



# FFTE: A High-Performance FFT

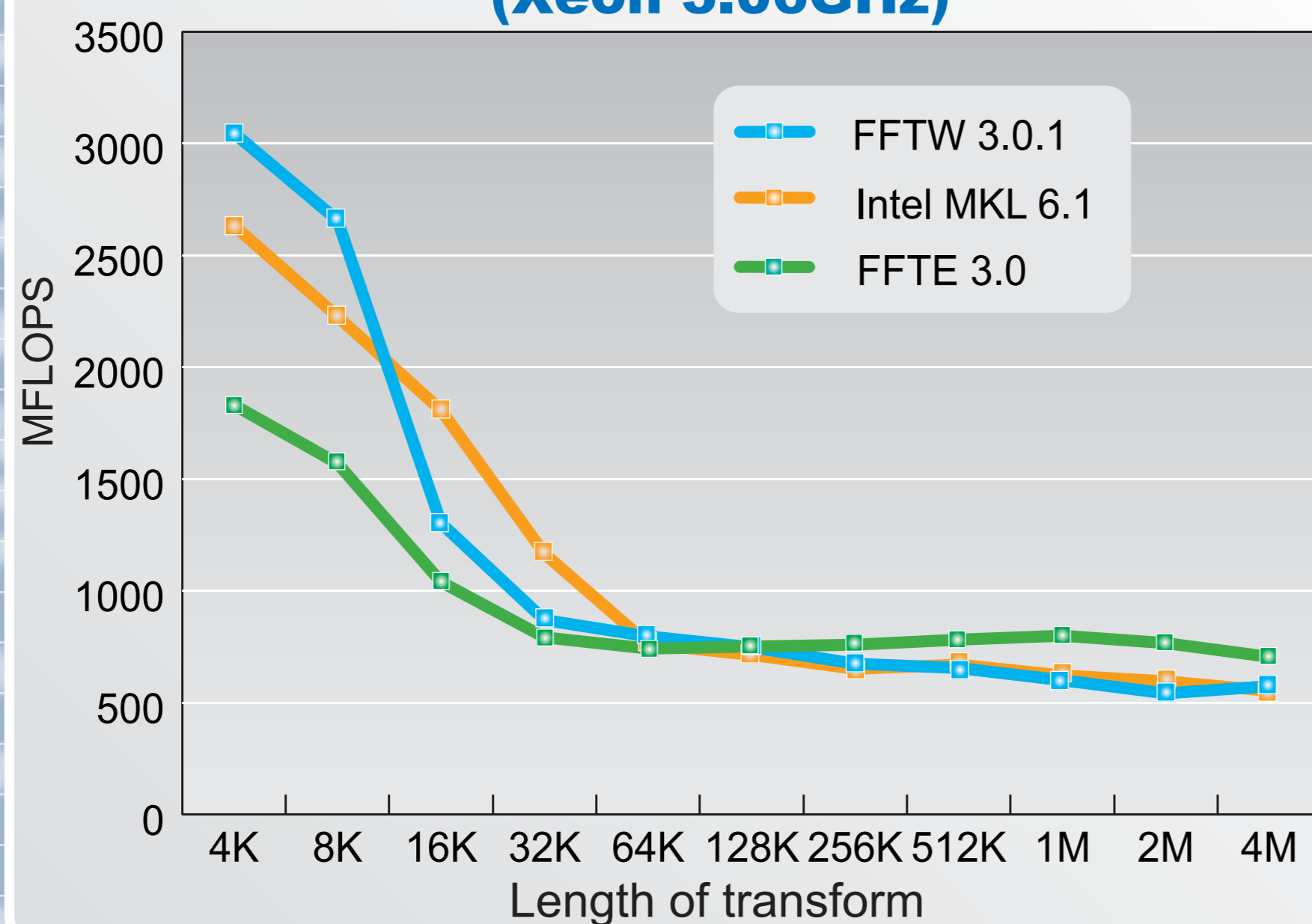
## Introduction

- ▶ **FFTE** is a Fortran subroutine library for computing the Fast Fourier Transform (FFT) in one or more dimensions.
- ▶ It includes complex, mixed-radix and parallel transforms.
- ▶ FFTE is typically faster than other publically-available FFT implementations, and is even competitive with vendor-tuned libraries.

