Experiments for "Profiling the BLAST bioinformatics application for load balancing on high-performance computing clusters"

From Wikidb

Contents

Introduction

More experiments needed for the publication.

Location on Betsy: **/scratch/mikem/UserSupport/trinity.cheng/blast_surface**

Source codes on GitHub

Uploaded the below listed and mentioned in this Wiki page scripts to:

- https://github.com/DIDSR/BLAST_LOAD_BALANCING/tree/main/scripts

1. batch_run_v2.sh
2. create_db_and_query.sh

3. extract_blast2.sh
4. failed_rows.sh
5. sge_extractblast_v3.sh

- Not, the below instructions to create directories on the GitHub site was used, so ".gitkeep" file is just a placeholder for creating a directory: "You cannot create an empty folder and then add files to that folder, but rather creation of a folder must happen together with adding of at least a single file. This is because git doesn't track empty folders.

On GitHub you can do it this way:

- Go to the folder inside which you want to create another folder
- Click on New file
- On the text field for the file name, first write the folder name you want to create
- Then type /. This creates a folder
- You can add more folders similarly
- Finally, give the new file a name (for example, .gitkeep which is conventionally used to make Git track otherwise empty folders; it is not a Git feature though)
- Finally, click Commit new file."

Ref.: https://stackoverflow.com/questions/12258399/how-do-i-create-a-folder-in-a-github-repository

Getting a new query file

Downloaded from any:

> http://www.ncbi.nlm.nih.gov/sra/SRP102422
> https://www.ncbi.nlm.nih.gov/sra/SRP102422
> https://trace.ncbi.nlm.nih.gov/Traces/index.html?view=run_browser&acc=SRR5713923&display=download

https://trace.ncbi.nlm.nih.gov/Traces/sra-reads-be/fasta?acc=SRR5713923

Convert:

> See FASTQ file to FASTA file (https://scl-wiki.fda.gov/wiki/index.php/Technical_questions#FASTQ_file_to_FASTA_file)
> https://bioinformaticsworkbook.org/dataWrangling/fastaq-manipulations/converting-fastq-format-to-fasta.html#gsc.tab=0

" Using SED

> sed can be used to selectively print the desired lines from a file, so if you print the first and 2rd line of every 4 lines, you get the sequence header and sequence needed for fasta format.
>
> time sed -n '1~4s/^@/>/p;2~4p' SRR5713923.fastq > SRR5713923_2.fasta

Preparing database and query datasets

Scale-out on 4096 nodes

1. Based on Scale-out 4096 table prepare files m_values_4096.txt and n_values_4096.txt
2. Create an SGE script shown in create_db_and_query.sh.

   Adjust "QUERY_FILE=" and "SPLIT_QUERY=" lines in create_db_and_query.sh to select the old or new query file.

3. Submit an SGE job:

   qsub create_db_and_query.sh

Scale-out 4096 table

```
m       n       mxn     Di = D/m Qj = Q/n
1       4,096   4,096   523,449 17,848
2       2,048   4,096   261,725 35,695
3       1,365   4,095   174,483 53,555
4       1,024   4,096   130,862 71,389
5       819     4,095   104,690 89,258
6       682     4,092   87,242  107,188
7       585     4,095   74,779  142,961
8       512     4,096   65,432  142,778
16      256     4,096   32,716  285,555
32      128     4,096   16,358  571,110
64      64      4,096   8,179   1,142,220
```

m_values_4096.txt

```
8179
16358
32716
65432
130862
261725
```

n_values_4096.txt

```
17848
35695
53555
71389
89258
107188
124961
142778
285555
393022
480935
571110
696210
821372
1142220
```

