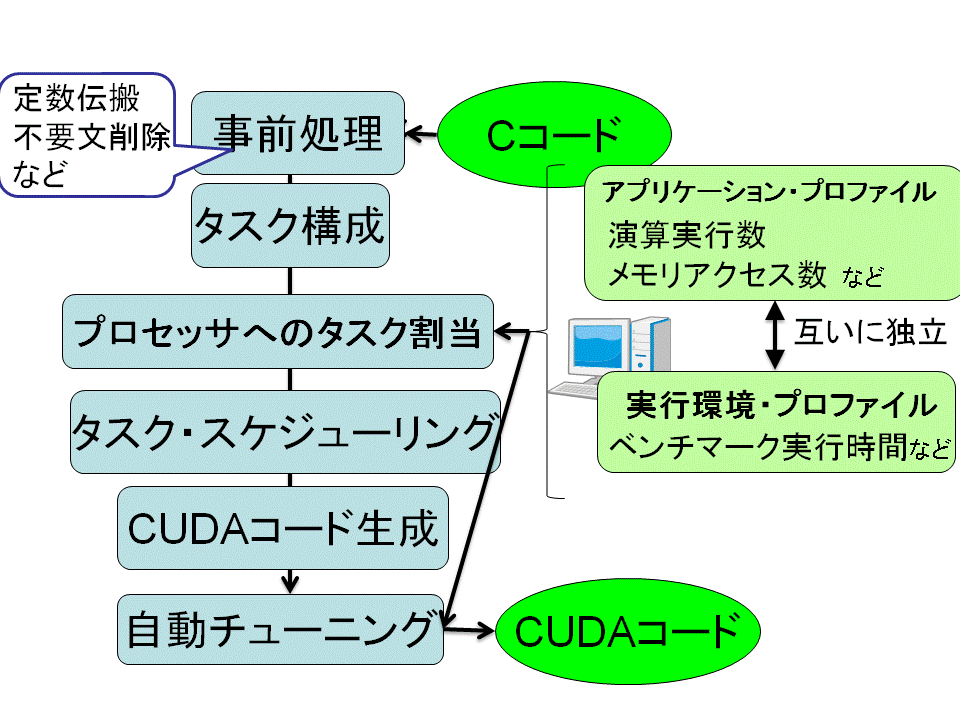
●研究内容・目的



新しいコンパイラ技術の創成

教授　杉野 暢彦

研究分野：コンパイラ、自動並列化、コード変換・最適化

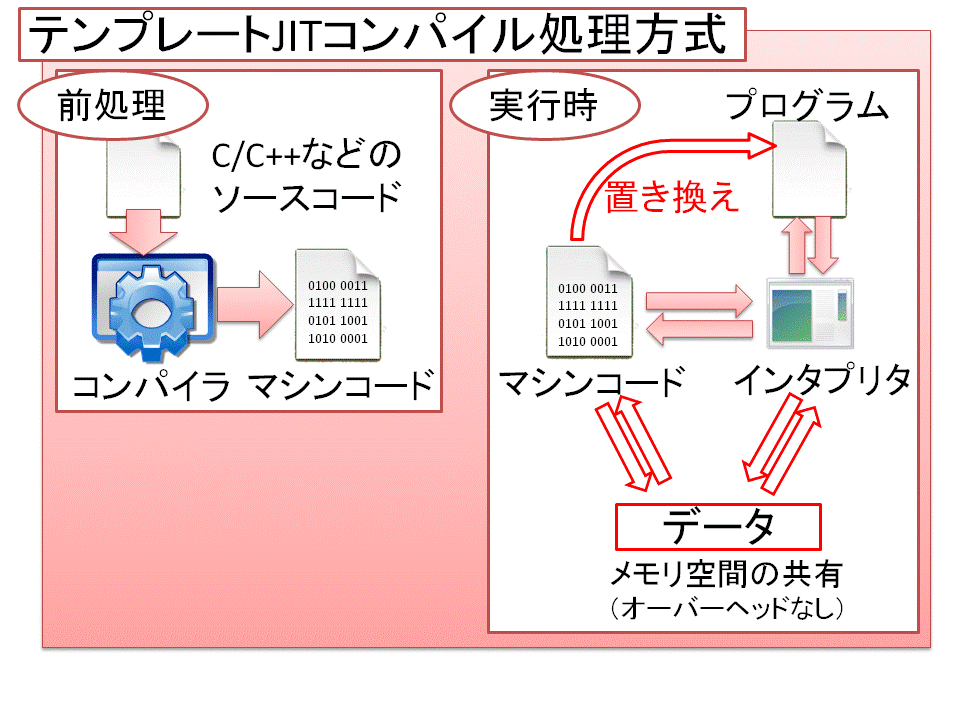
　汎用高性能プロセッサ及び埋込プロセッサ用の高効率・高性能コンパイラの開発をめざしている。プロセッサは年々複雑化し、それを利用したアプリケーションも大規模化・多様化してきている、現在のシステム開発においてはハードウェアだけでなく、ソフトウェアによりハードウェアリソースを如何に活用できるかが、システムの性能を大きく左右する重要な課題となっている。コンパイラを代表とする自動コード生成、自動並列化、コード最適化技術はその中でも鍵として注目されている。

