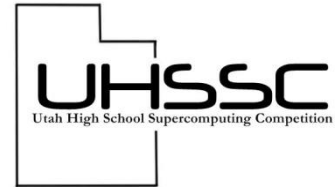


Utah High School Supercomputing Competition Challenge Definitions for 2012

Dan McGuire

Dec 25, 2011 updated Jan 22, 2012



Systems Configuration Challenge

This is a timed challenge. All teams will start at the same time with a wiped system. Instructions for this challenge are available at <http://www.uhssc.org/getting-started.html>.

The following software must be installed:

1. Install CentOS 5.7 on the headnode
2. Set up a local repository on the headnode.
3. Install all seven nodes via PXE boot.
4. Configure NTP all nodes
5. Configure passwordless SSH
6. Configure NFS
7. Configure NIS
8. Install MPICH2
9. Install MPI4PY
10. Run multi-node "Hello World" (see below)

Restrictions/allowances

1. Must use the hardware provided, no substitutions (for example, CD/DVD ROM)
2. Can attach anything to the nodes' USB ports.
3. May use a single laptop linked into the local network.
4. May modify the installed repository as needed.
5. May use a command line version of CentOS on the head node.
6. Cannot PXE boot the head node from the laptop.
7. Laptop must be connected to the local network through the Cisco switch. (This limits speed over the local network to 100 Mb/s ensuring a level playing field.)

Determination of Winner:

1. Finish time will be recorded by judges using a single clock, after the successful completion of the "Hello, World" program. All eight configured nodes MUST report with the "Hello, World" program. The order of reporting is not important.
2. Installation of system will be verified by the judges.
3. Irregularities in the installation must be corrected before a finish time can be recorded.
4. Earliest finish time wins.

"Hello, World" Program:

```
#!/usr/bin/env python
from mpi4py import MPI
```

```
size = MPI.COMM_WORLD.Get_size()
rank = MPI.COMM_WORLD.Get_rank()
name = MPI.Get_processor_name()
print("Hello, World! I am process %d of %d on %s." % (rank, size,
name))
```

The Systems Configuration Challenge is the second longest, taking about 1-2 hours. All teams must finish their system configuration as it is required for the second Software Optimization Challenge.

Software Optimization Challenge

The Software Optimization Challenge is done by the teams at their home locations. It will involve a multiprocessor implementation of a Word Frequency Analysis Program provided to the teams. The teams must then determine how to distribute the workload amongst their eight nodes to minimize execution time. The execution times of these implementations will be then evaluated at the UHSSC competition

An example of a single node Word Frequency Analysis program is provided on the UHSSC website, as well as part of this document. This program takes about 110 seconds to execute on a single head node processor, and it generates a list of words with the number of times that they occur. The output of this program will act as a standard for judging the “correctness” of the teams’ implementations.

Restrictions/allowances:

1. No hardware in addition to what was initially provided.

Determination of Winner:

