

# Total View Debugger 利用の手引

東京工業大学学術国際情報センター  
2015.04  
version 1.0

# 目次

Total View Debugger 利用の手引き	1
1. はじめに	1
1.1 利用できるバージョン	1
1.2 概要	1
1.3 マニュアル	1
2. TSUBAME での利用方法	2
2.1 Total View Debugger の起動	2
(1) TSUBAMEにログイン	2
(2) バージョンの切り替え	2
(3) GUIの起動	2
2.2 ライセンス使用状況の確認	4
2.3 並列プログラムのデバッグ	4

# Total View Debugger 利用の手引き

## 1. はじめに

本書は、Total View Debugger を東京工業大学学術国際情報センターの TSUBAME で利用する方法について説明しています。また、TSUBAME を利用するにあたっては、「TSUBAME利用の手引き」もご覧下さい。利用環境や注意事項などが詳細に記述されています。

### 1.1 利用できるバージョン

TSUBAMEで利用可能なバージョンは次の通りです。

アプリケーション名	バージョン
Total View Debugger	8.8.0-2, 8.9.0-2, 8.9.2-0, 8.14.1-8

#### 備考

GPU - Future Support Plan があります。

[http://www.totalviewtech.com/forms/beta\\_registration.html#features](http://www.totalviewtech.com/forms/beta_registration.html#features)

### 1.2 概要

TotalView Debuggerは、GUIベースのソースコードデバッガです。シミュレーションとモデリング、特殊効果、航空宇宙、アニメーション、テレコミュニケーション、金融、科学的分析法、バイオテクノロジーなど、幅広い分野で利用されています。

Total View Debuggerの特長は次の通りです。

- 業界標準の並列デバッガ
- メモリデバッグ機能
- 非侵入型のメモリデバッグでリーク検出
- 各種言語/コンパイラをサポート
- C/C++, Fortran 77/90
- 操作性に優れたGUI
- マウス操作で主要なコマンドを実行可能
- 分散、並列プログラミングモデルをサポート
- 大規模プロジェクトの複雑なプログラムもデバッグ可能
- 様々な言語をサポート(C/C++, Fortran 77/90, UPC)
- 各種プラットフォーム、アーキテクチャに対応(Linux, UNIX, Mac OS X, AIX, Solaris, Cray 他)
- マルチスレッドプログラミング
- MPI プログラミング

### 1.3 マニュアル

マニュアルは、インストールディレクトリ内に PDF ドキュメントとして格納されています。各バージョンごとのドキュメント格納ディレクトリは次の通りです。

Ver 8.9.2-0

```
/usr/apps.sp3/isv/TotalView/8.9.2-0/toolworks/totalview.8.9.2-0/doc
```

Ver 8.14.1-8

```
/usr/apps.sp3/isv/TotalView/8.14.1-8/toolworks/totalview.8.14.1-8/doc/pdf
```

## 2. TSUBAME での利用方法

### 2.1 Total View Debugger の起動

#### (1) TSUBAMEにログイン

次のコマンドを入力し、TSUBAMEにログインします。

```
$ ssh -Y login-t2.g.gsic.titech.ac.jp -l USER-ID
```

#### 備考

- -l USER-ID の -l は数字の1ではなくアルファベットLの小文字です。
- GUI 起動のため、cygwin などの X サーバソフトウェアを用いて TSUBAME に接続してください。
- ssh コマンドによる TSUBAME へのログインの際、X11転送のため-Y オプションを指定してください。

#### (2) バージョンの切り替え

特にバージョンの指定がない場合は、バージョン 8.14.1-8 が起動するようになっています。

バージョンを切り替える場合は、それぞれ以下のように環境変数設定を行ってください。

#### 備考

バージョン 8.9.0-2 より前のバージョンは、2014年8月のTSUBAMEのOSアップグレード前に導入されたものとなります。2014年8月以降のTSUBAMEの環境では、正常動作しない可能性がありますのでご注意ください。

○バージョン8.8.0-2を使用する場合

```
<bash 系の場合> $ export PATH="/usr/apps/isv/TotalView/8.8.0-2/toolworks/totalview.8.8.0-2/bin:${PATH}"  
<csh 系の場合> % setenv PATH "/usr/apps/isv/TotalView/8.8.0-2/toolworks/totalview.8.8.0-2/bin:${PATH}"
```

