

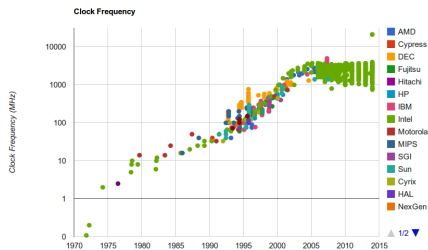
指導教員	田浦健次郎 教授	研究場所	工学部 2 号館
------	----------	------	----------

# Big Computation, Big Dataの世界を切り開こう!

<http://www.eidos.ic.i.u-tokyo.ac.jp/> (紹介文 → <http://bit.ly/1PmZPYz>)

## 1 はじめに

現在、自然科学や工学のありとあらゆる分野で、コンピュータシミュレーションが必須の道具となっています(生体分子の働きの理解, 気候変動の予測, 新材料の設計, 銀河形成の過程の理解, など). また, 自然言語処理や機械学習など, 高度な情報処理技術の近年の進展は, 巨大なデータを高速に処理する技術により支えられています. 計算機は最近まで, ソフトウェアを大して書き換えなくても, CPUの逐次処理性能, 中でも周波数の向上によって高速化されてきましたが, 近年では周波数の向上は明確に頭打ち(Dennardスケール則の終焉)となり, 高性能化は並列処理(ソフトウェアの並列化によって, 複数CPUや複数の計算機ノードを活用すること)達成されなくてはならなくなりました.



CPUの周波数向上の頭打ち (source: <http://cpudb.stanford.edu/>)

2015年現在, 最高クラスの計算機は数万から数十万の計算ノードを結合しています.

## 2 研究テーマ

我々の追求したい研究テーマの中心は, 高性能な並列計算を誰もが行えるようにするための, ソフトウェアです. 目指すは並列処理の「高性能」と「高水準」の両立です. そのために, 新しい並列プログラミング言語や, プログラミングなしで並列処理が行えるツールを設計・実現することを中心的な目標としつつ, 言語処理系(コンパイラ), 実行時システム, オペレーティングシステム, ファイルシステム, シェル, ユーザレベルのツールなどあらゆるレイヤにまたがった実現方法を研究しています. 以下は進行中の研究プロジェクトの一部です.

**大規模並列環境での自動負荷分散・分散共有メモリ:** 数千台の計算ノードをネットワークで結合した環境(数千・数万CPU)で, 処理系が自動的に負荷を分散し, かつ全てのデータを単一の計算機にあるかのように扱える並列プログラミング処理系に関する研究です.

**MassiveThreads 超軽量スレッドライブラリ:** 前項の土台となる, 通常のスレッドよりも二桁高速にスレッドを作れるスレッドライブラリです (<https://github.com/massivethreads/>).

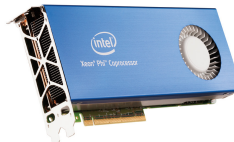
**タスク並列処理のSIMD化コンパイラ:** 現在のCPUの高速化・低消費電力化に必須の技術であるSIMD(1命令で, 複数のデータに対して同じ演算を施す)命令の利用範囲を, 従来のループから, 動的に作られるスレッドに広げ, 動的な並列処理の性能を大幅に高める研究です.

**不揮発メモリのためのファイルシステム:** 通常のメモリのように(CPUから直接load/store命令で)アクセスでき, かつHDD/SDDのような不揮発性(電源を切っても消えない性質)を持つメモリが登場しつつあります. これを, 耐障害性の必要な処理の高速化に最大限活かせるようにするための, OS, ファイルシステムのインタフェースと実装方式に関する研究です.

**並列シェル・データ処理環境:** 既存のプログラムを組み合わせた大規模データ処理を, クラスタ, クラウドなど, 所構わずどこでも行えるようにした, 並列処理ツールです (<http://www.logos.t.u-tokyo.ac.jp/gxp/>).

研究成果は論文として発表するだけでなく, 実用レベルのソフトウェアとして構築し, オープンソースソフトウェアとして公開することを学生にも推奨しています.

## 3 計算機環境



東大や東工大の最新のスパコン, 研究室内の36-64コアのマルチプロセッサやIntel Xeon Phi メニーコアプロセッサなど, 並列処理の力, 楽しさを味わうのに最適な計算機環境があります.

## 4 どのような人が育ってほしいか?

「自分で世界を広げられる人」「自分で考える人」になってほしいと思います. 必要な情報, 多くの問いに対する表面的な答えが検索一つで見つかるこの時代に「考える輩」であり続ける事が一層重要になっています. プログラミングの上手下手は決して話の中心ではない. 逆に, プログラミングがうまいからといって奢らず, 「自分の知らない分野をどんどん自分で勉強・吸収して行ける人」になってほしいと思っています. 並列処理の応用は広大で, それをいくらかでも深く知り, 設計や実装を進めなくてはなりません. 決して短時間で叶うことではありませんが, 幅広い分野を自分で学んでいける姿勢が重要です.