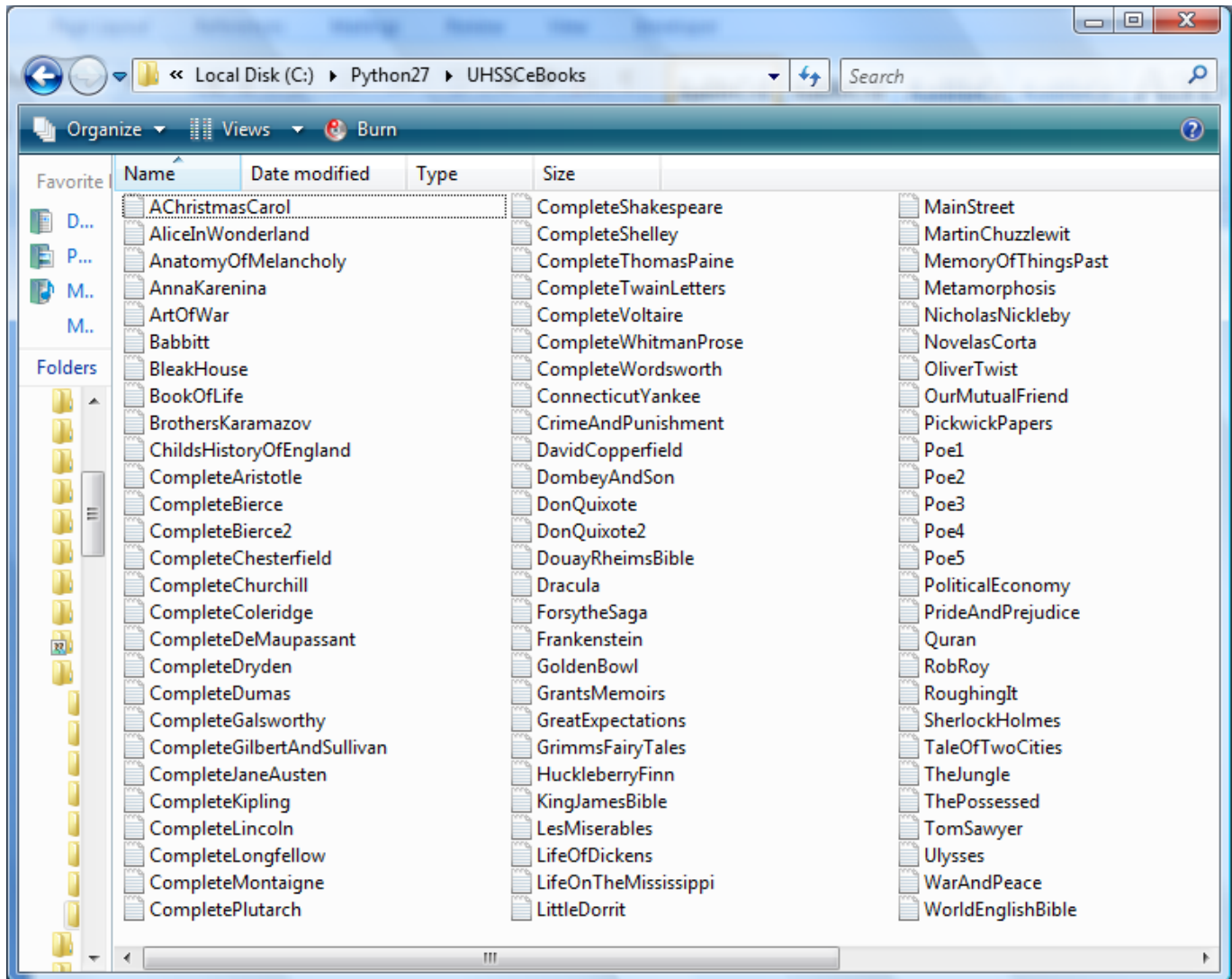
1. The reference Word Frequency program must be run first, and the duration to the nearest millisecond recorded*. This is the REFERENCE duration.
2. The team’s Word Frequency implementation must record the execution duration in a method identical to the reference program. The duration of execution will then be recorded to the nearest millisecond**. This is the TEAM duration.
3. The output of the teams’ programs will be deemed correct if a “diff” command comparing team’s output to the reference program’s yields “no differences”.
4. The team with the largest ratio of REFERENCE/TEAM durations wins.

The teams will be performing the Word Frequency analysis on a selection of eBooks downloaded from Project Gutenberg. These are available on the UHSSC website. The size of the library of eBooks is approximately 140 Mbytes.

* The REFERENCE duration is measured by averaging ten trials. If any one of the trials varies from the mean by more than two standard deviations, then that trial will be discarded and re-done.

** The TEAM duration is measured by averaging twenty trials. If any one of the trials varies from the mean by more than two standard deviations, then that trial will be discarded and re-done.

The UHSSC eBook Library for the Software Optimization Challenge:



```

#
# UHSSC word frequency reference code
#
# see http://www.daniweb.com/software-
development/python/code/216747
#
import re
import time

# here's the list of entries in the UHSSC eBooks Library
# thanks to Project Gutenberg www.gutenberg.org
filenames = [ "./UHSSCeBooks/AChrisTmasCarol.txt",
               "./UHSSCeBooks/AliceInWonderland.txt",
               "./UHSSCeBooks/Dracula.txt",
               "./UHSSCeBooks/Frankenstein.txt",
               "./UHSSCeBooks/GrimmsFairyTales.txt",
               "./UHSSCeBooks/HuckleberryFinn.txt",
               "./UHSSCeBooks/Metamorphosis.txt",
               "./UHSSCeBooks/LesMiserables.txt",
               "./UHSSCeBooks/AnnaKarenina.txt",
               "./UHSSCeBooks/DonQuixote.txt",
               "./UHSSCeBooks/DonQuixote2.txt",
               "./UHSSCeBooks/NovelasCorta.txt",
               "./UHSSCeBooks/PrideAndPrejudice.txt",
               "./UHSSCeBooks/Ulysses.txt",
               "./UHSSCeBooks/TaleOfTwoCities.txt",
               "./UHSSCeBooks/TomSawyer.txt",
               "./UHSSCeBooks/GreatExpectations.txt",
               "./UHSSCeBooks/OliverTwist.txt",
               "./UHSSCeBooks/DavidCopperfield.txt",
               "./UHSSCeBooks/BleakHouse.txt",
               "./UHSSCeBooks/PickwickPapers.txt",
               "./UHSSCeBooks/NicholasNickleby.txt",
               "./UHSSCeBooks/LittleDorrit.txt",
               "./UHSSCeBooks/OurMutualFriend.txt",
               "./UHSSCeBooks/MartinChuzzlewit.txt",
               "./UHSSCeBooks/DombeyAndSon.txt",
               "./UHSSCeBooks/ChildsHistoryOfEngland.txt",
               "./UHSSCeBooks/KingJamesBible.txt",
               "./UHSSCeBooks/PoliticalEconomy.txt",
               "./UHSSCeBooks/WorldEnglishBible.txt",
               "./UHSSCeBooks/MemoryOfThingsPast.txt",
               "./UHSSCeBooks/Quran.txt",
               "./UHSSCeBooks/ConnecticutYankee.txt",
               "./UHSSCeBooks/RoughingIt.txt",
               "./UHSSCeBooks/LifeOnTheMississippi.txt",
               "./UHSSCeBooks/Poe1.txt",
               "./UHSSCeBooks/Poe2.txt",
               "./UHSSCeBooks/Poe3.txt",
               "./UHSSCeBooks/Poe4.txt",
               "./UHSSCeBooks/Poe5.txt",
               "./UHSSCeBooks/CompleteShakespeare.txt",

