UN5390: Scientific Computing I

Fall 2016

Assignment #12 (Due 11:59 am on Sunday, 11th Dec 2016; 20%)

Guidelines

1. Every problem in this assignment has a value beyond its numerical score. It is either a *swan kick* moment for what you already know OR a *wax on, wax off* training for something you will learn soon OR a combination thereof. Treating every problem as a tiny research project is very highly encouraged

2. You may seek help from anyone as long as you understand all such help and cite it appropriately. Write the programs and scripts in compliance with programming etiquette discussion as clarity and readability account for 25% of the assigned credit

3. Read through/Review the course material and additional references, if necessary. Give yourself enough time to complete the assignment on or ahead of time, and show continuous improvement. Check the submission for spelling and grammatical correctness. Contact the instructor if you didn't understand the material

4. Typeset your solutions/answers in `${USER}_12.tex`, `${USER}_12.pdf`, and other files as indicated throughout this assignment. Include imagery to explain your answers as and when necessary. Do not include the source code or data in the `${USER}_12.tex` document. There needs to be at least one commit per problem to the GitHub repository to show timely work, demonstrate your work ethic, and a final tagged version of the assignment for submission to earn any credit

5. Graduate students can solve any three (worth 6.67% each) while undergraduate and non-traditional students can solve any two problems (worth 10% each). Remaining problems represent an opportunity to *do a little more* and are worth 0.50% each. Deadline for seeking help from the instructor: 7:59 am on Friday, 9th Dec 2016

Assignment submission workflow

1. Replace `#1` with the appropriate problem number. Keep all files and folders under
   `${UN5390}/CourseWork/Week_12/${USER}_12/`

2. Before starting to work on a problem (or part thereof)

   ```
   cd ${UN5390}
   git pull
   ```

   Suggestions to correct/improve your answers, if any when requested by you, can be found in `${UN5390}/CourseWork/Week_12/suggested_corrections_12.txt`

3. After each problem (or a part thereof), commit to the repository

   ```
   cd ${UN5390}/LaTeXTemplates/Course/
   git pull
   git add ${USER}.bib
   cd ../../CourseWork/Week_12/
   git add ${USER}_12
   git commit -m "Submitting problem #1 in assignment #12"
   git push origin master
   ```

4. When the assignment in its entirety is ready for final submission

   ```
   cd ${UN5390}/CourseWork/Week_12/
   git pull
   git add ${USER}_12
   git tag -a a12 -m "Submitting assignment #12"
   git push origin a12
   ```

5. If you made edits to your submission after tagging but before the final due date, add a minor revision number to the tag

   ```
   cd ${UN5390}/CourseWork/Week_12/
   git pull
   git add ${USER}_12
   git tag -a a12.1 -m "Submitting assignment #12"
   git push origin a12.1
   ```

   and so on

Problem 1

Top 500 [1] publishes the list of the 500 most powerful supercomputers (known within the computing community as *The List*) twice each year: once during ISC [2] held in Europe in June, and once during SC [3] held in the US in November. HPL benchmark, discussed as part of the matrix methods, forms the basis of this list [4–6].

Prepare a table of the top 10 from the November 2016 edition of this list (refer to the course material for which fields to include). Write two paragraphs with 4-6 short and meaningful sentences about any two supercomputers of your choice detailing their parent institution, intended audience and usage, and cost of acquisition, deployment and operation, etc.

Problem 2

Suppose that you are an experienced researcher at an advanced degree granting institution with an excellent track record of research, teaching and synergestic activities. Your current computing infrastructure includes access to a HPC cluster with 1024 processors with 4 GB RAM per processor, 56 Gbps InfiniBand network and 64 TB archival quality storage. Suppose that the research problem of your interest can scale well and has a size N when modeled as a computer program. This problem requires time t to solve when the aforementioned infrastructure is used with an uninhibited solo access. At least four runs are necessary to verify the results and publish them in a peer-reviewed journal.

Further suppose that an anonymous donor has decided to fund your research covering all (i.e., hardware, software, personnel, etc.) expenses. As a result, you now have an uninhibited access to a brand new HPC cluster with 8192 processors (10% faster than the existing ones) with 16 GB RAM per processor, 100 Gbps InfiniBand network and 1 PB archival quality storage along with 128 TB fast scratch space.

In the context of the videos seen throughout the semester showcasing the impact of supercomputing [7–18] as well as your research needs/experience, if any, answer the following: would you rather solve a problem of size N in approximately 1/8th the time OR tackle a problem of size $8N$ in approximately the same amount of time OR some combination in between? Does your answer change if you had tenure and you couldn't lose your job?

