Выпускная квалификационная работа бакалавра

Политика регулирования частот центрального процессора в ядре Linux для приложений реального времени

Глаз Роман Сергеевич

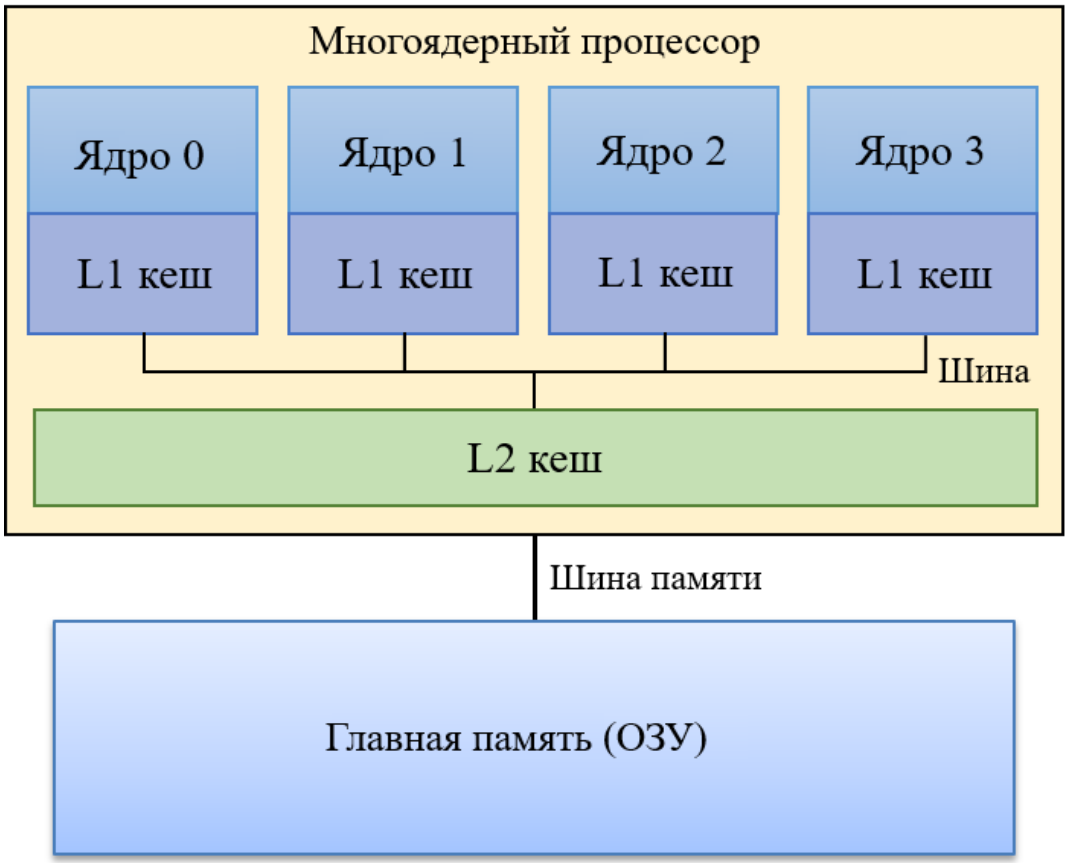
Московский физико-технический институт
Кафедра микропроцессорных технологий в интеллектуальных системах управления

20 июня 2024 г.

Актуальность

1. Производительность современных ЦП ограничена производительностью системы памяти.
2. Память: несколько уровней кешей и ОЗУ.
3. Современные операционные системы (Linux, FreeBSD) не учитывают задержки памяти при регулировании тактовых частот ЦП и планировании задач, из-за чего возможен неоправданный рост тактовых частот и энергопотребления.
4. Для приложений реального времени важно избегать деградации производительности при возможном наименьшем энергопотреблении ядер ЦП.