```

```

"/UHSSCeBooks/CompleteMontaigne.txt",
"/UHSSCeBooks/CompleteDeMaupassant.txt",
"/UHSSCeBooks/GrantsMemoirs.txt",
"/UHSSCeBooks/CompleteJaneAusten.txt",
"/UHSSCeBooks/ForsytheSaga.txt",
"/UHSSCeBooks/RobRoy.txt",
"/UHSSCeBooks/CompleteLongfellow.txt",
"/UHSSCeBooks/CompleteColeridge.txt",
"/UHSSCeBooks/CompleteShelley.txt",
"/UHSSCeBooks/CompleteDumas.txt",
"/UHSSCeBooks/CompleteAristotle.txt",
"/UHSSCeBooks/CompleteChurchill.txt",
"/UHSSCeBooks/LifeOfDickens.txt",
"/UHSSCeBooks/CompleteChesterfield.txt",
"/UHSSCeBooks/GoldenBowl.txt",
"/UHSSCeBooks/CompleteLincoln.txt",
"/UHSSCeBooks/CompleteGilbertAndSullivan.txt",
"/UHSSCeBooks/CompleteThomasPaine.txt",
"/UHSSCeBooks/DouayRheimsBible.txt",
"/UHSSCeBooks/CompleteTwainLetters.txt",
"/UHSSCeBooks/CompleteWhitmanProse.txt",
"/UHSSCeBooks/CompleteGalsworthy.txt",
"/UHSSCeBooks/CompletePlutarch.txt",
"/UHSSCeBooks/CompleteVoltaire.txt",
"/UHSSCeBooks/CompleteBierce.txt",
"/UHSSCeBooks/CompleteBierce2.txt",
"/UHSSCeBooks/CompleteWordsworth.txt",
"/UHSSCeBooks/CompleteDryden.txt",
"/UHSSCeBooks/CompleteKipling.txt",
"/UHSSCeBooks/ArtofWar.txt",
"/UHSSCeBooks/TheJungle.txt",
"/UHSSCeBooks/Babbitt.txt",
"/UHSSCeBooks/BookOfLife.txt",
"/UHSSCeBooks/MainStreet.txt",
"/UHSSCeBooks/BrothersKaramazov.txt",
"/UHSSCeBooks/ThePossessed.txt",
"/UHSSCeBooks/CrimeAndPunishment.txt",
"/UHSSCeBooks/AnatomyOfMelancholy.txt",
"/UHSSCeBooks/WarAndPeace.txt"]

```

```

# must record Start Time as required by UHSSC competition
StartTime=time.time()
print "Start time: ",StartTime," seconds"

```

```

# create dictionary of word:frequency pairs
freq_dic = {}

```

```

for filename in filenames:

```

```

    # read eBook and split into words
    print "Analyzing eBook: ",filename
    word_list = re.split('\s+', file(filename).read()).lower()

```

```

print 'Words in this eBook:', len(word_list)

# punctuation marks and numbers to be removed
punctuation = re.compile(r'[.?!&_,";'\--
)0123456789%$\\\/=#\[\\]')

# remove punctuation/numbers and make an entry into the
dictionary
for word in word_list:
    word = punctuation.sub("", word)
    if len(word)==0: continue
    try:
        freq_dic[word] += 1
    except:
        freq_dic[word] = 1

print 'Unique words so far:', len(freq_dic)

# create list of (key, val) tuple pairs
freq_list = freq_dic.items()
# sort by key or word
freq_list.sort()

# output to text file
OutputFile = open('UHSSCWordFrequency.txt', 'w')
for word, freq in freq_list:
    s=word+", "+repr(freq)+"\n"
    OutputFile.write(s)
OutputFile.close()

# record end time and compute duration
EndTime=time.time()
print "End time: ",EndTime," seconds"
Duration = EndTime-StartTime
print "Duration: ",Duration," seconds"

```

The following (last) page should not be released to the teams. It describes the “Hotseat” Competition and it will be released at the event on March 28th.