

# K-scope導入チュートリアル

2014年12月12日(金)

理化学研究所 計算科学研究機構

運用技術部門ソフトウェア技術チーム

寺井 優晃

- 準備
- Omni XcalableMP (XMP) のコンパイル
- 補助ツールのコンパイル
- 中間コードの生成
- K-scopeの起動
- トラブルシューティング

# はじめに

K-scopeは、アプリケーションのプライマリな開発者ではない立場で、HPC分野におけるチューニングを行なっているエンジニア、研究者を対象とした、Fortranソースコード分析の支援ツールです。このツールを用いることで、単純なエディタやビューワーを利用するのに比べて、ソースコードの全体像を把握する手間を低減することができます。

また、本ツールは、Pure Javaで書かれており、特定のフレームワークやライブラリに依存していないため、これを元に機能拡張することは容易だと考えます。

なお、K-scopeの分析には、Fortranソース以外に、Omni XscalableMP (XMP)コンパイラのF-Frontの出力（中間コード）を使用します。本資料では、K-scopeで分析に入るまでの最初のステップをチュートリアルとして示します。

## ★ステップ 1

K-scopeはJavaアプリケーションです。実行には、Javaのランタイムが必要です。JRE/JDKをダウンロードして、PC等にインストールしてください。

## ★ステップ 2

K-scopeは、XMPのフロントエンドが出力する中間コードを使用します。現状では京のログインノードにXMPがインストールされているので、京のユーザであればそれを利用して中間コードを生成することができます。

京のユーザでない場合、PC等にXMPのインストールが必要となります。コンパイルには以下のものを予めインストールする必要があります。

- Apache-ant
- libxml2\_dev
- byacc
- flex
- bison

# XMPの導入

現在、京コンピュータのログインノードにXMPがインストールされています。ログインユーザであれば、このステップは省略することができます。(/opt/aics/omni/bin/atool)

F-Front は XMPコンパイラのフロントエンドプログラムです。  
まずはXMPコンパイラを以下サイトからダウンロードします。

<http://www.hpcs.cs.tsukuba.ac.jp/omni-compiler/xcalablemp/download.html>

```
$ tar xjvf omnixmp-0.6.0.tar.bz2
$ cd omnixmp-0.6.0

$ vi Makefile.in
```

Versionによってはファイル名が多少異なる可能性があります。  
適当に置き換えてください。

次にF\_Frontのコンパイルとインストールをします。  
インストール先は適当に変更下さい。

```
$ ./configure --prefix=/home/teraim/opt/xcalablemp
$ make
$ make install
```

通常は必要ありませんが、バックエンドコンパイラの指定もここで行います。指定方法は ./configure -help で参照できます。

インストールに成功すると、--prefixで指定したディレクトリ配下(bin/)に実行モジュール群が生成されます。なおK-scopeは、F\_Front のみ使用します。

# 中間コード生成の手順

F-Frontを用いてFortranソースから中間コードに変換するプロセスは、最適化、ロードモジュールを作成しないという点を除けば、コンパイルと同等のプロセスを意味します。

基本的にはMakefile、Mkincludeファイル中のコンパイラを指定する変数（FC等）を書き換えるだけで完了します。

なお、処理系が提供するようなヘッダファイル“mpif.h”等がinclude文により呼ばれている場合はF-Frontが参照するので、-Iオプションを用いて以下のように指定する必要があります。もしヘッダファイルが参照できない場合は、Fortranソースの修正が必要になります。

また、ソースコード中にUSE文を用いてMPI関数を定義していた場合は、include文を用いたヘッダファイルによる定義に変更する必要があります。

以下は典型的なアプリのMakefile/Mkincludeの変更後の例です。

```
FC = atool
```

```
FFLAGS = -O0 -Xf -I/home/teraim/opt/mpich/include -cpp -emit-frontend-result -Wf,-endlineno
```

この際、-Wf,-endlineno がないと中間コードに行番号が出力されないため、K-scopeでのツリーが正常に表示されない場合がありますのでご注意ください。

# C言語のソースが含まれる場合

Cによるソースコードがアプリケーション・スイートに含まれている場合は、以下のような対処方法があります。ただし、C言語はK-scope自身が対応していないこともあり、現状は困難であると、ご理解ください。