○バージョン8.9.0-2を使用する場合

```
<bash 系の場合> $ export PATH="/usr/apps.sp3/isv/TotalView/8.9.0-2/toolworks/totalview.8.9.0-2/bin:${PATH}"  
<csh 系の場合> % setenv PATH "/usr/apps.sp3/isv/TotalView/8.9.0-2/toolworks/totalview.8.9.0-2/bin:${PATH}"
```

#### (3) GUIの起動

最初に、デバックする実行モジュールを作成します。

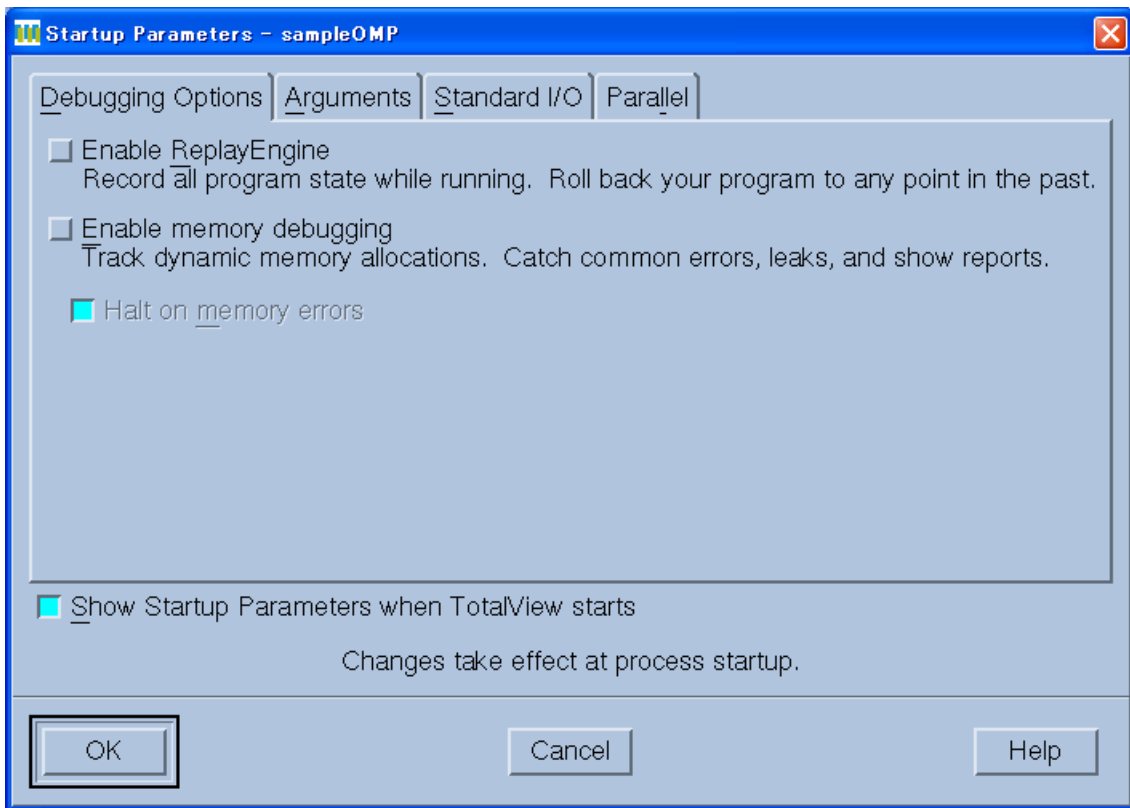
```
gfortran -g -o sampleOMP sampleOMP.f wclock.c
```

