequence Plugin Version: 1.0

Robert Mijaković

13.04.2015

Contents

Chapter 1

Introduction

Optimization of scientific codes requires optimizing many aspects of the application. As PTF plugin's optimizes only a single aspect of the application a special type of plugins, called meta-plugins, are developed. Meta-plugins act as an intermediate between PTF and the component plugins. Two meta-plugins are available with PTF, a fixed sequence and an adaptive sequence.

The fixed sequence meta-plugin first loads the three component plugins. It then runs each plugin individually in the following fashion. First, the first plugin is selected and executed. When it finishes its execution the meta-plugin selects the next component plugin one after another. After all plugins has finished the meta-plugin reports the best combination of paramters to the user. Due to limitations of the current approach the next plugin does not take into account the best variant detected by the previous plugin.

The other meta-plugin an adaptive sequence of the same component plugins. The approach for both plugins is basically the same except that the adaptive meta-plugin first run a pre-analysis that gathers the information for the criteria used to select the plugins to run.

PTF focuses mostly on scientific codes and therefore meta-plugins incorporates component plugins that can be applied on any scientific code. The meta-plugins incorporate component plugins namely:

1. Compile flag Selection Plugin
2. MPI Parameters Plugin
3. DVFS Plugin

As both meta-plugins run component plugins, it is highly recommended to check their user's guides for more information about how to configure and run them.

Chapter 2

Quick Start

2.1 Quick installation

Fixed and adaptive sequence plugins are being installed along with the Periscope Tuning Framework. Please refer to the *PTF Installation Guide* for a complete description of the installation process.

2.2 Basic configuration

The fixed and adaptive sequence plugins run component plugins which might have special configuration files. Therefore, please consult user's guides of individual plugins.

2.2.1 Fixed sequence plugin

Due to its simlicity the fixed sequence plugin does not require the configuration file.

2.2.2 Adaptive sequence plugin

In order to use the adaptive sequence plugin, a set of configuration instructions are required. These instructions are read at execution time from the configuration file defined.

To start with, copy the default configuration file `as_config.cfg.default` into the folder containing the executable of your application and rename it to `as_config.cfg`.

```
$PSC_ROOT/templates/cfs_config.cfg.default →
$APP_ROOT/.../cfs_config.cfg
```

For example, for the NPB benchmarks[1], copy the configuration file into the `bin` folder:

```
>cp $PSC_ROOT/templates/as_config.cfg.default
NPB3.3-MZ/bin/as_config.cfg
```

The adaptive sequence plugin decides which plugins to run based on the criteria.

The criteria used in the adaptive plugin are:

- Compiler Flags Selection: The instructions per second rate is below a given threshold.

$$Condition = 1 \quad iff \quad f_{IPS} < f_{IPSThreshold} \tag{2.1}$$

$$Severity = f_{IPS} = \frac{N_{Instructions}}{t_{Exec}} \tag{2.2}$$

- MPI Parameters: The execution time spent for MPI exceeds a certain percentage of the phase execution time.

$$Condition = 1 \quad iff \quad f_{MPI} < f_{MPIThreshold} \tag{2.3}$$

$$Severity = f_{MPI} = \frac{t_{MPIExec}}{t_{Exec}} \tag{2.4}$$

- Dynamic Voltage and Frequency Scaling: The instructions per Joule rate is below a given threshold.

$$Condition = 1 \quad iff \quad f_{IPJ} < f_{IPJThreshold} \tag{2.5}$$

$$Severity = f_{IPJ} = \frac{N_{Instructions}}{E_{Exec}} \tag{2.6}$$

Edit `as_config.cfg` to reflect thresholds for your application.

```
threshold "compilerflags" = 1;
threshold "dvfs" = 1;
threshold "mpiparameters" = 0.1;
```

[1] See `http://www.nas.nasa.gov/publications/npb.html` for downloading and documentation.

2.3 Running Meta-plugins

The fixed and adaptive sequence plugins runs as any other plugins within the Periscope Tuning Framework. They can be started using `psc_frontend` (see also *PTF User's Guide*) by setting the `tune` flag to `fixedseqeuence` or `adaptivesequence`.

```
--tune=fixedseqeuence
--tune=adaptiveseqeuence
```

For the NPB BT-MZ example with the adaptive sequence plugin, one would call from within the folder containing the execution file:

```
psc_frontend --apprun="./bt-MZ.W"
--mpinumprocs=1 --tune=adaptiveseqeuence
--force-localhost --as-config="as_config.cfg"
--cfs-config="cfs_config.cfs"
```

This will start the measurements and the CFS tuning strategy for the uninstrumented version of the BT benchmark using one process.

Please note that the application *has to be built* and the executable file passed to the `apprun` flag *must exist* when calling `psc_frontend`.

Chapter 3

How To Use the Tuning Advice

Upon successful completion, the fixed and adaptive sequence plugin outputs a list of all tested scenarios for indidivudal component plugins as well as the id of the best scenario for each sub-plugin.

For more information please check How to Use the Tuning Advice chapters of the component plugins.

Chapter 4

Limitations and Known Issues

Due to limitations of the current approach the next plugin does not take into account the best variant detected by the previous plugin.

For more information please check Limitations and Known Issues chapters of the component plugins.