

# Paving the Way: Multicore and Multi-Multicore

Matt Reilly

Chief Engineer

SiCortex, Inc

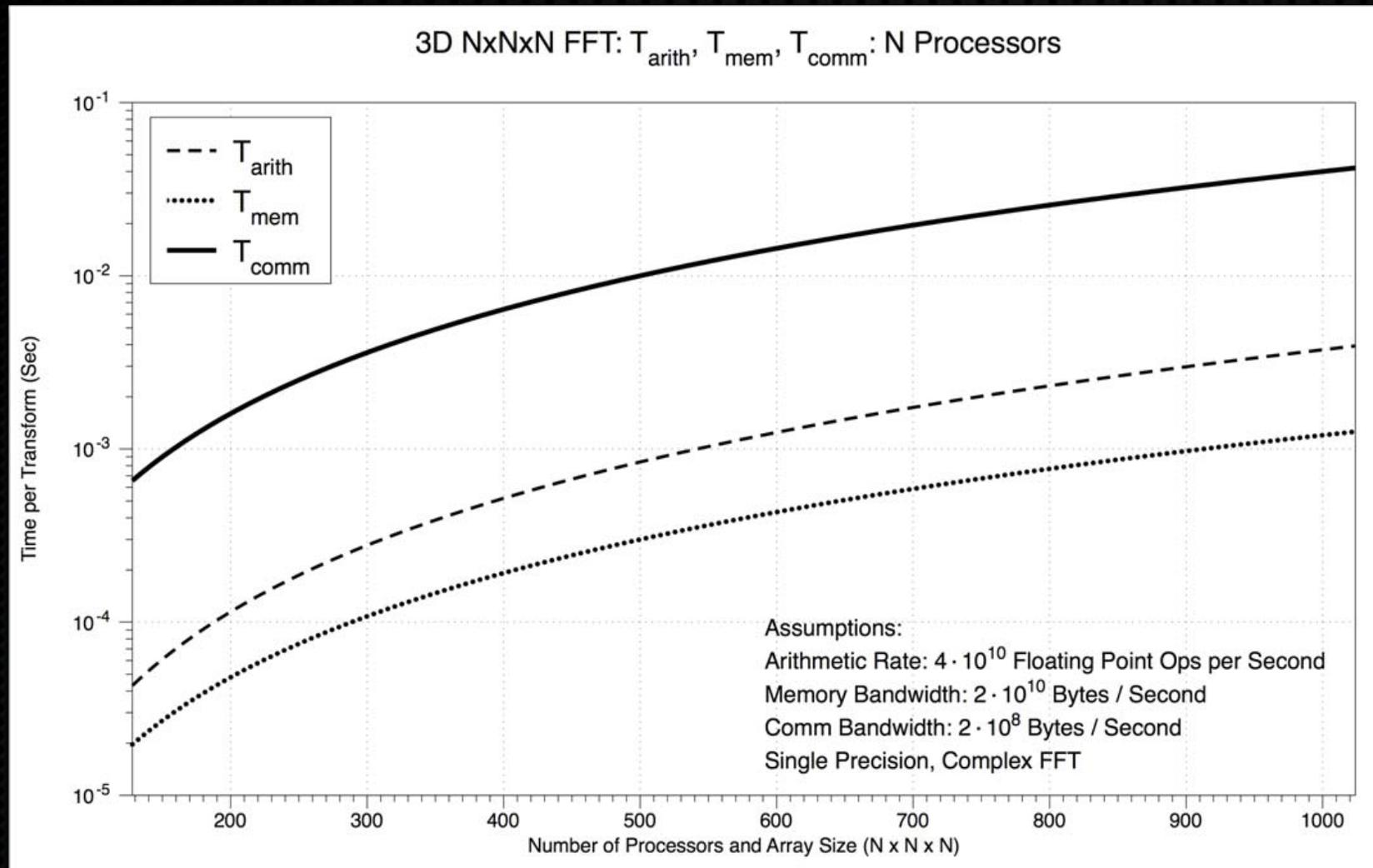


SiCortex

# MultiCore and Multi MultiCore

- Multicore isn't enough: There are real time constraints that will demand greater execution and memory rates than can be achieved on one chip.
- We need to develop new models for programming, beyond PThreads, and MPI.
- We need a new generation of “Big N” enthusiasts/geeks.

$$\text{When } T_{\text{sol}} = T_{\text{arith}}/N + T_{\text{mem}}/N + f(N)T_{\text{comm}}$$



# Programming Models

- “Hide it all” libraries
  - Attractive in many domains
  - Requires a few wizards/architects
  - Be prepared to throw stuff away
- “Expose it all” libraries
  - Small devoted priesthood
- “Hide it all” languages
  - Destined to fail: they don’t recognize the NUMA hierarchy”
- “Realistic” languages
  - Growing presence
  - Interesting approaches to “reality mapping”

# Why Johnny Can't Write Parallel Programs

- Parallel programming is relegated to “afterthought” in curricula.
  - Time for us to exercise a little influence
  - But please, save us from “Industry wide coordinated initiatives”
- Few “playgrounds” available.
  - This is easy: Install accessible parallel machines *everywhere*
  - Big national monuments are not the answer

# The Take-Away

- Successful Multi-Multicore adventures will reverse the model:
  - $T_{sol} = T_{arith}/N + T_{mem}/N + T_{IO} + f(N)T_{comm}$
- Current programming techniques are inadequate.
- We need more competent “high processor count” parallel programmers.