create_db_and_query.sh

```
#!/bin/bash

# 07/29/2022

# input file: m_vals.txt, n_vals.txt
```

```
BASE_DIR=/scratch/mikem/UserSupport/trinity.cheng/blast_surface

# QUERY_FILE=${BASE_DIR}/orig_query_split/SRR5713923                    # new query
QUERY_FILE=${BASE_DIR}/orig_old_query_split/orig_query                  # old query

BASE_DB_DIR=/projects/mikem/UserSupport/ncbi/nt_2020
DB_NAME=nt
DB_FILE=${BASE_DB_DIR}/nt

MAKEBLASTDB=/projects/mikem/applications/centos7/blast2.12.0_fda/ncbi-blast-2.12.0+-src/c++/ReleaseMT/bin/makeblastdb

DBSEQUENCES=$(cat m_vals_4096.txt)
QUERYSEQUENCES=$(cat n_vals_4096.txt)

for dsequence in $DBSEQUENCES
do
        # location of extracted db
        SPLIT_DB=${BASE_DB_DIR}/orig_db_split/"${DB_NAME}""${dsequence}"
        echo $SPLIT_DB
        # If database does not already exist, extract fragments and create split database. Then, run makeblastdb to index.
        if [ ! -f $SPLIT_DB ];
        then
                awk -v N=$dsequence 'BEGIN {N_start=1; RS=">"}; {if (NR>N_start && NR<=N_start+N) {print ">" $0}}' $DB_FILE >
                $MAKEBLASTDB -in $SPLIT_DB -dbtype nucl
        fi
done

for qsequence in $QUERYSEQUENCES
do
        # location of extracted query
        # SPLIT_QUERY=${BASE_DIR}/orig_query_split/orig_query"$qsequence"                       # new query
        SPLIT_QUERY=${BASE_DIR}/orig_old_query_split/orig_query"$qsequence"                     # old query
        echo $SPLIT_QUERY
        # If query does not already exist, extract fragments and create split query.
        if [ ! -f $SPLIT_QUERY ];
        then
                awk -v N=$qsequence 'BEGIN {N_start=1; RS=">"}; {if (NR>N_start && NR<=N_start+N) {print ">" $0}}' $QUERY_FILE
        fi
done
```

Scale-out on 2048 nodes

1. Based on Scale-out 2048 table prepare files m_values_2048.txt and n_values_2048.txt
2. Create an SGE script shown in create_db_and_query.sh.

   Adjust (a) "QUERY_FILE=" and (b) "SPLIT_QUERY=" lines in create_db_and_query.sh to select the old or new query file.
   (c) "DBSEQUENCES=" and (d) "QUERYSEQUENCES=" lines to select scale out table for 2048 nodes.

3. Submit an SGE job:

   qsub create_db_and_query.sh

- **Note:** Since m_values_2048.txt and n_values_2048.txt are sub-sets of m_values_4096.txt and n_values_4096.txt respectively, there is no need to run create_db_and_query.sh for 2048 scale-out if it was run for 4096 scale-out.

Scale-out 2048 table

```
m       n       mxn     Di=D/m  Qj=Q/n
1       2,048   2,048   523,449 35,695
```

```
2       1,024   2,048   261,725 71,389
3       682     2,046   174,483 107,188
4       512     2,048   130,862 142,778
8       256     2,048   65,432  285,555
16      128     2,048   32,716  571,110
32      64      2,048   16,358  1,142,220
```

m_values_2048.txt

```
16358
32716
65432
130862
261725
523449
```

n_values_2048.txt

```
35695
53555
71389
89258
107188
124961
142778
285555
393022
480935
571110
696210
821372
1142220
```

Experiments

batch_run_v2.sh (also found at **/scratch/mikem/UserSupport/trinity.cheng/blast_surface/batch_run_v2.sh**) is used to batch run the experiments. batch_run_v2.sh in turn runs sge_extractblast_v3.sh via *qsub* SGE command.

"threads" is added to the DESC field in the job submit line to avoid overwriting results when running the experiments in the batch mode:

```
qsub -N "blast_array_${thr}_${t}" -l h_vmem=${MEM}G -pe thread $thr -l dell01=true -l gpus=0 sge_extractblast_v3.sh type1test"
```

So, the DESC field is combination of:

- Test type
- Test number
- Number of threads used
- Number of db sequences and
- Number of query sequences

This way BLAST will generate unique files for every qsub to avoid overwriting when run in batch mode.

Running with the new query

If we want to run with **the new query** before the batch run we need to (in extract_blast2.sh file located on **/scratch/mikem/UserSupport/trinity.cheng/blast_surface/extract_blast2.sh**):

a. Comment out lines: 14, 20 and 48
b. Uncomment lines: 13, 19 and 47.

