

# Introduction to Library Generators

---

CS 498: Compiler Optimizations  
Fall 2006

University of Illinois at Urbana-Champaign



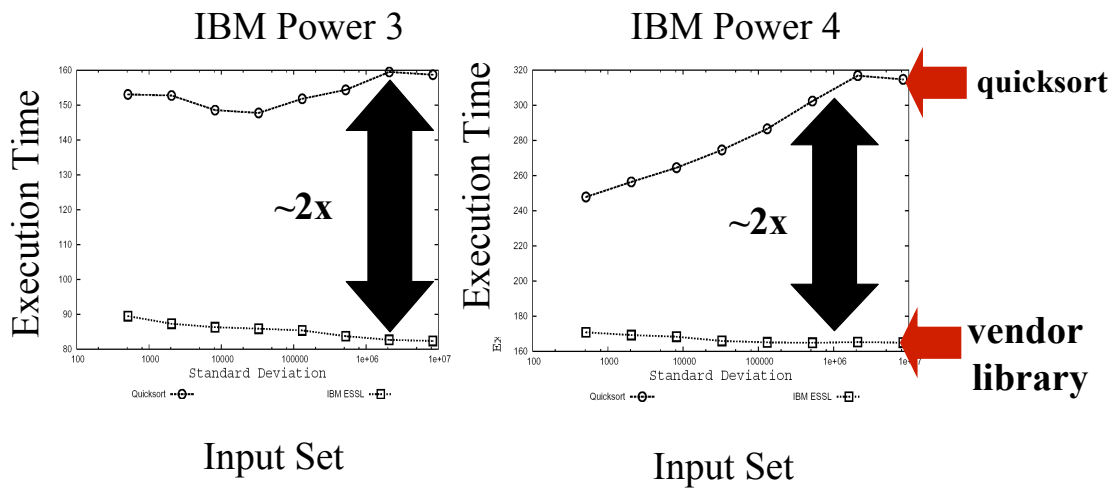
## Motivation

---

- ◆ Program optimization is a difficult task
  - Computers are becoming very complex
  - Interactions between hardware and software are difficult to understand
  - Difficult to choose the best transformations or parameters

