

XMLを用いたツール間連携に向けて

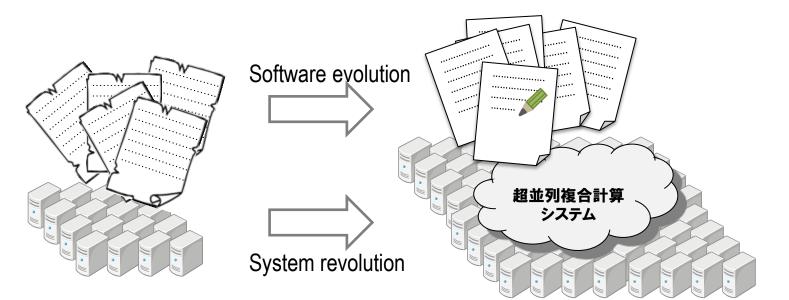
1st XcalableMP Workshop in 秋葉原 2013年11月1日@秋葉原UDX 6F

滝沢 寛之

<tacky@isc.tohoku.ac.jp>
東北大学大学院情報科学研究科

JST CREST「進化的アプローチによる 超並列複合システム向け開発環境の創出」

- ・ 背景:HPCシステムアーキテクチャの劇的変化
 - 大規模化、多様化、複合化(ヘテロ化)・・・
- ・目的:超並列複合計算システム向け開発環境
 - システムの進化に対応可能なアプリ開発方法の確立
 - → 既存のアプリケーション資産の有効活用



HPCT DITTELL



専門の異なるプログラマたちのチームワーク

- ペアプリ開発者(計算科学者)
 - 表現したいもの:アルゴリズムとプログラムの関係
 - → 正しい計算結果を出力するプログラムの作成
- ▲チューナー (計算機科学者)
 - 表現したいもの:プログラムとシステムの関係
 - → 対象システムで高速動作するプログラムの作成



デームワークを阻む壁



東北大学サイバーサイエンスセンター

- 長年の高速化支援実績

年	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012
件数	2	8	8	9	10	7	18	20	8	16	10	15	8	8	13	6
平均ベクトル化 性能向上比	1.9	46.7	4.5	2.5	1.6	2.2	6.7	2.9	1.3	2.9	33	9.4	381	47	16.2	19.7
平均並列化性能向上比	11.1	18.4	31.7	8.6	4.9	2.8	18.6	4.5	5.3	8.1	1.9	5.1	3.6	48	17.2	15.3

連携は難しい...

- 最適化作業中にアプリ開発者が新バージョンを作成した
- 最適化の結果,アプリ開発者が管理できないコードになった
- 新しいシステム向けにアプリを最適化し直さなければならなくなった



アプリ開発者とチューナーの上手な役割分担は?

→ 高速化のためのアプリコード改変を最小限に抑制

コード最適化の抽象化



ディレクティブや変換スクリプト

- アプリコードを編集する代わりにディレクティブを追記
- アプリコードとは別に変換スクリプトを記述

・主にループ最適化の研究事例

- CHiLL (Univ. Utah)
- POET (Georgia Tech)
- LoopTool (Rice Univ)
- Orio (OSU)
- ROSE (CHILL, POET, and ROSE)
- ABCLibScript (Univ Tokyo)

名前がついているような 基本的なループ最適化 はすでに提供済み

提供されていない変換には対応困難 記述自体が特定のシステムに特化する傾向





2D Convolution: CUDA-CHiLL recipe and optimized code

Sequential Code for(i=0;i< N;++i)for(j=0;j< N;++j)for(k=0;k< M;++k)for(1=0;1< M;++1)c[i][j] = c[i][j] + a[k+i][l+j] * b[k][l];**CUDA-CHILL Recipe** N=4096, M=32, TI=32, TJ=16, Tl=4

tile_by_index({"i 12 control="

permute(0,{"j","i

cudaize: conversion to a kernel function.

copy to shared: data copy to shared memory

Optimized Code

__shared__ float (_P1[47])[31];

by + 3 * ty + 2; tmp++)