See also Lines to modify in extract_blast2.sh.
Then you may adjust the below parameters in batch_run_v2.sh:

```
num_threads=(1 2 4 8)
NUM_REPEAT=3
```

And then run the below command from **/scratch/mikem/UserSupport/trinity.cheng/blast_surface** prompt:

```
bash  batch_run_v2.sh
```

Running with the old query

If we want to run with **the old query** before the batch run we need to (in extract_blast2.sh file located on **/scratch/mikem/UserSupport/trinity.cheng/blast_surface/extract_blast2.sh**):

a. Uncomment lines: 14, 20 and 48
b. Comment out lines: 13, 19 and 47

See also Lines to modify in extract_blast2.sh.
Then you may adjust the below parameters in batch_run_v2.sh:

```
num_threads=(1 2 4 8)
NUM_REPEAT=3
```

Then run the below command from **/scratch/mikem/UserSupport/trinity.cheng/blast_surface prompt**:

```
bash  batch_run_v2.sh
```

Lines to modify in extract_blast2.sh

Lines 13, 14:

```
13 # TIME_SUMMARY_DIR=${BASE_DIR}/time_summary                                  # new query result dir
14 TIME_SUMMARY_DIR=${BASE_DIR}/time_summary_old_query                          # old query result dir
```

Lines 19,20:

```
19 # QUERY_FILE=${BASE_DIR}/orig_query_split/SRR5713923                       # new query file
20 QUERY_FILE=${BASE_DIR}/orig_old_query_split/orig_query                      # old query file
```

Lines 47, 48:

```
47 # SPLIT_QUERY=${BASE_DIR}/orig_query_split/orig_query"$NREC_QUERY"          # new query split
48 SPLIT_QUERY=${BASE_DIR}/orig_old_query_split/orig_query"$NREC_QUERY"        # old query split
```

sge_extractblast_v3.sh

Author: Trinity Cheng, 2021 Summer Intern.

```
#$ -P CDRHID0014
#$ -cwd
#$ -l h_rt=48:00:00
#                                               $ -l h_vmem=2.5G
#$ -S /bin/sh
#$ -j y
#$ -o sge_results
#                               $ -N blast_array
#                       $ -t 1-90
#                               $ -pe thread 8

# This script runs all MxN combinations from m_vals and n_vals in an array job
# usage: qsub -l <nodename> sge_extractblast_v3.sh <description>
# type 1 = -q '*@@betsy_original'
# type 2 = -l bigbox
# type 3 = -l sm01
# type 4 = -l sm02
# type 5 = -l hpe01=true -l gpus=0
# type 6 = -l dell01=true -l gpus=0

# DESC is a description of the current job/experiment
DESC=$1
DBFILE=$2
QUERYFILE=$3

# DBFILE=m_vals.txt
# QUERYFILE=n_vals.txt


# MxN array, M is row/database, N is column/query

DBNUM=$(cat $DBFILE | wc -l)
QUERYNUM=$(cat $QUERYFILE | wc -l)

ROW=$((((((SGE_TASK_ID-1))/$QUERYNUM))+1))
COL=$((((((SGE_TASK_ID-1))%$QUERYNUM))+1))

NREC_DB=$(head -n $ROW $DBFILE | tail -n 1)
NREC_QUERY=$(head -n $COL $QUERYFILE | tail -n 1)

# run extract_blast with $NREC_DB database fragments and $NREC_QUERY query fragments
./extract_blast2.sh $NREC_DB $NREC_QUERY $DESC $NSLOTS
```