- Cのソースがあまり重要でなく、かつ数ファイル程度で、とりあえずアプリケーション全体の間コード生成を行いたい場合は、C-Frontによってxmod中間オブジェクトを生成する方法があります。この場合、C-Frontを用意する必要があることと、該当ソースコードに対してcppを先に通しておき、フロントエンド（C-Front）を直接ハンドリングする必要があります。
- 同様にCのソースがあまり重要でない場合、FortranソースコードからのCソースコード上のサブルーチン（関数）呼び出しをコメントアウトし、Makefileを修正することで回避する方法があります。当然ながら、アプリケーションの構造を理解されている必要があります。

# 補助ツールのコンパイル

以上、F-Frontについて説明しましたが、F-Frontはユーザ向けのリッチなインターフェイスを持っていないので、今回、補助ツール（ドライバ）を別途用意しています。プリプロセッサとの煩雑な連携は、このドライバを用いることで軽減することができます。

必要に応じて、以下サイトからダウンロードください。

<http://www.aics.riken.jp/ungi/soft/kscope/>

なお、補助ツールの利用方法や引数等の詳細は、『補助ツール利用手引き』を参照して下さい。

```
$ tar xvzf atool.tgz
$ cd tool
$ ./configure --prefix=/home/teraim/opt/xcalablemp
$ make
$ make install
```

--prefix=<path>で指定するパスは、F\_Frontがインストールされている場所を指定して下さい。

現在、atoolはOmni XcalableMPコンパイラの開発元から正式に提供されています。



# K-scopeの起動

K-scopeの配布アーカイブを展開します。

なお、Versionによってファイル名が若干異なる可能性があります。

```
$ tar xvzf kscope_bin_${version}.tgz
```

`${version}` の部分を適当に置き換えてください。  
(例: version 0.2.4の場合は、kscope\_bin\_0.2.4.tgz)

解凍後、kscope.jarと2つのディレクトリ (propertiesとresources)が生成されます。このディレクトリは起動に必要ですので削除しないで下さい。

GUI環境下で以下コマンドを実行することで、ツールが起動します。

```
$ java -jar kscope.jar
```

起動後は、ファイル ⇒ プロジェクトの新規作成、タイトルを入力して、プロジェクトフォルダと中間コードのフォルダを指定した後に、新規作成ボタンを押すとツール側での解析が開始します。

解析が正常に終了すると、左側のエクスプローラビューにツリーが表示されます。解析ダイアログの表示までに若干のタイムラグがある場合があります。

操作の詳細は『[K-scope利用手引き](#)』を参考にして下さい。

# K-scopeの起動の補足

起動は以下のように `-Xmx` オプションでヒープの最大サイズを指定することが推奨されます。アプリケーション依存ですが、可能な限り大きめのサイズを指定した方が良いです。

```
$ java -jar -Duser.language=ja -Xmx1024m kscope.jar
```

また、`-Duser.language` で英語(en)と日本語(ja)を切り替えます。英語がデフォルトですが、実行環境（ロケール）によっては指定がなくても自動的に日本語が選択される場合があります。

陥りやすい問題点を以下に整理します。

- 中間コードの作成には、通常のコンパイル時と同じように、手続きや変数が宣言され、構文エラーが無い必要があります。つまりコンパイルできないものは解析できません。
- F\_Frontの仕様により、USE文を用いてMPI関数が定義されていた場合は、include文によって定義されるようにソースコードを書き換える必要があります。つまり、ソースコードのないライブラリをUSE文で結合する場合は注意が必要です。
- 補助ツールを用いて、Fortranソースから中間コードへの変換が失敗する場合、F-Frontを直接ハンドリングして、変換が可能か確認した方が良い場合があります。それで変換できない場合は、ソースコードに問題があるか、F-Frontに問題がある可能性があります。F-Frontによる変換ができる場合は、補助ツールの問題の可能性もあります。
- 新規プロジェクト等で中間コードをパースする際に、メモリが不足する場合があります。メモリが不足すると、動作が不安定になりダイアログの進捗バーが固まる場合があります。
- propertiesディレクトリのプロパティファイルの読み込みに失敗した場合、K-scopeは起動しません。それらは配布パッケージに含まれています。