## Features

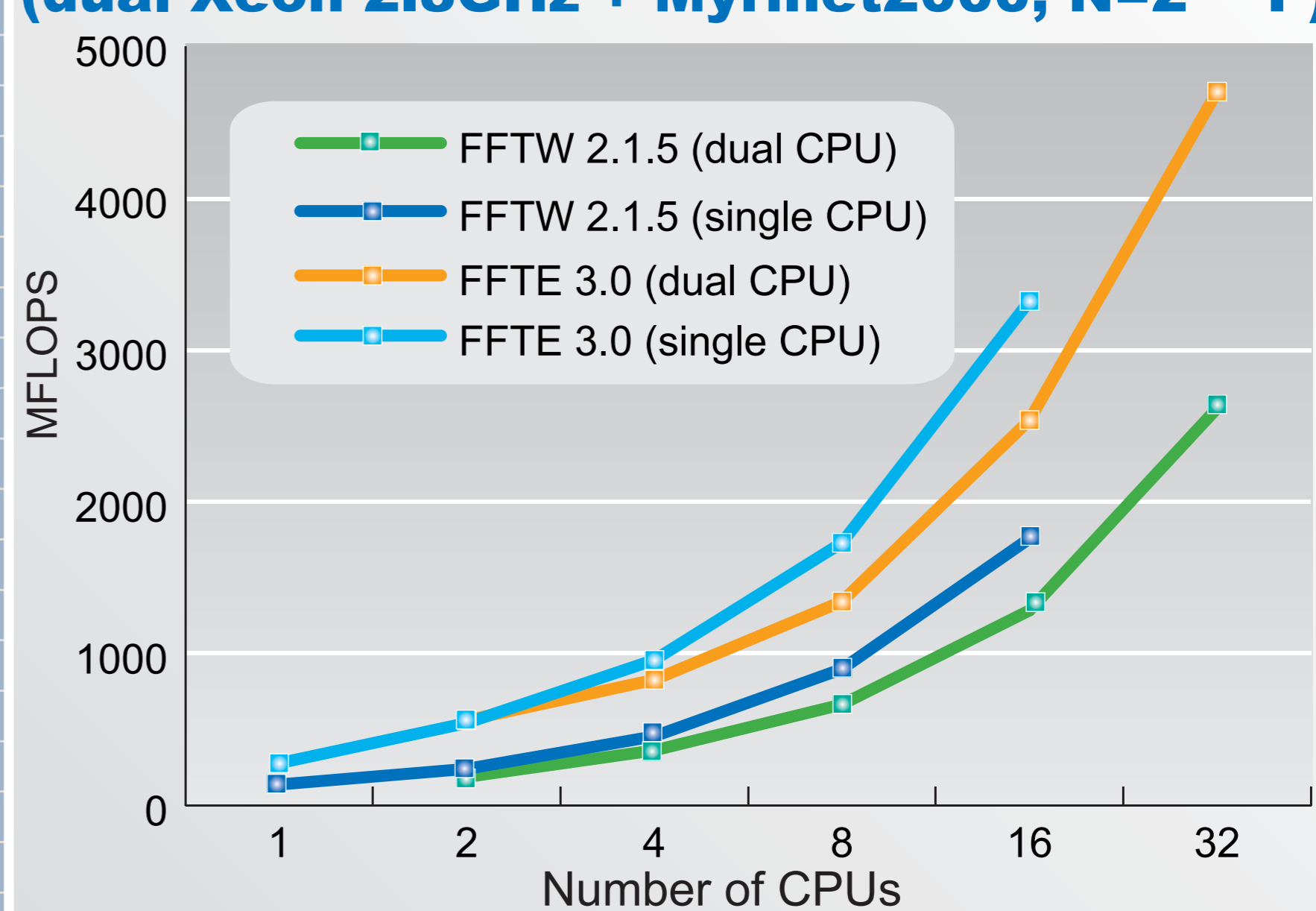
- ▶ Scope: Library of sequential / parallel FFT subroutines
- ▶ Target: Shared / Distributed memory computers (OpenMP and MPI)
- ▶ Goals
  - High-performance
  - Ease of use
  - Portability
- ▶ Design
  - Performance: One goal for large FFTs is to minimize the number of **cache misses**
  - Ease of use: Routine interfaces are similar to sequential **SGI SCSL** or **Intel MKL** routines
  - Portability: Fortran77 + OpenMP + MPI
- ▶ Approach
  - Many FFT routines work well when data sets **fit into a cache**.
  - When a problem size exceeds the cache size, however, the performance of these FFT routines **decreases** dramatically.
  - We combine the multicolumn FFTs and transpositions to **reduce** the number of cache misses.

**Performance of 1-D FFTs  
(Xeon 3.06GHz)**



- For  $N \geq 128K$  the FFTE is faster than both the FFTW and the Intel MKL.
- The performance of the FFTE remains at a high level even for a large problem size, owing to cache blocking.

**Performance of Parallel 3-D FFTs  
(dual Xeon 2.8GHz + Myrinet2000,  $N=2^{24} \cdot P$ )**



- For 32 CPUs, the FFTE runs about 1.8 times faster than the FFTW.
- Although the FFTW requires three all-to-all communication steps, the FFTE requires only one all-to-all communication step.