extract_blast2.sh

Author: Trinity Cheng, 2021 Summer Intern.

```
#!/bin/bash

# updated 7/21/2021

# input file: fasta format, delimiter '>'
# retrieve the first M records from 2.5 GB database, N records from query, put BLAST results in filenameMxN
```

```
# usage: extract_blast2.sh <M> <N> <filename>

# BASE_DIR=/scratch/trinity.cheng/blast_surface

BASE_DIR=/scratch/mikem/UserSupport/trinity.cheng/blast_surface

TIME_SUMMARY_DIR=${BASE_DIR}/time_summary                               # new query result dir
# TIME_SUMMARY_DIR=${BASE_DIR}/time_summary_old_query                   # old query result dir

mkdir -p ${BASE_DIR}/sge_results
mkdir -p ${TIME_SUMMARY_DIR}

QUERY_FILE=${BASE_DIR}/orig_query_split/SRR5713923                            # new query file
# QUERY_FILE=${BASE_DIR}/orig_old_query_split/orig_query                      # old query file


BASE_DB_DIR=/projects/mikem/UserSupport/ncbi/nt_2020
DB_NAME=nt
DB_FILE=${BASE_DB_DIR}/nt

# BASE_DB_DIR=/scratch/trinity.cheng/blast_surface/orig_db_split
# DB_NAME=orig_db
# DB_FILE=${BASE_DB_DIR}/${DB_NAME}

BLAST=/projects/mikem/applications/centos7/blast2.12.0_fda/ncbi-blast-2.12.0+-src/c++/ReleaseMT/bin/blastn
MAKEBLASTDB=/projects/mikem/applications/centos7/blast2.12.0_fda/ncbi-blast-2.12.0+-src/c++/ReleaseMT/bin/makeblastdb


echo $QUERY_FILE
echo $DB_FILE

NREC_DB=$1
echo $NREC_DB
# location of extracted db
SPLIT_DB=${BASE_DB_DIR}/orig_db_split/"${DB_NAME}""${NREC_DB}"
echo $SPLIT_DB

NREC_QUERY=$2
echo $NREC_QUERY
# location of extracted query
SPLIT_QUERY=${BASE_DIR}/orig_query_split/orig_query"$NREC_QUERY"                # new query split
# SPLIT_QUERY=${BASE_DIR}/orig_old_query_split/orig_query"$NREC_QUERY"          # old query split
echo $SPLIT_QUERY

DESC=$3

SLOTS=$4

# If query does not already exist, extract fragments and create split query.
if [ ! -f $SPLIT_QUERY ];
then
        awk -v N=$NREC_QUERY 'BEGIN {N_start=1; RS=">"}; {if (NR>N_start && NR<=N_start+N) {print ">" $0}}' $QUERY_FILE > $SPL
fi

# If database does not already exist, extract fragments and create split database. Then, run makeblastdb to index.
if [ ! -f $SPLIT_DB ];
then
        awk -v N=$NREC_DB 'BEGIN {N_start=1; RS=">"}; {if (NR>N_start && NR<=N_start+N) {print ">" $0}}' $DB_FILE > $SPLIT_DB
        $MAKEBLASTDB -in $SPLIT_DB -dbtype nucl
fi

BASE_OUT=/scratch/mikem/UserSupport/trinity.cheng/blast_surface
# run BLAST and time, put results in sge_reuslts/"$DESC"_"$NREC_DB"X"$NREC_QUERY"
# output the time in seconds into time_summary.txt file

if [ $SLOTS == 1 ]
then
   NUM_THREADS=""
else
   NUM_THREADS="-num_threads $SLOTS"
fi

echo "CMD: time $BLAST $NUM_THREADS -dbseqnum $NREC_DB -query $SPLIT_QUERY -db $SPLIT_DB"
TIMEFORMAT="%E %U %S";
(time $BLAST $NUM_THREADS -dbseqnum $NREC_DB -query $SPLIT_QUERY -db $SPLIT_DB) &> ${BASE_OUT}/sge_results/"$DESC"_"$NREC_DB"x
```

```
sleep 1
TIME=$(tail -n 1 ${BASE_OUT}/sge_results/"$DESC"_"$NREC_DB"x"$NREC_QUERY")

# output line: "M N time" for every M and N configuration into time_summary file.
## echo "$NREC_DB" "$NREC_QUERY" $TIME >> ${BASE_OUT}/time_summary/new_times_for_modeling/time_summary_"$DESC".txt
echo "$NREC_DB" "$NREC_QUERY" $TIME >> ${TIME_SUMMARY_DIR}/time_summary_"$DESC"_"$SLOTS"cpus.txt


# QUERY_FILE=${BASE_DIR}/orig_query_split/orig_query
# DB_FILE=${BASE_DIR}/orig_db_split/orig_db
# DB_NAME=SRR5713923
# DB_FILE=${BASE_DIR}/orig_db_split/${DB_NAME}
```

