GHOST-MP

2013 年 11 月版

操作マニュアル

監修　　：　東京工業大学 秋山研究室 GHOST-MP チーム

資料作成：　株式会社 情報数理バイオ

メールアドレス：　ghost-mp@bi.cs.titech.ac.jp

内容

1. コンパイル

1.1. 必要なソフトウェア

京、または、SCLS 計算システム(FX10)では boost C++ライブラリを使用する。本プログラムに boost 1.41.1 ライブラリとその京用のパッチを添付しているので、これをコンパイルする必要がある。

1.2. コンパイル手順

コンパイルすべきプログラムは、boost C++ライブラリと GHOST-MP の 2 つのプログラムである。

1.2.1. アーカイブの解凍展開

```
$ tar xvzf ghostmp-k-201311.tar.gz
```

1.2.2. Boost C++ライブラリのコンパイル

```
$ cd boost_1_46_1
$ patch -p0 < ./boost_1_46_1-r1788.patch
$ ./bootstrap.sh
$ ./bjam
```

boost_1_46_1 ディレクトリ以下に boost C++ライブラリが作成される。
boost_1_46_1-r1788.patch は京用の patch である。
作成した boost C++ライブラリは、GHOST-MP の Makefile 中で

```
BOOST_HOME=../boost_1_46_1
```

と指定して使用する。

1.2.3. GHOST-MP のコンパイル

GHOST-MP のコンパイルでは、データベース作成用のプログラム ghostmp_mkdb と、ホモロジー計算用のプログラム ghostmp の 2 つの実行プログラム作成する。

```
$ cd ghostmp
$ make clean          (←もし存在すれば、オブジェクトファイルをすべて消去する)
$ make                (←ghostmp_mkdb と ghostmp を作成する)
```

コンパイルに成功するとバイナリファイル ghostmp_mkdb と ghostmp が作成される。
GHOST-MP の Makefile は、すでに京、または、SCLS 計算システム(FX10)用に設定されているので編集する必要はないが、Makefile 中のマクロ定義について次項で説明する。マクロ定義を変更すると、生成される実行プログラムが変更する。

1.2.4. GHOST-MP の Makefile のマクロ定義

-D 指定	プログラム内容
-DCHUNK	チャンク分割版を生成する。指定がない場合はオリジナル版（すなわちベースとなった GHOSTX そのものが mpidp で並列分散実行されるプログラム）を生成。 チャンク分割版は、データ集合通信によってオリジナル版より DB 読み込み効率を改善している。
-DIOMASTER	3 階層(master,submaster,worker)版を生成する。 指定がない場合は、2 階層(master,worker)版を生成。 -DCHUNK の指定が必須である。 3 階層版は、京におけるファイル I/O 効率をデータ通信によって 2 階層版より改善している。
-DKEI	京固有の関数を使うために、京環境の mpi-ext.h をインクルードする。京、または、SCLS 計算システム(FX10)以外では定義しない。
-DDEBUG_LOG	開発用のログ出力。

図 1　GHOST-MP の Makefile のマクロ定義

京用にはデフォルトで

-DCHUNK -DIOMASTER -DKEI

が設定されている。

2. データベースの作成

GHOST-MP でホモロジー検索を行う場合、事前にデータベースの作成が必要である。データベースは ghostmp_mkdb を実行して作成する。

データベースは FASTA フォーマットで記述された DNA またはタンパク質配列のファイルから生成する。ghostmp_mkdb は、FASTA フォーマットのファイルを データベースファイルに変換する。

ghostmp_mkdb は 2 つの引数（[-i dbFastaFile]と[-o dbName]）が必要である。ghostmp_mkdb は、1 つの FASTA ファイルをいくつかのチャンクに分割し、 それぞれのチャンクについて 5 つのファイルを生成する（.inf, .ind, .nam .pos, .seq）。GHOST-MP で検索を行うためには、生成された全てのファイルが必要である。ghostmp_mkdb によるデータベース構築にあたって、ユーザーはチャンクサイズを指定できる。小さなチャンクサイズを指定した場合、データベースの構築と検索の際のメモリ使用量は少なくなるが、検索の効率（速度）が低下する。チャンクサイズとしてデフォルトの 1GB を選択した場合、データベースの構築の際に約 10GB、検索の際に約 13GB メモリを使用する。

