Cpu Dynamics (DCVS) Design Document

January 26, 2010

Table Of Contents

[1 Introduction 3](#_Toc252380548)

[2 Overview 3](#_Toc252380549)

[3 NPA Interface 4](#_Toc252380550)

[3.1 core/cpu 4](#_Toc252380551)

[3.1.1 REQUIRED 4](#_Toc252380552)

[3.1.2 IMPULSE 4](#_Toc252380553)

[3.1.3 ISOCHRONOUS 4](#_Toc252380554)

[3.2 core/cpu/algorithm/disable 5](#_Toc252380555)

[3.3 core/cpu/algorithm/mipsset 5](#_Toc252380556)

[3.4 core/cpu/monitor/enable 5](#_Toc252380557)

[4 Core 5](#_Toc252380558)

[5 Target Specific Data 5](#_Toc252380559)

[6 OS Specific Support 5](#_Toc252380560)

[7 Algorithm 6](#_Toc252380561)

[7.1 Algorithm Interface 6](#_Toc252380562)

[7.2 Enabling the Algorithm Log 6](#_Toc252380563)

[8 CPU Monitor 6](#_Toc252380564)

# Introduction

Cpu Dynamics is the name for DCVS as it applies to the CPU, and is necessary because the term DCVS (Dynamic Clock and Voltage Scaling) is generic, and can apply to many subsystems.

This document details the implementation of Cpu Dynamics. Although it mentions interfaces, it is not intended to be a technology team user guide.

# Overview

Cpu Dynamics is designed to be maximally reused, and easily ported to new targets. The high level view of the system is shown below.

Core

OS Specific Support

Target Specific Data

Algorithm

NPA Interface (/core/cpu)

The blocks shown are:

* NPA Interface
  + Resource named “/core/cpu”
  + Common Interface code
* Core
  + Code for managing algorithms
* Target Specific Data
* OS Specific Support
* Algorithm
  + This is interchangeable, and several are provided.

On WM targets the Algorithm and OS Specific Support blocks are split between the Kernel and the OAL Layer as shown below.

OAL OS Specific Support

Kernel OS Specific Support

Algorithm – Kernel

Algorithm - OAL

Interrupt

ioctl

# NPA Interface

The NPA interface block defines how clients interface with DCVS. This is a collection of resources (a node) that allows appropriate controls and inputs into the system. The individual resources in this node are described in this section.

## /core/cpu

The /core/cpu resource provides the main interface to CPU Dynamics, and is described in CoreCpu.h. It provides the three client interfaces shown below.

### REQUIRED

Clients requests of this type will be summed, and will define a MIPS floor. The system MIPS will not be allowed to go below this level.

This client type is useful for fixed, known loads, or loads of high priority that cannot afford any of the typical latency associated with Dynamic Algorithms.

### IMPULSE

Impulse requests affect the system immediately by forcing the system MIPS to the maximum value. The system MIPS are then allowed to drop based on the dynamics of the present algorithm.

This client type is useful for user inputs such as keypresses which require high responsivity, but do not have a predefined end of the workload.

### ISOCHRONOUS

Isochronous workloads are defined by repeated, generally consistent workloads of uncharacterized level. There is a clear starting point for each iteration, and a workload deadline is known at that time. There is also a clear stopping point for the workload.

Isochronous clients are intended to be learned as the load is repetitively executed. Initially the system will be driven to maximum performance during the active workload, but as the workload is characterized, the system will be allowed to go to a wider range of frequencies that still meet the performance requirements. This tuning system is NOT yet implemented.

A Good example of an Isochronous client is video playback. The load is uncharacterized since it differs on a per input basis, and it is roughly similar from frame to frame.

## /core/cpu/algorithm/disable

This resource takes only a REQUIRED is initialized to 0, meaning enable. This value could be set to 1, disabling the algorithm. It is NOT recommended except during testing operation. When this resource transitions from 0 to 1, the MIPS value requested in the system will be set to maximum.

## /core/cpu/algorithm/mipsset

This resource takes only a REQUIRED client, and is only relevant when the resource **/core/cpu/algorithm/disable** is set to 1, thus disabling the algorithm. This resource takes values in MIPS. It expects only ONE client, and simply uses the last request as the system MIPS setting.

## /core/cpu/monitor/enable

Idle Monitoring, and the associated logging is a separate operation from CPU Dynamics. It is provided here primarily to give tech teams necessary insight into system loading. When a REQUIRED client sets this value to 1, the monitoring subsystem will be enabled, and will begin logging data to the “CPU Monitor” log.

The “CPU Monitor” log is disabled by default, and must be manually enabled for any data to be actually captured.

# Core

# Target Specific Data

# OS Specific Support

# Algorithm

## Algorithm Interface

## Enabling the Algorithm Log

The algorithm log captures information about the running state of the current algorithm. It is named “CPU Dynamics”. This log is disabled by default, and must be manually enabled for data to be captured. The log can be enabled in two ways. T32, Application settings ini file.

# CPU Monitor