batch_run_v2.sh

```
#!/bin/bash
# How to run:
# Specify DBFILE, QUERYFILE, num_threads and NUM_REPEAT below
# And run:
# bash batch_run_v2.sh

LOG_DIR=log
mkdir -p ${LOG_DIR}

D=`date +"%FT%T"`
# OUT="batch_run_log_"${D//:}".log"
OUT="${LOG_DIR}/batch_run_log_"${D//:}".log"

DBFILE="m_vals_2048.txt"
QUERYFILE="n_vals_2048.txt"

LEN_DB=`cat $DBFILE | wc -l`
LEN_QR=`cat $QUERYFILE | wc -l`
TASKS=$((LEN_DB*LEN_QR))

num_threads=(1 2 4 8)
NUM_REPEAT=3

for thr in ${num_threads[@]}; do
    echo "" 2>&1 | tee >> ${OUT}
    echo "Submit jobs with threads: $thr" 2>&1 | tee >> ${OUT}
        MEM=$((22/$thr))
        for (( t=1; t<=${NUM_REPEAT}; t++ ))
        do
                echo "" 2>&1 | tee >> ${OUT}
                echo "Threads: ${thr}, Test: $t" 2>&1 | tee >> ${OUT}

                CMD="qsub -t 1-"${TASKS}" -N "blast_array_${thr}_${t}" -l h_vmem=${MEM}G \
                -pe thread $thr -l dell01=true -l gpus=0 sge_extractblast_v3.sh \
                type1test"$t"_"$thr" $DBFILE $QUERYFILE"

                echo "$CMD" 2>&1 | tee >> ${OUT}
                $CMD
                sleep 1

                CMD="qsub -t 1-"${TASKS}" -N "blast_array_"${thr}"_"${t}"" -l h_vmem=${MEM}G \
                -pe thread "${thr}" -q '*@@betsy_original' sge_extractblast_v3.sh \
                type2test"$t"_"$thr" $DBFILE $QUERYFILE"
                echo "$CMD" 2>&1 | tee >> ${OUT}
                qsub -t 1-"${TASKS}" -N "blast_array_"${thr}"_"${t}"" -l h_vmem=${MEM}G \
                -pe thread "${thr}" -q '*@@betsy_original' sge_extractblast_v3.sh \
                type2test"$t"_"$thr" $DBFILE $QUERYFILE
                sleep 1

                CMD="qsub -t 1-"${TASKS}" -N "blast_array_${thr}_${t}" -l h_vmem=${MEM}G \
                -pe thread $thr -l bigbox sge_extractblast_v3.sh \
                type3test"$t"_"$thr" $DBFILE $QUERYFILE"
                echo "$CMD" 2>&1 | tee >> ${OUT}
```

```
                $CMD
                sleep 1

                CMD="qsub -t 1-"${TASKS}" -N "blast_array_${thr}_${t}" -l h_vmem=${MEM}G \
                -pe thread $thr -l sm01 sge_extractblast_v3.sh \
                type4test"$t"_"$thr" $DBFILE $QUERYFILE"
                echo "$CMD" 2>&1 | tee >> ${OUT}
                $CMD
                sleep 1

                CMD="qsub -t 1-"${TASKS}" -N "blast_array_${thr}_${t}" -l h_vmem=${MEM}G \
                -pe thread $thr -l sm02 sge_extractblast_v3.sh \
                type5test"$t"_"$thr" $DBFILE $QUERYFILE"
                echo "$CMD" 2>&1 | tee >> ${OUT}
                $CMD
                sleep 1

                CMD="qsub -t 1-"${TASKS}" -N "blast_array_${thr}_${t}" -l h_vmem=${MEM}G \
                -pe thread $thr -l hpe01=true -l gpus=0 sge_extractblast_v3.sh \
                type6test"$t"_"$thr" $DBFILE $QUERYFILE"
                echo "$CMD" 2>&1 | tee >> ${OUT}
                $CMD
                sleep 1
        done
done
```

```
#!/bin/bash
# How to run:
# Specify DBFILE, QUERYFILE, num_threads and NUM_REPEAT below
# And run:
# bash batch_run.sh

LOG_DIR=log
mkdir -p ${LOG_DIR}

D=`date +"%FT%T"`
# OUT="batch_run_log_"${D//:}".log"
OUT="${LOG_DIR}/batch_run_log_"${D//:}".log"

DBFILE="m_vals_2048.txt"
QUERYFILE="n_vals_2048.txt"

LEN_DB=`cat $DBFILE | wc -l`