__shared__ float (_P2[16])[17];float tmp3;

for $(tmp = 16 * by + 3 * ty; tmp \le min(16 * by + 30, 16 *$

for $(tx1 = 2 * tx; tx1 \le min(2 * tx + 1, 46); tx1++)$

```
normalize_index("j")
normalize_index("i")
cudaize("Kernel_GPU", {a=(N+M)*(N+M), b=M*M,
    c=(N+M)*(N+M),{block={"ij","ii"},
    thread={"i","i"}})
copy_to_shared("tx","a",-16)
copy_to_shared("tx","b",-16)
copy_to_registers("tx", "c")
Unroll_to_depth(1)
```

```
P2|tx1||tmp| = b|tmp||tx1|;
syncthreads();
tmp3 = c[k + 16 * by][tx + 32 * bx];
for (k = 0; k \le 15; k++)
     for (1 = 0: 1 \le 15: 1++)
       tmp3 = tmp3 + P1[1 + tx][k + ty] * P2[1][k];
c[k + 16 * by][tx + 32 * bx] = tmp3;
```

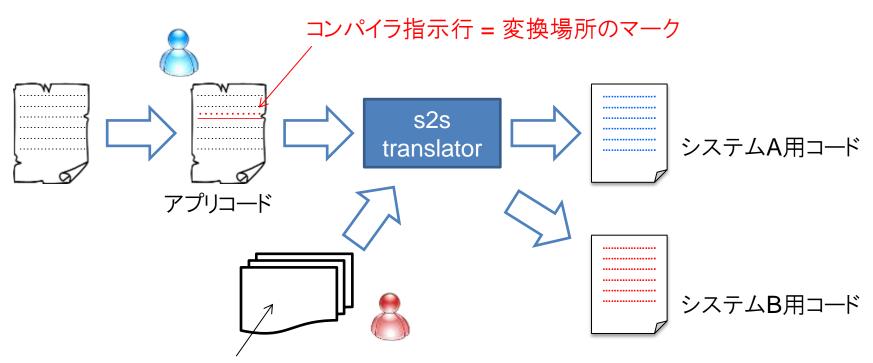
Figure by MALIK MURTAZA KHAN

ユーザ定義コード変換の必要性

TOHOKU

多様なシステムに適応するためには様々なコード変換が必要

→ 既存の変換の組み合わせだけでは表現不可

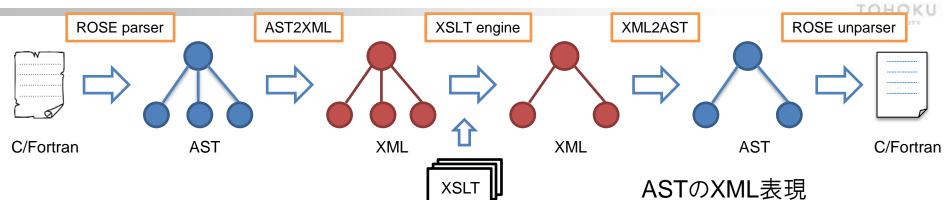


変換レシピ = 変換規則

- コンパイラ指示行の実際の挙動を定義
- システムごとに異なるレシピを利用可能
- アプリケーション開発者が独自な指示行を定義可能

Xevolver フレームワーク





C/Fortranのプログラム

for(a=0;a<N;a++)
{c=c+3;}</pre>



変換レシピから直接ASTを変更できるので 構文に基づく再利用性の高い変換規則を定義可能

XMLを介したプログラマとROSEの連携

71010711112200

<SgForStatement address="0x7fc839c38010">

<SgForInitStatement address="0x22b36a0">

<SgExprStatement address="0x2301ab0">

<SgAssignOp address="0x22e6520">

<SgVarRefExp address="0x22ccdd0" name="a"/>

<SgIntVal address="0x224b0a8" value="0" />

</SgAssignOp>

</SgExprStatement>

</SgForInitStatement>

<SgExprStatement address="0x2301b08">

<SgLessThanOp address="0x2317370">

<SgVarRefExp address="0x22cce38" name="a"/>

<SgIntVal address="0x224b110" value="100" />

</SgLessThanOp>

</SgExprStatement>