Пример организации памяти



Пример организации памяти в устройстве с многоядерным ЦП

Цели работы

1. Разработать модель производительности ядер ЦП с учётом поведения и характеристик устройств памяти (кеши ЦП, ОЗУ).
2. Разработать способ применения модели в ядре операционной системы Linux, реализовать политику регулирования частот ядер ЦП.

Задачи

1. Реализовать в ядре операционной системе Linux (версии 6.1):
   - 1.1 Набор драйверов, позволяющих изменять тактовые частоты ЦП в эмулируемой системе архитектурным симулятором Gem5.
   - 1.2 Возможность сбора статистики микроархитектурных событий ядер ЦП, связанных с работой кешей, для архитектуры ARM64.
2. Разработать модель производительности ядер ЦП:
   - 2.1 Спроектировать теоретическую модель производительности ЦП.
   - 2.2 Реализовать набор бенчмарков, подходящих для построенной модели, и собрать для них статистику микроархитектурных событий при различных частотах ЦП.
   - 2.3 Найти коэффициенты модели на основе снятых данных.
3. Реализовать политику регулирования тактовых частот ядер ЦП в ядре Linux, используя разработанную модель.

Обзор существующих решений

1. В ядре Linux неявным образом предполагается линейная зависимость производительности ядра процессора от его тактовой частоты:
   - 1.1 В подсчёте утилизации потоков исполнения и ядра ЦП.
   - 1.2 В регулировании тактовых частот ядер ЦП, в том числе на основе подсчитанной утилизации.
2. Существует ряд альтернативных моделей производительности ЦП, которые, как правило:
   - 2.1 Используют данные, которые нельзя получить в режиме реального времени.
   - 2.2 Не учитывают наличие возможности регулирования тактовых частот ЦП.
   - 2.3 Применимы только для конкретных приложений (отсутствует обобщение).