2.1. ghostmp_mkdb の使用方法

```
SYNOPSIS
%> ghostmp_mkdb [-i dbFastaFile] [-o dbName] [-l chunkSize]
```

```
OPTIONS（[ ]内はデフォルト値）
（必須）
   -i STR      データベース構築のための FASTA フォーマットのタンパク質配列
   -o STR      データベース名
（任意）
   -l INT      データベースのチャンクサイズ（バイト）  [1073741824 (=1GB)]
```

図 2　使い方の概要

```
ghostmp_mkdb -i ./db.fasta -o exdb
```

図 3　使用例

3. GHOST-MP の使用方法

SYNOPSIS

%> mpiexec -n numNodes ghostmp [-tb tableName または -cf configName]
[-lg log_file] [-rc flag_rankchank] [-dx x_size] [-dy y_size] [dz z_size]
[-po flag_post] [-bc flag_broadcast] [-rd flag_redistribution]
[-rt number_of_retries] [-ot output_file_order]
([-i queries] [-o output] [-d database])
[-v maxNumAliSub] [-b maxNumAliQue] [-M scoreMatrix]
[-G openGap] [-E extendGap]
[-l CandidatesSize] [-s lowerCutoff] [-T UpperCutoff]
[-S searchLength] [-q queryType] [-t databaseType]
[-r EntropyThreshold1] [-f EntropyThreshold2]
[-a numThreads] [-L maxNumHits] [-w maxAliLen]

OPTIONS（並列分散処理のオプション、[ ]内はデフォルト値）

-tb STR　JOB テーブルファイル名
　または
-cf STR　CONFIG ファイル名（-tb と -cf のどちらか一方だけ必要）
-lg STR　ログ出力ファイル名 [mpidp.log]
-rc INT　全ランクの分割方法の指定 [1]
　0: -dx -dy -dz でサイズ指定した直方体で京のトーラス空間を分割
　1: 全ランクを連番的にチャンク数個に分割
-dx INT　直方体の x 方向サイズの指定（-rc 0 を指定したときだけ有効）[2]
-dy INT　直方体の y 方向サイズの指定（-rc 0 を指定したときだけ有効）[1]
-dz INT　直方体の z 方向サイズの指定（-rc 0 を指定したときだけ有効）[1]
-po INT　計算の最後で post 処理を実行する(1)かしない(0)か。
　（-po 1 は、-rt 0 を指定したときだけ有効）[0]
-bc INT　submaster だけチャンク DB を読み、worker に Broadcast 通信する(1)
　か、Broadcast 通信しないですべての worker ノードでチャンク DB を読む(0)
　かの指定 [1]
-wo INT　worker でホモロジー検索結果ファイルを出力する(1)か、submaster で出力する(0)か、submaster で出力する場合は worker からデータ通信する。[1]
-rd INT　2 階層(master,worker)版のときに計算の最後で、先に計算が終了したランクに別のチャンク DB の計算を割り当てる(1)か否(0)か。3 階層版のときは無視する。[0]
-rt INT　ノード障害時の JOB の実行繰り返し数の上限 、通常は 0（デフォルト）に設定する。-rt が 0 でない場合は MPI_Abort()でプログラムが終了する。[0]

```
  -ot INT     出力ファイルの JOB リスト内での順番。アプリケーションが出力する
              計算結果ファイルの有無を確認するためで必須ではない。
              （JOB リストで出力ファイルの指定がない場合、あるいは、-cf 指定の場合
               は無効）
OPTIONS（ホモロジー検索のオプション、[ ]内はデフォルト値）
 (必須：JOB テーブルファイル内で指定、CONFIG ファイル使用時はデフォルトで指定)
  -i STR     クエリファイル名（FASTA フォーマット）
  -o STR     出力ファイル名
  -d STR     データベース名
 (任意)
  -v INT     ヒットした各配列に対する最大アラインメント数 [1]
  -b INT     クエリ 1 つに対する最大出力数 [1]
  -M STR     スコアマトリクスのファイル名 [BLOSUM62]
  -G INT     ギャップ開始ペナルティ [11]
  -E INT     ギャップ伸長ペナルティ [1]
  -l INT     クエリ読み込み時のチャンクサイズ (Bytes) [134217728 (=128MB)]
  -s INT     シード検索時のスコアカットオフ下限 [1]
  -T INT     シード検索時のスコアカットオフ上限 [30]
  -S INT     シード検索時の最大アラインメント長 [8]
  -q STR     クエリの配列タイプ、p (protein) or d (dna) [p]
  -t STR     データベースの配列タイプ、 p (protein) or d (dna) [p]
  -r FLOAT   クエリ配列を並列効率向上のために並べ換えるときのエントロピー閾値
             [3.0]
  -f FLOAT   クエリ配列をフィルタリングするときのエントロピー閾値 [0.0]
  -a INT     スレッド数 [1]
  -L INT     最大ヒット数 [2097152]
  -w INT     最大アラインメント長 [50]
```