<SgPlusPlusOp address="0x2332900">

<SgVarRefExp address="0x22ccea0" name="a"/>

</SgPlusPlusOp>

<SqBasicBlock address="0x7fc839d06120">

<SgExprStatement address="0x2301b60">

<SgAssignOp address="0x22e6590">

<SgVarRefExp address="0x22ccf08" name="c"/>

<SgAddOp address="0x234c070">

<SgVarRefExp address="0x22ccf70" name="c"/>

<SgIntVal address="0x224b178" value="3" />

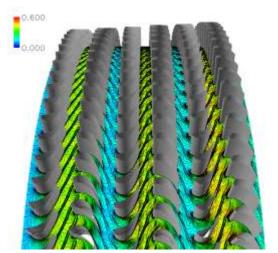
</SgAddOp> </SgAssignOp>

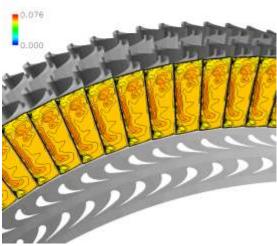
野価アプリケーション



・ 数値タービン

- タービン多段流路内の非定常3次元流れをシミュレート
 - 東北大学大学院情報科学研究科 · 山本研究室
- NEC SX-9@東北大学で2013年現在利用
 - SX-9向けにさまざまな最適化
 - 似たような形のループが多数出現 = 同じような変換を繰り返す必要







東北大山本研究室 http://www.caero.mech.tohoku.ac.jp/research/NumericalTurbine/NumericalTurbine.html

ループ変更 + OpenACCディレクティブ

тоноки

SX version

```
DO 200 M=1,MF
 DO 200 K=1,KF
    DO 200 J=1,JF
      DO 200 L=lstart,lend
        II1 = IS(L)
        II2 = II1+1
        II3 = II2+1
        IIF = IT(L)
        IIE = IIF-1
        IID = IIE-1
        DO 200 I=II2,IIF
          IF (I.LE.II3.OR.I.GE.IIE)THEN
            STBC=0.0D0
          ELSE
            STBC=1.0D0
          END IF
```

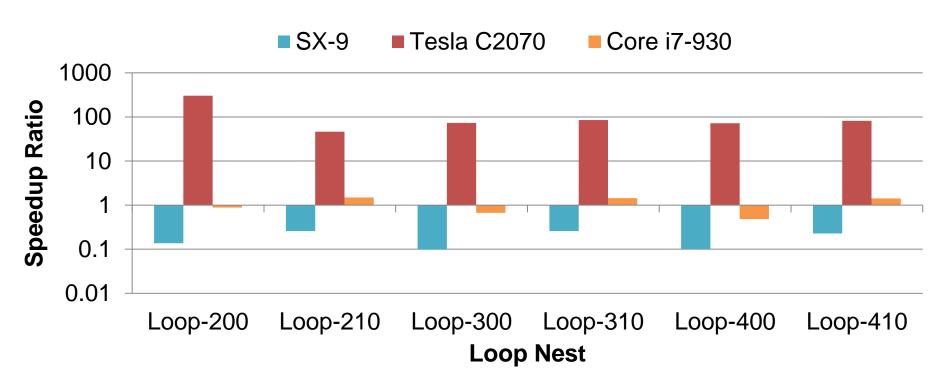
OpenACC version

```
!$acc loop private(L)
DO 200 M=1,MF
!$acc loop gang
  DO 200 K=1,KF
!$acc loop gang, vector
    DO 200 J=1,JF
!$acc loop vector
      DO 200 I=1,inum
!$acc loop seq
        DO L=1start,lend
          IF (I.ge.IS(L) .and. I.le.IT(L)) EXIT
        END DO
        IF (i.ne.IS(L)) THEN
           IF (I.LE.(IS(L)+2).OR.I.GE.(IT(L)-1)) THEN
             STBC=0.0D0
           ELSE
             STBC=1.0D0
           END IF
```