Discuss the answer (or a path thereto) with your advisor and attempt to include her/his thoughts. If you do not have an advisor and/or any research experience, try to formulate your answer assuming that t for each run is 90 days (i.e., CPU time of $90 \times 24 \times 1024$ hours). The answer, in its entirety (i.e., including images, tables, mathematical expressions, etc., if needed), must at least be one page (single side) and must not exceed three pages. Arrange your thoughts into easily readable paragraphs with short and meaningful sentences.

Problem 3

You wrote a program to perform matrix multiplication in the previous assignment. The definition used to populate participating matrices, $A\,(N \times N)$ and $B\,(N \times 1)$, was as follows.

$$a_{ij} = i \times (N - i - 1) \times j \times (N - j - 1)$$

$$b_{ij} = i + j + 2$$

Copy the program, `matrix_multiply.EXTN`, as `matrix_multiply_p.EXTN` and parallelize it. You may use OpenMP for C/C++, `parfor` loop in MATLAB, etc. as demonstrated during in-class discussions and sample programs. Run the parallel program for $N = 64, 128$ using `NPROC` = 1, 2, 4, 8, 16. Plot and explain the raw as well as scaled performance of this program. You may use aptly named Gnuplot (or other program but not Microsoft Excel) files if your programming language of choice does not provide implicit visualization.

Problem 4

You have been hired as a new tenure-track faculty in a research oriented university with teaching responsibilities. You will have two graduate students to work with but the department covers their tuition and stipend. Well-defined projects, quality publications, good evaluations from students about teaching techniques, and externally funded proposals are necessary – but might not be sufficient – for promotion.

Translate your computational start-up budget into computing infrastructure (standalone workstations, laptops, front end, login node, storage node, compute nodes, switching system, racks, cables, and software) so as to *comfortably* pursue your research activities for the first two years (i.e., first review of progress towards tenure-track) based on the following scenarios:

1. $35,000 in a small sized university (e.g., Valparaiso University)

   There is no centralized IT and the severely under-staffed localized data centers provide racks but do not have dual power supply or a data backup plan. University does not own a shared computing cluster or research software licenses.

   Following charges are applied per month: network connection fee ($25), systems administration fee ($75) and data center fee ($50).

2. $75,000 in a medium sized university (e.g., Michigan Tech, pre-2013)

   University has centralized IT and licenses for most commonly used software. Data centers have dual power supply and racks are provided at no cost. Except for monthly network connection fee ($25), there is no additional fee. However, there is no shared computing cluster or a data backup plan

3. \$125,000 in a large university (e.g., The University of Michigan)

   University offers localized IT support on a per school basis (college of arts and science, college of engineering, etc.). Data centers have dual power supply and licenses for most commonly used software are available. A shared computing cluster and data backup plan exist as well

   (a) Should you choose to use the university cluster, following charges will be applied: \$5 per user account creation (one time), \$0.05 per CPU core per hour of usage per user and \$250 per TB per year

   (b) Should you choose to own a computing cluster, you will need to buy every component and following monthly charges will be applied: network connection fee (\$15), systems administration fee (\$50), data center fee (\$35). Data backup is available at \$250 per TB per year

You have the freedom to purchase anything that aids your research/teaching endeavors but the department chair must approve, in writing, every purchase exceeding \$5,000. Approximate cost of components is as follows:

1. Workstation (Intel E5-2698 2.20 GHz 20 processors, 128 GB RAM, two 256 GB RAID1 SSD for OS, two 1 TB RAID1 data drive and single power supply) with 24" monitor, keyboard and mouse $\simeq$ \$8,000

2. Front end (Intel E5-2698 2.20 GHz 40 processors, 256 GB RAM, two 480 GB RAID1 SSD for OS, two 2 TB RAID1 data drive and dual power supply) $\simeq$ \$10,000

3. Storage node (Intel E5-2698 2.20 GHz 40 processors, 512 GB RAM, two 250 GB RAID1 SSD for OS, dual power supply, 96 TB raw and 80 TB usable research space with RAID60) $\simeq$ \$25,000

4. Compute node (Intel Xeon E5-2698 2.20 GHz 40 processors, 512 GB RAM, 480 GB Intel Enterprise SSD, 56 Gbps InfiniBand network) $\simeq$ \$15,000