ここでは、-gのオプションを付加します。sampleOMPという名前の実行モジュールが出来ているはずで、

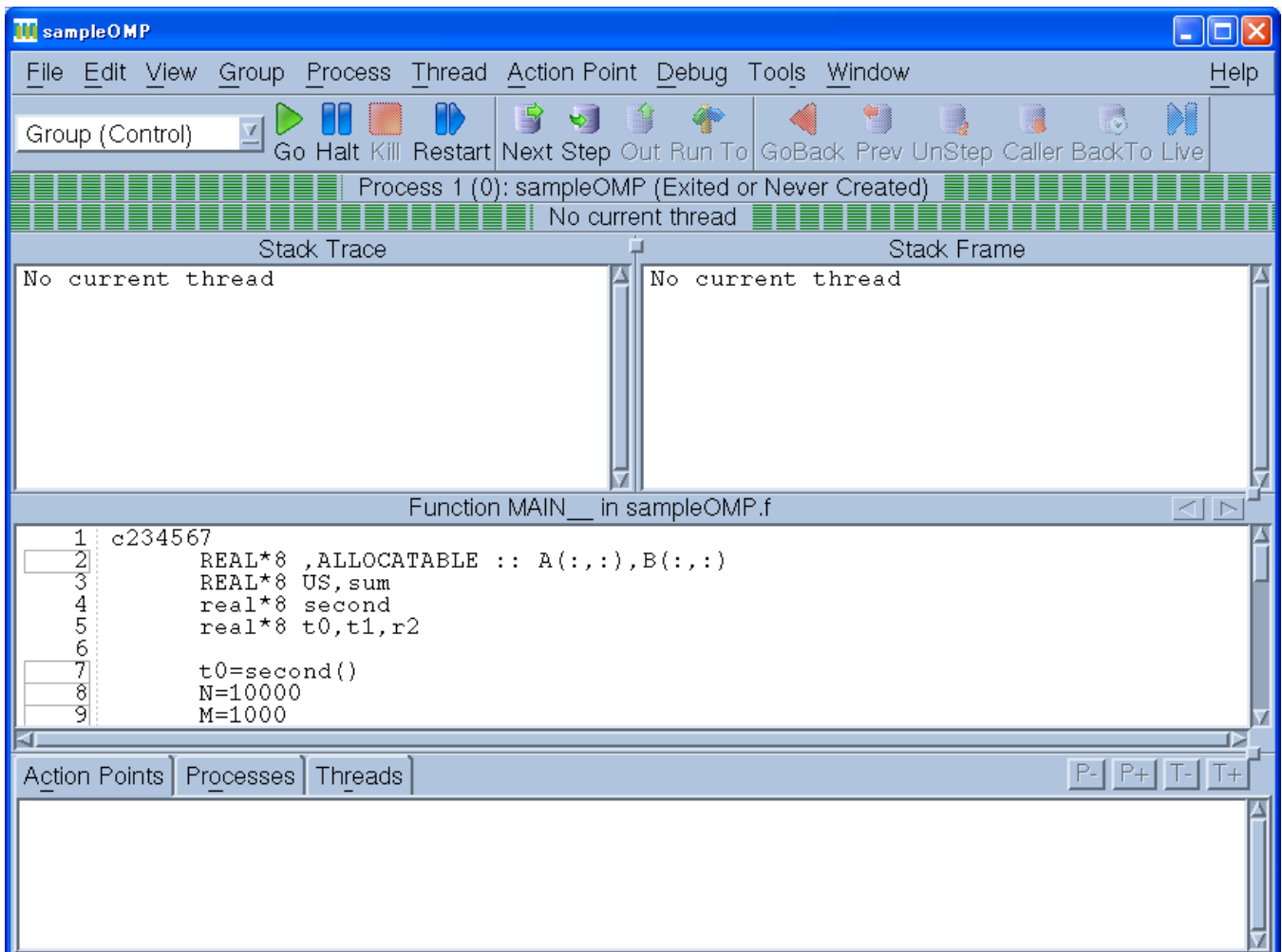
```
totalview sampleOMP
```

これにより、以下のような起動画面が表示されます。(プログラムを指定していないと選択画面が出ます)

## 2. TSUBAME での利用方法



OKをクリックすることで次の画面に遷移します。



終了する場合は、[File]-[Exit]を選択してください。

## 2.2 ライセンス使用状況の確認

TSUBAME 全体で同時利用できるユーザ数には上限(1024まで)があります。次のコマンドにより、ライセンス利用状況を確認できます。

```
$ /usr/apps/isv/flexlm/lmutil lmstat -S toolworks -c 27040@t2zlic01
```

