

Десятая независимая научно-практическая  
конференция «Разработка ПО 2014»

23 - 25 октября, Москва



# Метод оптимизации энергоэффективности приложений компилятором

Токарь Илья

# Энергоэффективность

- энергэffективностъ =  $\frac{\text{производительность}}{\text{потребление}}$
- тах энергэffективностъ  $\neq$  тах производительностъ
- тах энергэffективностъ  $\neq$  min потребление

# Задача

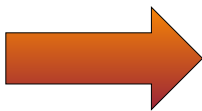
- Максимизировать энергоэффективность
- Без значительных потерь производительности
- За разумное время компиляции

# Решение

- Разные участки кода работают на разной частоте :

## DVFS - dynamic voltage and frequency scaling

```
Foo() //1333 МГц
{
    /* some code */
}
Bar() //1333 МГц
{
    /* some code */
}
```



```
Foo() //1333 МГц
{
    set_freq(1333)
    /* some code */
}
Bar() //800 МГц
{
    set_freq(800)
    /* some code */
}
```

# Выбор регионов понижения частоты

- Нелинейная зависимости производительности от частоты : память (кэш-промахи)
- Нелинейная зависимость потребления от частоты :

$$P = V_f^2 * f, \text{ можно понижать напряжение}$$

# Задача выбора регионов понижения частоты

- Достаточно приближённого решения
- Задача многокритериальная оптимизации : энергоэффективность, производительность

- $\max_{f_1..f_n} \frac{P(f)}{E(f)}$

- $P \geq P_{max} * (1 - \alpha)$

- $t_{comp} \leq t_{comp-max}$

# Генетический алгоритм

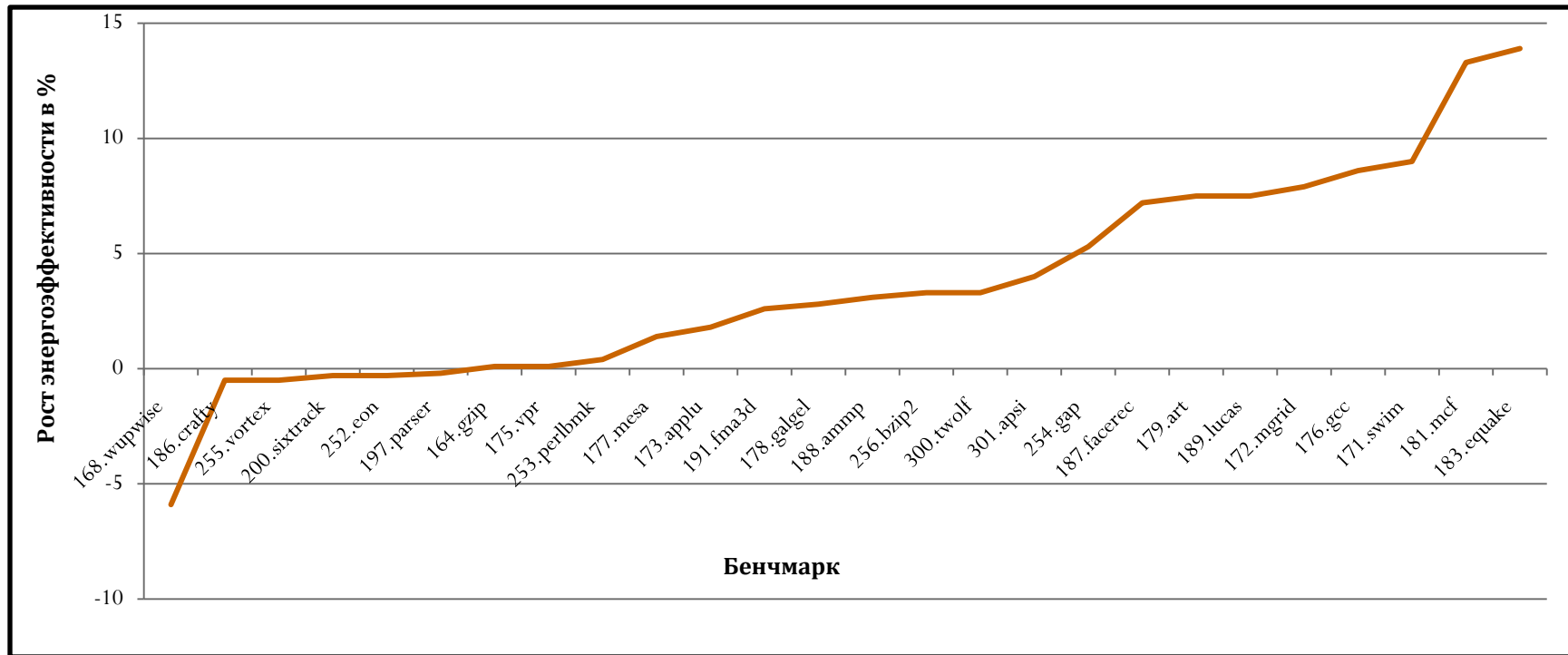
- NSGA II
- Целевые функции  $1 - \frac{E}{E_{max}}$ ,  $2 - \frac{1}{\exp(P_{max} - P)}$
- E,P оцениваются профилированием
- Время компиляции зависит от числа итераций, задаётся параметром
- Геном : вектор частот каждой функции
- Оператор мутации : изменение одной из частот
- Оператор кроссовера : вектор средних частот

# Реализация

- Собрать данные профилирования – perf record
- Найти функции для исполнения на пониженной частоте – реализация алгоритма
- Инструментировать код – проход в GCC после встраивания (inline) функций



# Результаты



# Результаты

- Хорошие результаты
- (до 13,5% роста энергоэффективности в отдельных тестах, 2% float geomean)
  - Несколько больших функций
  - Memory-bound
- Плохие результаты
  - Одна функция
  - Много маленьких функций
  - Неточные данные профилирования
  - Накладные расходы на переключение частоты

# Выводы

- Предложен генетический алгоритм оптимизации энергоэффективности приложений для мобильных платформ компилятором
- Алгоритм хорошо себя показал на ряде задач
- Проведён анализ применимости
- Дальнейшее развитие (интеграция с операционной системой, jit компиляторами, циклы как регионы понижения частоты, расстановка на графе вызовов)