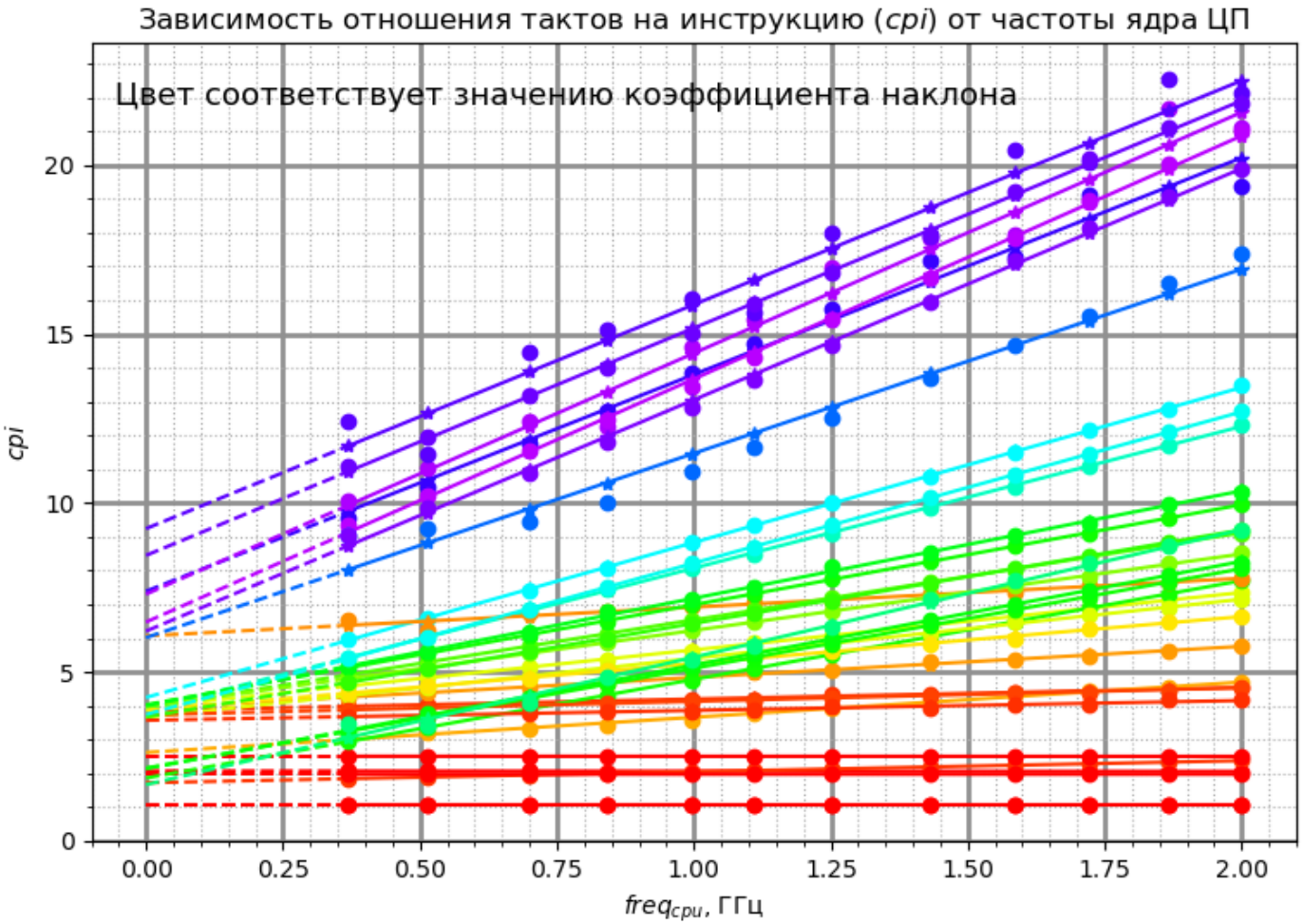
Исследование и построение решения

Построена следующая теоретическая модель производительности (для системы с 3-мя уровнями кешей):

$$cpi = cpi_{L1/L2} + \alpha_{mem}^{par} \cdot (\beta_1 \cdot (l2pi - l3pi) + \beta_2 \cdot l3pi) \cdot freq_{cpu} \quad (1)$$

1. cpi – количество затраченных тактов на инструкцию.
2. $l2pi$ и $l3pi$ – количество промахов в кеши 2-ого и 3-ого уровней на единицу инструкции.
3. $freq_{cpu}$ – тактовая частота ядра процессора.
4. α_{mem}^{par} – динамический параметр, характеризующий параллелизм обращений ядра ЦП в память, β_1 и β_2 – константы системы памяти.

Сбор и обработка данных



Механизмы, реализованные в ядре Linux

1. Драйверы *clk* и *cpufreq* для поддержки изменения тактовых частот ядер ЦП в Gem5.
2. Сбор статистики микроархитектурных событий для каждого ядра ЦП и потока исполнения в режиме реального времени и подсчёт их утилизации.
   2.1 При смене контекста потока исполнения, ротации счётчиков микроархитектурных событий и по таймеру.
3. Выбор тактовых частот на основе собираемой статистики (как по ядрам ЦП, так и по потокам исполнения).
   3.1 При пробуждениях ядра процессора, при миграциях потоков исполнения на другие ядра ЦП и по таймеру.
4. Политику выставления тактовых частот на основе голосования выбранных частот по ядрам ЦП.

Результаты и преимущества решения

Преимущества решения:

1. Учёт параметров, зависимых от рабочей нагрузки (кол-во обращений в ОЗУ, высокие уровни кешей).
2. Введение параметра параллелизма обращений в память.
3. Независимость модели от специфики работы конвейера ядра ЦП.
4. Возможность использования в ЦП с технологией многопоточных ядер.

Результаты:

1. Снижение энегопотребления синтетического приложения реального времени на 6.4% при сохранении производительности.

Благодарю за внимание!