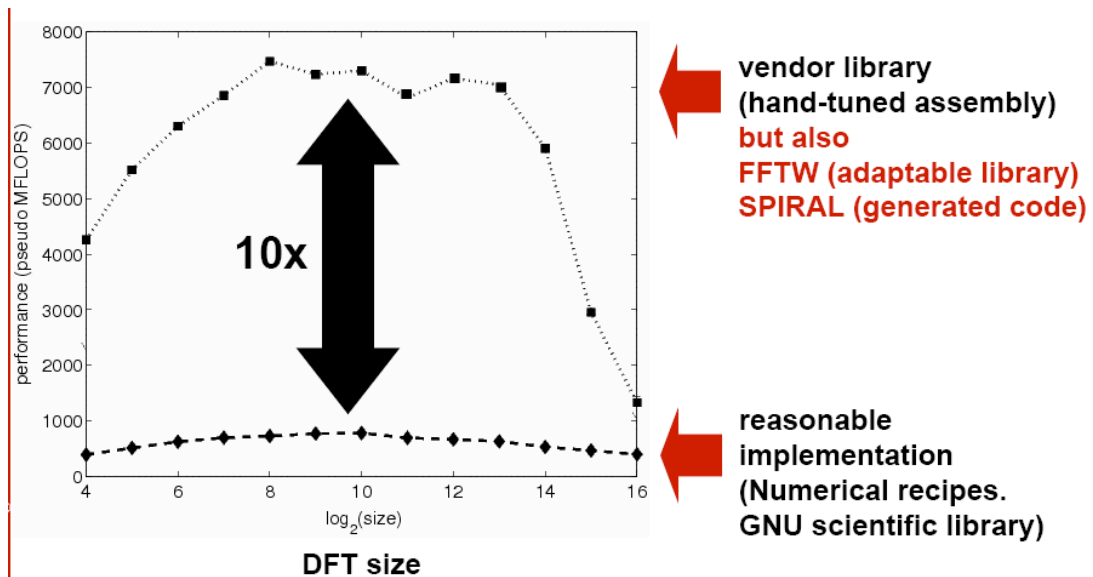


# Compilers vs. Libraries in Sorting



3

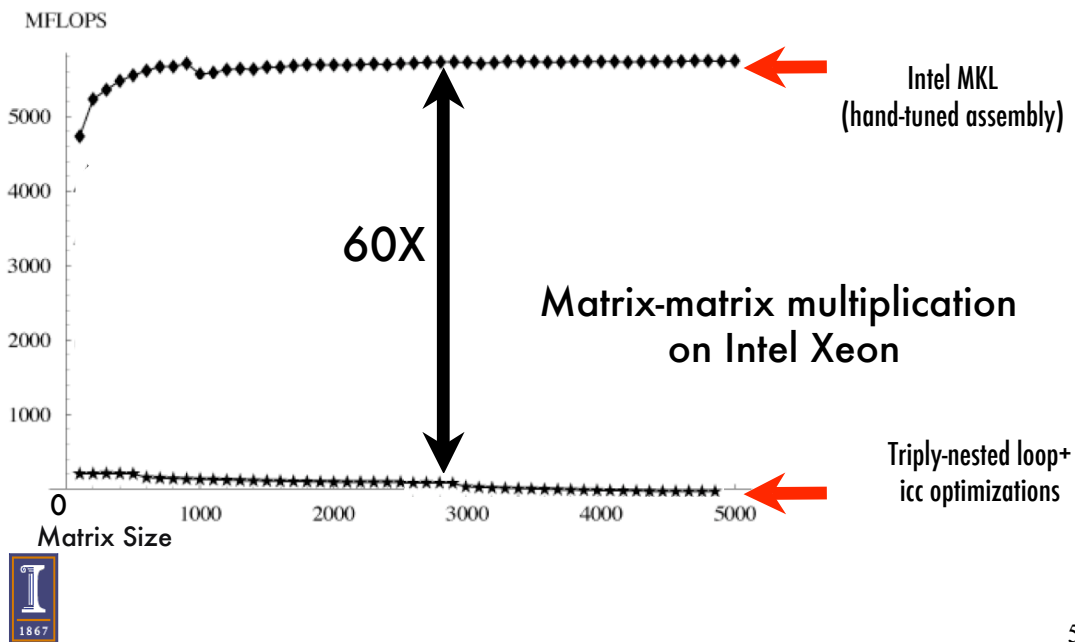
# Compilers versus libraries in DFT



4

# Compilers versus libraries in Matrix-Matrix Multiplication (MMM)

---

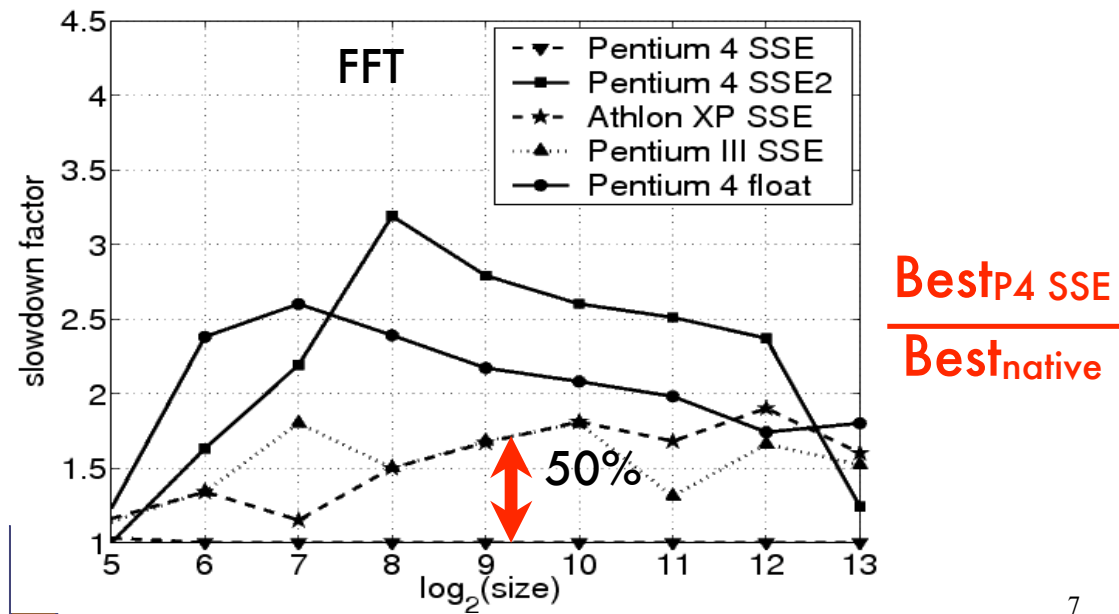


## Libraries and Productivity

---

- ◆ Building libraries is one of the earliest strategies to improve productivity.
- ◆ Libraries are particularly important for performance
  - High performance is difficult to attain and not portable.

# Porting hand-optimized code to a new platform



7

## Solution: Library Generators

- ◆ Emerging new field.
- ◆ Goal is to automatically generate highly efficient code for each target machine.
- ◆ Typically, a generator is executed to empirically search the space of possible algorithms/implementations.
  - No need to tune the library to the architectural characteristics of a new machine



8

# Library Generators

---

- ◆ Automatic generation of libraries would
  - Reduce development cost
  - For a fixed cost, enable a wider range of implementations and thus make libraries more usable.
- ◆ Advantage over compilers: Can make use of semantics
  - More possibilities can be explored.
- ◆ Disadvantage over compilers: Domain specific.



9

## Library Generators (Cont.)

---

- ◆ Examples:
  - In linear algebra: ATLAS, PhiPAC
  - In signal processing: FFTW, SPIRAL
- ◆ Library generators produce a pre-defined set of algorithms.
  - Exception: SPIRAL accepts formulas and rewriting rules as input.



10

## Library Generators (Cont.)

---

- ◆ At installation time, Library Generators apply empirical optimization.
  - They search for the best version in a set of different implementations
  - But ... number of versions is astronomical.
    - Heuristics are needed.



11

## Library Generators (Cont.)

---

- ◆ LGs must output C code for portability.
- ◆ Uneven quality of compilers =>
  - Need for source-to-source optimizers
  - Or incorporate in search space variations introduced by optimizing compilers.



12

# Library Generators (Cont.)

---