●研究テーマ

１．GPGPU向けコンパイラ

　GPGPU向けにアプリケーションを開発するためには、アーキテクチャとCUDAを始めとする専用言語の知識が必要になる上に、プログラマは性能を引き出すために様々に試行錯誤を繰り返すことになり、あまり容易ではありません。そこで、C言語からCUDAへとコードを再構成するコンパイラを提案しています。提案コンパイラでは、入力コードをタスクに分割した後、 各タスクの性質に応じてCPU/GPUへの割り当てを行い、更にGPUアーキテクチャに合わせたチューニングを行い、実行効率を向上できます。また、自動中チューニングにおいて、消費エネルギーの低減を目指すことも可能です。

GPGPU向けソース（C）-to-ソース（CUDA）コンパイラ

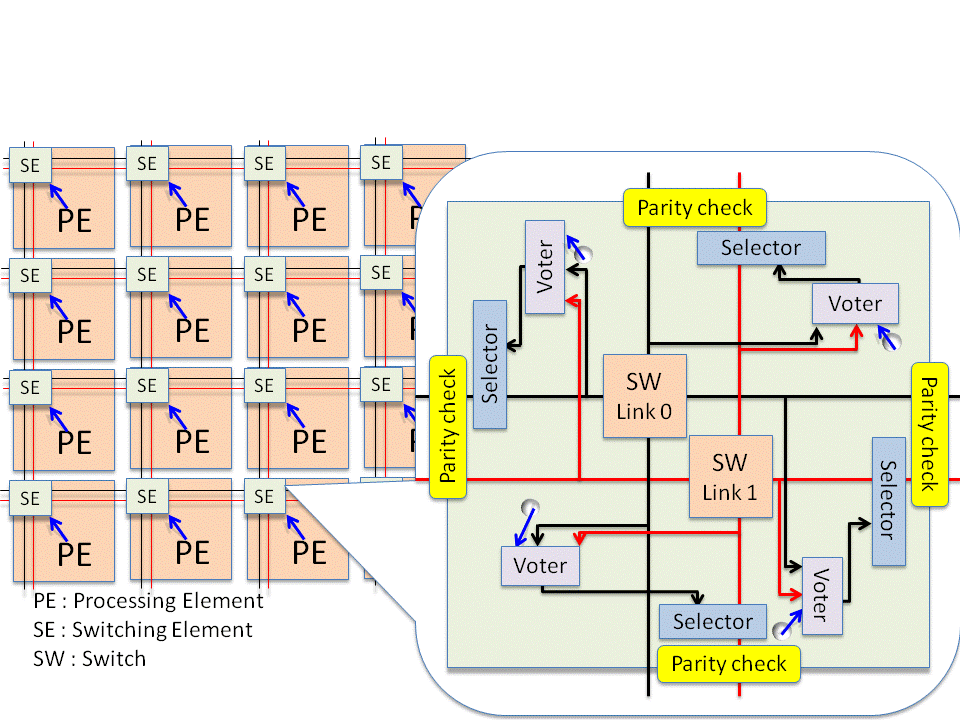


２．動的言語向け自動並列化JITコンパイラ

　携帯機器などのアプリケーションはpythonなどの動的言語スクリプトで書かれることが多く、通常、それらは仮想マシン上において、インタプリタ及び処理の高速化を指向したJIT（Just-In-Time）コンパイル方式の処理系により、逐次実行的に実装されています。しかしながら、この実装では「並列化」という概念が欠けており、携帯機器でもCPUがマルチコア化している現在では、並列処理により処理速度を向上させることも嘱望されています。そこで、動的言語処理系に「自動並列化」の概念を導入して、その１つの実装方法を提案しています。例題プログラムによる検証実験では、並列化による性能向上を確認しています。

動的言語処理系の自動並列化JITコンパイル方式

３．動的再構成可能プロセッサを用いた技術

　動的再構成可能プロセッサは、並列処理やパイプライン処理による高い演算能力を持つだけでなく、動的に回路構成情報であるコンテキストを変更できることにより、様々なアプリケーションに柔軟に対応可能であり、現在注目されている技術の１つです。この動的再構成可能プロセッサと汎用プロセッサを組み合わせて利用する場合に、効率的な処理を可能とするためには、与えられたプログラムのどの部分をコンテキストとして実現するか、また、実行中のどのタイミングでコンテキストを書き換えるか、など様々な要素を考慮する必要があり、今後の研究成果が期待されています。一方、動的再構成可能ユニットを改良して、ビット誤りや故障に強い高信頼なシステムの設計に応用できる技術についても提案している。

高信頼性システム向けの動的再構成可能ユニット

●教員からのメッセージ

　コンパイラ関連の研究では、プロセッサアーキテクチャに直接関係する最適化や自動並列化などの部分については、まだまだ手探りの部分もあります。実際にプログラミングすることも必要になりますが、何よりもブレークスルーを産む原動力は熱意と根気、それに多方面からものを捉えられる頭の柔らかさ、残された難しい問題に挑戦する元気のあるバラエティに富んだ学生の皆さんを歓迎します。また、教員はTAIST-Tokyo Techにも参加しており、海外の学生や研究者と積極的に交流したい学生も歓迎します。

●関連する業績，プロジェクトなど

1. 湯本厚史，杉野暢彦, “プロファイルに基づくGPGPUのためのソースコード変換コンパイラ,” ETNET2012（情報処理学会SLDM,EMB，電子情報通信学会CPSY,DC）（2012年3月）

2. R. Ikeda, etal. “An Automatic Parallelization Scheme Used in JIT Compilation for Dynamic Language Applications,” Proceedings of ICICTES 2012, （2012年3月）

3. K. Umehara, etal. “A Dynamically Reconfigurable Processor for Highly Reliable Systems,” Proceedings of ICICTES 2013, （2013年1月）