

Санкт-Петербургский государственный университет  
Математико-механический факультет  
Кафедра системного программирования

**Дипломная работа**  
**«Оптимизации генерации кода в JIT-  
компиляторе виртуальной машины  
Java»**

Научный руководитель  
Куксенко С.В.

Рецензент  
Салищев С.И.

Выполнил  
Проничкин Дмитрий 544гр.

2008

# Актуальность работы

- Постоянное совершенствование микроархитектуры процессоров
- Особенности новой микроархитектуры Intel Core
- Увеличение важности производительности front-end процессора
- Работа front-end влияет не только на производительность, но и на флуктуацию при ее измерении
- Отсутствие работ, учитывающих данные особенности микроархитектуры Core, даже в Intel Compiler

# Результаты

- Разработаны идеи и эвристики для оптимизации линейаризации и выравнивания кода
- Опробована схема удаления ветвлений в коде
- В генераторе кода JIT-компилятора виртуальной машины Apache Harmony реализованы улучшения линейаризации и выравнивания кода, удаления ветвлений
- Получен прирост производительности на микротестах и популярных бенчмарках, таких как SciMark (Monte Carlo – прирост **60%**)
- Увеличена стабильность метрик производительности
- Изменения приняты и интегрированы в Apache Harmony

# Apache Harmony

- Открытая реализация виртуальной машины Java
- JIT-компилятор Jitrino.OPT – оптимизирующий компилятор с возможностью профилировки и перекомпиляции
- Особенности front-end микроархитектуры Core не учтены

# Выравнивание кода

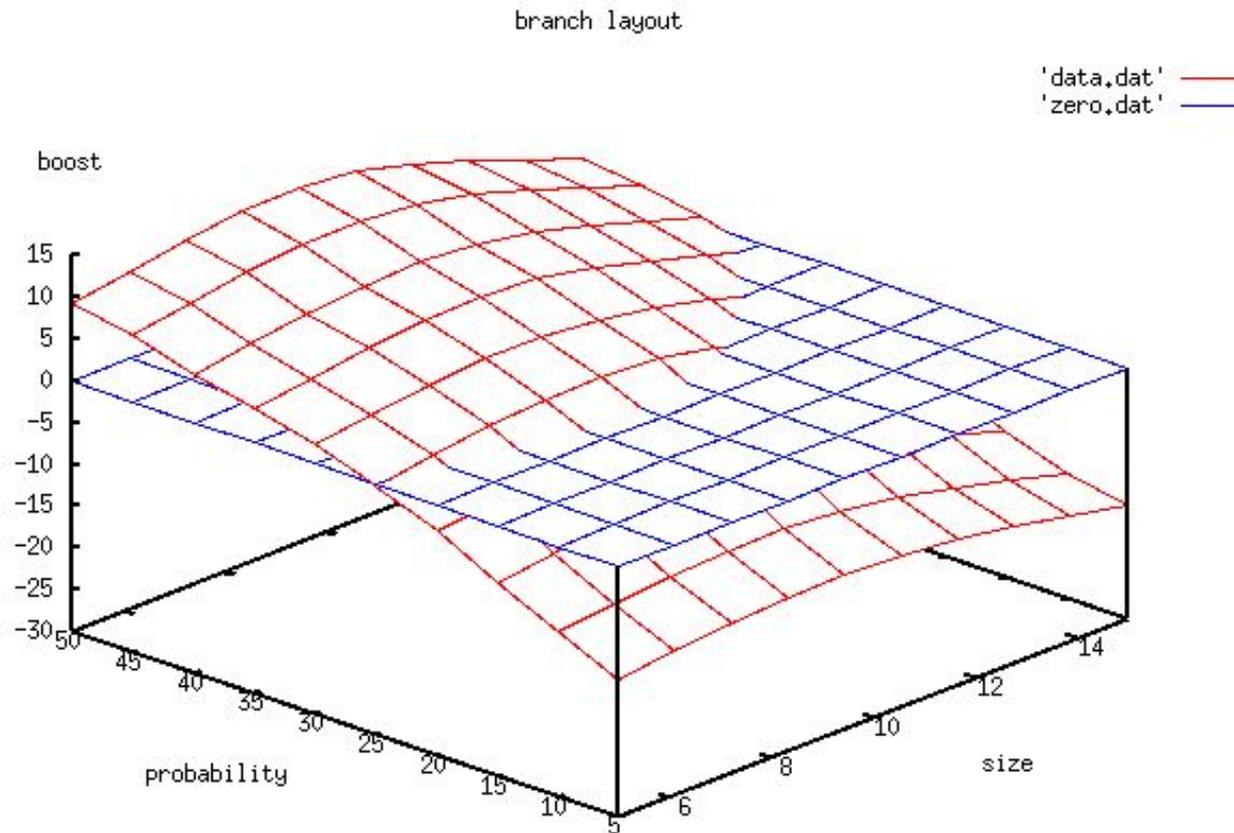
- Отсутствие trace cache по сравнению с микроархитектурой NetBurst
- Линия выборки (fetch line) 16 байт
- Особенности предсказателя переходов
- Дополнительная возможность процессора – loop stream detector
- Все эти особенности учтены и разработана эвристика для выравнивания кода

# Линеаризация кода

- Расположение базовых блоков графа потока управления в линейном порядке
- Алгоритм “bottom-up”, имеющий много свойств, положительных для front-end процессора

- Найдены возможности для улучшения под микроархитектуру Core

- Разработана эвристика для оптимизации алгоритма



# Удаление ветвлений

- Все современные микроархитектуры – конвейерные суперскалярные
- Неправильно предсказанный условный переход приводит к задержке работы конвейера
- Иногда можно удалить ветвление, заменив его на более сложные, но линейные вычисления
- Схема удаления ветвлений опробована на бенчмарке SciMark, получен значительный прирост производительности