LEN_QR=`cat $QUERYFILE | wc -l`
TASKS=$((LEN_DB*LEN_QR))

num_threads=(1 2 4 8)
NUM_REPEAT=3

for thr in ${num_threads[@]}; do
    echo "" 2>&1 | tee >> ${OUT}
    echo "Submit jobs with threads: $thr" 2>&1 | tee >> ${OUT}
        MEM=$((22/$thr))
        for (( t=1; t<=${NUM_REPEAT}; t++ ))
        do
                echo "" 2>&1 | tee >> ${OUT}
                echo "Threads: ${thr}, Test: $t" 2>&1 | tee >> ${OUT}
                CMD="qsub -t 1-"${TASKS}" -N "blast_array_${thr}_${t}" -l h_vmem=${MEM}G \
                -pe thread $thr -l dell01=true -l gpus=0 sge_extractblast_v3.sh \
                type1test"$t"_"$thr" $DBFILE $QUERYFILE"
                echo "$CMD" 2>&1 | tee >> ${OUT}
                $CMD
                sleep 1

                CMD="qsub -t 1-"${TASKS}" -N "blast_array_"${thr}"_"${t}"" -l h_vmem=${MEM}G \
                -pe thread "${thr}" -q '*@@betsy_original' sge_extractblast_v3.sh \
                type2test"$t"_"$thr" $DBFILE $QUERYFILE"
                echo "$CMD" 2>&1 | tee >> ${OUT}
```

```
                qsub -t 1-"${TASKS}" -N "blast_array_"${thr}"_"${t}"" -l h_vmem=${MEM}G \
                -pe thread "${thr}" -q '*@@betsy_original' sge_extractblast_v3.sh \
                type2test"$t"_"$thr" $DBFILE $QUERYFILE
                sleep 1

                CMD="qsub -t 1-"${TASKS}" -N "blast_array_${thr}_${t}" -l h_vmem=${MEM}G \
                -pe thread $thr -l bigbox sge_extractblast_v3.sh \
                type3test"$t"_"$thr" $DBFILE $QUERYFILE"
                echo "$CMD" 2>&1 | tee >> ${OUT}
                $CMD
                sleep 1

                CMD="qsub -t 1-"${TASKS}" -N "blast_array_${thr}_${t}" -l h_vmem=${MEM}G \
                -pe thread $thr -l sm01 sge_extractblast_v3.sh \
                type4test"$t"_"$thr" $DBFILE $QUERYFILE"
                echo "$CMD" 2>&1 | tee >> ${OUT}
                $CMD
                sleep 1

                CMD="qsub -t 1-"${TASKS}" -N "blast_array_${thr}_${t}" -l h_vmem=${MEM}G \
                -pe thread $thr -l sm02 sge_extractblast_v3.sh \
                type5test"$t"_"$thr" $DBFILE $QUERYFILE"
                echo "$CMD" 2>&1 | tee >> ${OUT}
                $CMD
                sleep 1

                CMD="qsub -t 1-"${TASKS}" -N "blast_array_${thr}_${t}" -l h_vmem=${MEM}G \
                -pe thread $thr -l hpe01=true -l gpus=0 sge_extractblast_v3.sh \
                type6test"$t"_"$thr" $DBFILE $QUERYFILE"
                echo "$CMD" 2>&1 | tee >> ${OUT}
                $CMD
                sleep 1
        done
done
```

failed_rows.sh

```
#!/bin/bash

# Specify DIR and OUTPUT in lines 5 and 6 below.
# Run as:
# bash failed_rows.sh

DIR=/scratch/mikem/UserSupport/trinity.cheng/blast_surface/time_summary
OUTPUT=failed_rows.txt
echo `date` > ${OUTPUT}
echo "Files on directory: $DIR" >> ${OUTPUT}

echo >> ${OUTPUT}
for file in ${DIR}/*; do
        RES=`awk ' NF==2 {print NR,$0} '  $file`
        if [ ! -z "$RES" ]
        then
                echo "Found defetive line(s) in ${file##*/}" >> ${OUTPUT}
                awk ' NF==2 {print NR,$0} '  $file >> ${OUTPUT}
                echo >> ${OUTPUT}
        fi
done

echo "See ${OUTPUT} for results."
```

- This page was last modified on 31 October 2022, at 10:53.