図 4　使い方の概要

```
mpiexec –n 24 ghostmp -tb $TABLE -lg $LOG -rc 0 -dx 2 -dy 2 -dz 3 –po 1
 -a $OMP_NUM_THREADS -b 5 -T 32 -r -1 -q d -t p -M $SCORE_MATRIX -G 9 -E 1
```

図 5　使用例

多数の Query ファイルを並列分散処理するために、JOB テーブルファイルまたは CONFIG ファイルを作成して、そのファイルのパスをオプション指定する（[-tb tableName]または[-cf configName]）。

JOB テーブルファイルと CONFIG ファイルの説明は次節の通り。

3.1. JOB テーブル

```
TITLE=ghostmp
PARAM=-i $1 -d $2 -o $3
../q.1  ./db/db ../o.1
../q.2  ./db/db ../o.2
../q.3  ./db/db ../o.3
../q.4  ./db/db ../o.4
../q.5  ./db/db ../o.5
../q.6  ./db/db ../o.6
../q.7  ./db/db ../o.7
../q.8  ./db/db ../o.8
../q.9  ./db/db ../o.9
../q.10 ./db/db ../o.10
```

図 6　JOB テーブルファイルの記入例

10 個のジョブの入力ファイル、データベース、出力ファイル（すなわち、GHOST-MP の-i -d -o オプションの３つ）を順に記載している。

	データ 1($1)	データ 2($2)	データ 3($3)	..
タイトル行（任意）	TITLE=XXXXXXXXXXXXXXXXX			
オプション入力指定（必須）	PARAM=option1 $1 option2 $2 option3 $3 …			
一行毎に JOB を指定（必須）	Job1data1	Job1data2	Job1data3	..
	Job2data1	Job2data2	Job2data3	..
	Job3data1	Job3data2	Job3data3	..
	..	..	..	..

図 7　JOB テーブルファイルのフォーマット

タイトル行（任意）とオプション入力指定行（必須）のあとに、1 行毎に並列実行すべき JOB のオプションデータを指定する。複数のカラムはタブ区切りにする。

3.2. CONFIG ファイル

```
!
! This is config file
!
TITLE
ghostmp

QUERY_NO
10

DATABASE_NAME
./db/db

QUERY_FILE_BASENAME
../q.

OUTPUT_FILE_BASENAME
../o.
```

図 8　CONFIG ファイルの記入例

図 6 の JOB テーブルファイルの例と全く同じ実行を CONFIG ファイルで指定した場合。10 個のジョブを記載している。

Keyword	意味
TITLE	タイトル
QUERY_NO	クエリファイルの数
DATABASE_NAME	データベース名のパス
QUERY_FILE_BASENAME	クエリファイル(BASENAME)のパス
OUTPUT_FILE_BASENAME	出力ファイル(BASENAME)のパス

図 9　CONFIG ファイルのフォーマット

5 個の Keyword の次の行に表の情報を記述する。1 行の中で、文字 '!' または '*' 以降の文字はコメントとする。また、行頭のスペースまたはタブは無視する。

BASENAME は以下に展開される。

"BASENAME"+"1", "BASENAME"+"2" ... "BASENAME"+"$QUERY_NO"

3.3. 検索結果ファイル

```
hsa:124045... hsa:124045... 1        139 0  0 1   139 1    139  2.04391e-76 283.878
hsa:124045... ptr:454320... 0.992126 127 1  0 13  139 14   140  5.96068e-68 255.758
hsa:124045... mcc:714360... 0.889764 127 14 0 13  139 14   140  5.05773e-59 226.098
hsa:124045... rno:292078... 0.586777 121 46 2 13  133 14   130  1.38697e-32 138.272
hsa:124045... mmu:320869... 0.559055 127 50 3 13  139 12   132  1.17414e-31 135.191
hsa:124045... pon:100434... 0.964912 57  2  0 13  69  14   70   3.65839e-25 113.62
hsa:124045... bta:100335... 0.449275 138 71 3 2   139 25   157  4.04482e-24 110.153
hsa:124045... aml:100464... 0.266667 75  46 2 13  81  1183 1254 0.820692    32.7278
hsa:124045... bfo:BRAFLD... 0.56     25  10 1 108 131 581  605  0.820692    32.7278
hsa:124045... tgu:100227... 0.261682 107 69 3 25  130 150  247  1.82831     31.5722
・・・
・・・
```