## 2.3 並列プログラムのデバッグ

totalviewを使ってプログラム動作を調べてみましょう。デバックする実行モジュールを作成します。

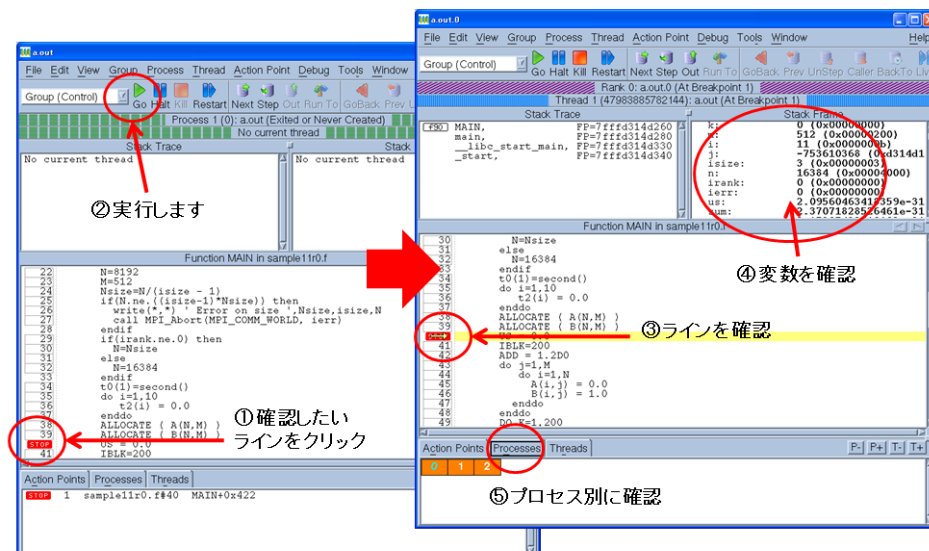
```
mpif90 -g sample11a.f wclock.c
```

ここでは、-gのオプションを付加します。a.outという名前の実行モジュールが出来ているはずですから、

```
totalview a.out
```

これにより、以下のような起動画面が表示されますので、mpi環境と並列数を入力します。

シングルの時と同じような画面が表示されてきますので、プログラムを調査したい部分のライン番号をクリックしてマーキングします。赤くSTOPと表示されたらメニューから「Go」を選択して実行します。



暫らくまわっていると指定のポイントで停止しますので、このときの変数の値を右上のウィンドウからチェックします。

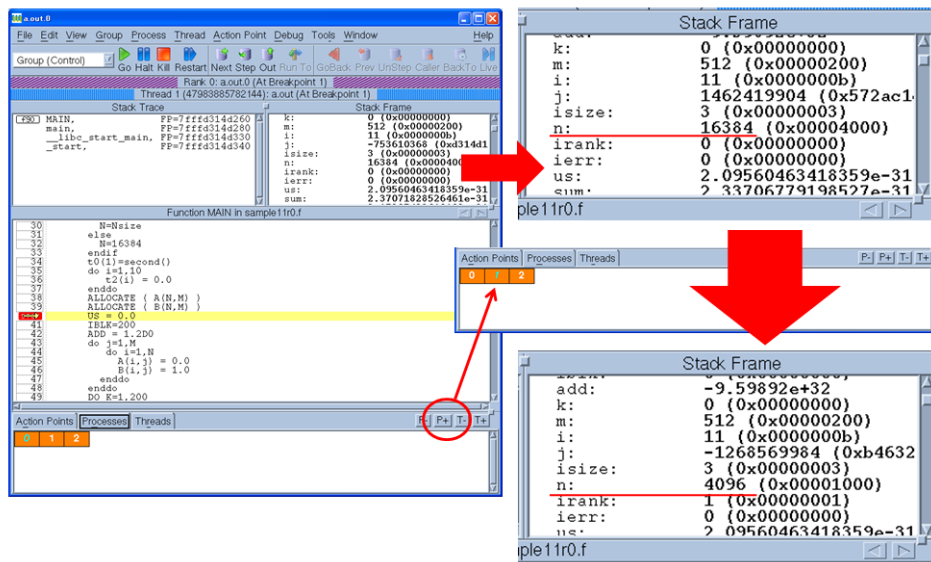
ここでは調査を分かりやすくするためにrankによって動作の異なるプログラムを用意しました。

```
N=8192
M=512
Nsize=N/(isize - 1)
if(N.ne.((isize-1)*Nsize)) then
  write(*,*) ' Error on size ',Nsize, isize,N
```

## 2.2 ライセンス使用状況の確認

```
call MPI_Abort(MPI_COMM_WORLD, ierr)
endif
if(irank.ne.0) then
  N=Nsize
else
  N=16384
endif
t0(1)=second()
```

このプログラムではrank=0にデータを集めるために配列を大きくとっています。下の図で rank=0でnの値は16384となっていますが、ここで、P+を選択して注目プロセスを移動します。rank=1では、n=4096となっていることが判ります。



このような操作を順次繰り返してデバックを進めてください。