性能那個結果



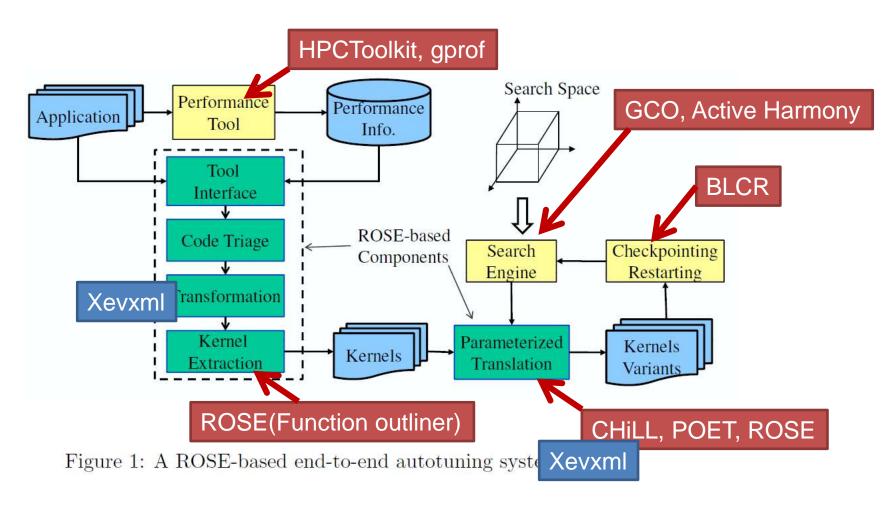
- ・SXとGPUでは異なる最適化が必要
 - SX向け数値タービンをOpenACC化した結果



変換レシピの切り替え→谷システムに適したバージョンを生成可能

ROSE Autotuningとの道第

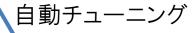




XMLデータを介した連盟



Empirical Tuning System



HPCToolkit

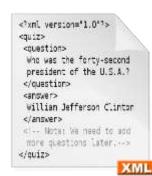


ROSE

高度な変換 解析結果データの挿入

他のコンパイラとの連携





相互運用の実現



可視化 リファクタリング (=対話的処理)

ユーザー定義の変換



XMLを開いたツール問題第に向けて

тоноки

・現状のXevolver実装

- ROSE Sage III IRのASTを忠実にXML化
 - ROSE ASTとの相互変換が容易 (^o^)
 - ROSEの仕様が変わる可能性 (T_T)
 - ROSE以外のツールとの連携は未考慮 (^_*;)

→ XMLのフォーマットを標準化する必要

- 寿命の長いHPCアプリの中長期的な開発には future-proofであることが重要
- ASTを書くためのXMLフォーマット = XCodeML

同音の違い (XevXML vs XcodeML)

```
тоноки
```

```
int a[10];
int xyz;
struct {int x; int y;} S;
void foo() {
   int *p;
   p = &xyz;
   a[4] = S.y;
}
```

XevXML(デフォルト表示)

```
<SgExprStatement>
  <SgAssignOp>
    <SgVarRefExp name="p"/>
    <SgAddressOfOp mode="0" >
        <SgVarRefExp name="xyz"/>
        </SgAddressOfOp>
    </SgAssignOp>
  </SgExprStatement>
```

XcodeML(仕様書より)

同音の違い (XevXML vs Photran)

```
TOHOK
```

XevXML(デフォルト表示)

Eclipse Photran

憩とめ



- ・ユーザ定義のコード変換の必要性
 - 既存のディレクティブの組み合わせだけでは性能可搬性を実現不可
 - → Xevolver: コード変換を容易に定義可能な環境
- 評価結果
 - 数値タービンでSXとGPUの性能可搬を達成
 - 比較的簡単な変換であれば容易に実現可能
 - 元のコードへのシステム特有の変更を回避可能
 - 特定のコードパターンの検出
- ・今後の展望
 - XMLによるAST記述の標準化
 - 他のツールとの連携



本発表は東北大学の研究グループで行われている成果をまとめたものです



小林広明



江川隆輔



小松一彦



平澤将一



王 春艶

多くの小林・滝沢研究室メンバーに支えられています

本研究はJST CREST「進化的アプローチによる超並列複合システム向け開発環境の創出」の支援を受けています