図 10　検索結果ファイルの出力例

GHOST-MP は検索結果をタブ区切りで出力する。

各カラムの内容は以下の通り

1. クエリ配列名
2. ヒット配列名
3. 一致率
4. アラインメント長
5. 不一致数
6. ギャップ数
7. クエリ配列におけるアラインメント開始位置
8. クエリ配列におけるアラインメント終了位置
9. ヒット配列におけるアラインメントの開始位置
10. ヒット配列におけるアラインメントの終了位置
11. E-value
12. 正規化スコア

3.4. ログファイル

```
GHOST-MP ver. 201311
      mpidp@bi.cs.titech.ac.jp   last updated: 2013/10/24

#RANK = 36
#Node = 36 (#RANK/Node = 1)

 used nodes list(id) :
  p05-044(0)  p05-045(1)  p05-032(2)  p05-033(3)  p05-020(4)
  p05-021(5)  p05-023(6)  p05-022(7)  p05-035(8)  p05-034(9)
  p05-047(10)  p05-046(11)  p05-036(12)  p05-037(13)  p05-024(14)
  p05-025(15)  p05-012(16)  p05-013(17)  p05-015(18)  p05-014(19)
  p05-027(20)  p05-026(21)  p05-039(22)  p05-038(23)  p05-040(24)
  p05-041(25)  p05-028(26)  p05-029(27)  p05-016(28)  p05-017(29)
  p05-019(30)  p05-018(31)  p05-031(32)  p05-030(33)  p05-043(34)
  p05-042(35)

Table file    : -tb ./table
Log file      : -lg ./log
RankC. option : -rc 0
KEI dx        : -dx 2
KEI dy        : -dy 2
KEI dz        : -dz 3
Other options : aln -a 8 -b 5 -T 32 -r -1 -q d -t p -M ./PAM30 -G 9 -E 1

TITLE=ghostmp
PARAM=-i $1 -d $2 -o $3
```

図 11　ログの出力例

計算で利用する RANK 数と Node 数、計算ホスト名の情報が出力される。また、計算実行時に指定されたオプションがエコー出力される他、JOB リストに記載されているタイトル(TITLE)、パラメータ情報(PARAM)が出力される。

実行終了時には、以下の表で説明する JOB 管理テーブル(JOB table :)と Worker 管理テーブル（Worker table :）が出力され、最後に経過時間(Elapsed time)が秒の単位で出力される。

表 1　JOB 管理テーブルの形式（スペース区切り）

Name	Status	Worker	Flag	Return	File	Worker	Flag	Return	File	‥
00001	1	1	1	0	0					
00002	1	2	1	0	0					
00003	2	3	0	0	0	6	1	0	0	‥
00004	1	4	1	0	0					
00005	2	5	0	0	0	7	1	0	0	‥
:	:	:	:	:	:	:	:	:	:	

表 2　JOB 管理テーブルの項目の意味

JOB 管理テーブルの項目	項目の意味
Name	JOB リストの各 JOB に割り当てられた 5 桁の ID
EXEC	JOB 毎の計算回数
WID	計算に割り当てられた rank
END	1:計算が終了した場合、0:不明な場合
RET	system()関数の戻り値
FILE	出力ファイルの JOB リスト内での順番フラグ-ot で指定された出力ファイルの確認： 1　：ファイルが存在した場合 -1：ファイルが存在しない場合 0　：未確認

表 3　Worker 管理テーブルの形式（タブ区切り）

Worker	Status	Name	Flag	Name	Flag	‥
1	10	00001	1	00009	1	
2	9	00002	1	00014	1	
3	4	00003	1	00012	1	
4	7	00004	1	00011	1	
5	4	00005	1	00013	1	
:	:	:	:	:	:	

表 4　Worker 管理テーブルの項目の意味

Worker 管理テーブルの項目	項目の意味
Worker	ワーカーの RANK 番号
EXEC	ワーカー毎の計算回数
Name	JOB リストの各 JOB に割り当てた 5 桁の Name
END	1:計算が終了した場合、0:不明な場合