5. Ethernet switch (Netgear 8 port 1 Gbps) $\simeq$ \$750
   Ethernet switch (Netgear 16 port 1 Gbps) $\simeq$ \$1,000
   Ethernet switch (Netgear 48 port 1 Gbps) $\simeq$ \$3,000
   InfiniBand switch (Mellanox 18 port 56 Gbps) $\simeq$ \$7,000
   InfiniBand switch (Mellanox 36 port 56 Gbps) $\simeq$ \$10,000
   Fabric manager license (required with InfiniBand) $\simeq$ \$250 per compute node
   Ethernet cable ($3m$) $\simeq$ \$5
   InfiniBand cable, copper ($3m$) $\simeq$ \$125; optical fiber ($3m$) $\simeq$ \$250
   48U rack/cabinet with dual power supply $\simeq$ \$5,000

Problem 5

Throughout the semester, you have been exposed to the theory and practice of estimating FLOPS as a measure of performance of a computing infrastructure. For a computer with N identical/homogeneous processors, it is given by

$$\text{FLOPS} \;=\; \text{N} \times \text{CPU speed} \times \frac{\text{FLOPs}}{\text{CPU cycle}}$$

Search the online literature, and identify `FLOPs/CPU cycle` information about your personal laptop/workstation, and compute the theoretical maximum value (R_{Theory}).

If your personal laptop/workstation has Intel [19] architecture, then download the *Intel Optimized LINPACK Benchmark* from Intel Math Kernel Library Benchmarks (note: you may need to create an account in the Intel Developer Zone; use Michigan Tech email while creating one). Using one of the very first *free time exercises* demonstrated in class as a starting point, identify the order of a square matrix that fits within 80%, 85% and 90% of the total RAM available in your system. Run the HPL benchmark for these matrix sizes while ensuring all other unnecessary programs/services are turned off, and note down the results as R_{Expt}. You are responsible for ensuring the safety of your system while running this benchmark.

Write a note comparing R_{Theory} and best of R_{Expt} noting that the latter is usually 85-90% of the former. If you couldn't find the `FLOPs/CPU cycle` information from the literature, use R_{Expt} to reverse engineer and estimate that value.

References

[1] Top500.
`https://top500.org`.

[2] ISC, The Event for High Performance Computing, Networking and Storage
`http://isc-hpc.com`.

[3] SC, The International Conference for High Performance Computing, Networking, Storage and Analysis
`http://sc16.supercomputing.org`.

[4] J. Dongarra, C. Moler, J. Bunch, and G. Stewart. *LINPACK Users' Guide*. Society for Industrial and Applied Mathematics, 1979.

[5] J. Dongarra, P. Luszczek, and A. Petitet. The LINPACK Benchmark: Past, Present and Future. *Concurrency and Computation: Practice and Experience*, 15:803, 2003.

[6] A. Petitet, C. Whaley, J. Dongarra, A. Cleary, and P. Luszczek. High Performance Computing Linpack Benchmark (HPL) 2.2, 2016.
`http://www.netlib.org/benchmark/hpl/`.

[7] Impact of Supercomputing: Introduction.
`https://www.youtube.com/watch?v=9m0gZ2Gft4Q`.

[8] Impact of Supercomputing: Aerospace.
`https://www.youtube.com/watch?v=jhApQIPQquw`.

[9] Impact of Supercomputing: Batteries.
`https://www.youtube.com/watch?v=1o4YViBAGU0`.

[10] Impact of Supercomputing: Climate Modeling.
`https://www.youtube.com/watch?v=RSQg_URCHKI`.

[11] Impact of Supercomputing: Diapers & Detergents.
`https://www.youtube.com/watch?v=PuCx50FdSic`.

[12] Impact of Supercomputing: Entertainment.
`https://www.youtube.com/watch?v=-7Bpo1Quxyw`.

[13] Impact of Supercomputing: Epilepsy & Parkinson's Treatments.
`https://www.youtube.com/watch?v=HOEAocq_U5o`
`https://www.youtube.com/watch?v=qx2dRIQXnbs`.

[14] Impact of Supercomputing: Human-Induced Climate Change.
`https://www.youtube.com/watch?v=QABMJapCIXo`.

[15] Impact of Supercomputing: Missing Plane.
`https://www.youtube.com/watch?v=CTUInP-7gfE`.

[16] Impact of Supercomputing: PayPal.
`https://www.youtube.com/watch?v=rBLL1uyP3ic`.

[17] Impact of Supercomputing: Precision Medicine.
`https://www.youtube.com/watch?v=PgqwjnavI78`.

[18] Impact of Supercomputing: Storm Prediction.
`https://www.youtube.com/watch?v=dbFWX_2HIy4`.

[19] Intel Corporation.
`http://intel.com`.