ASC Student Supercomputer Challenge (2018)

Preliminary Contest Notifications

Dear Participating Teams:

Thank you very much for participating in the ASC Students Supercomputer Challenge 2018 (ASC18). This document will provide detailed information about the preliminary round of the contest.

1. About the Preliminary Round

In the preliminary round, each registered team is required to submit a set of documents that include a proposal, optimized source code files and output files (detailed requirements specified in Appendix A). The proposal needs to be written in English, and will be reviewed by the ASC evaluation committee.

2. Submission Guideline

All teams should make their submissions to info@asc-events.org before 8:00 AM, March 13th, 2018(UTC/GMT +8:00). The confirmation of your submission will be sent to you by email. The submission should include the following items:

a) The proposal file (in .doc or .pdf format), named with the university or college name and the contact person name (e.g. AAAUniversity_BBB.doc).

b) The additional files should be compressed into one file (e.g. AAAUniversity_BBB.zip, other compression formats are also OK) except the requested RELION files (detailed requirements specified in Appendix A). The compressed file should at least include (detailed requirements specified in Appendix A):

- Output files of HPL
- Output files of HPCG
- Required files of Answer prediction for Search Query

3. For any further inquiries about the contest, please contact the ASC committee through the following emails:

a) Technical support: techSupport@asc-events.org
b) Contest organization: info@asc-events.org
c) News and media: media@asc-events.org

We wish you the best of luck at ASC18.

ASC18 Committee
01-16-2018

Appendix A: Proposal Requirements

I. A brief background description of the university's or the department's supercomputing activities (5 points)

1. Supercomputing-related hardware and software platforms
2. Supercomputing-related courses, trainings, and interest groups
3. Supercomputing-related research and applications
4. A detailed description of the key achievements on supercomputing research (no more than 2 items), attached with proof materials (published papers, award certificates, etc.)

II. Introduction of the Team (5 points)

1. Brief description of the building process of your team
2. Brief introduction of each team member (including group photos of the team)
3. Your team slogan.

III. Technical proposal requirements (90 points)

1. Design of your HPC system (15 points)

a) Within the 3,000-watt power budget, your system should be designed to achieve the best computing performance.

b) Specify your system's software and hardware configuration and interconnection. Describe the power consumption, evaluate the performance, and analyze the advantages and disadvantages of your proposed architecture.

c) Your system should be based on the Inspur NF5280M5 server. The components listed in the table below will be provided by Inspur to the teams that enter the final (The configuration may be changed). Other components (except the server itself) are acceptable, but should be prepared by the teams at their own costs. For example, you can change the number of NF5280M5 servers and accelerators, the type of the hard disk and memory, and even the type of the Ethernet in your proposed configuration.

Item	Name	Configuration
Server	Inspur NF5280M5	CPU: Intel Xeon Gold 6132 x 2，2.6GHz，14 cores Memory: 16G x 12，DDR4，2666Mhz Hard disk: 1T SATA x 1 *Power consumption estimation:* *6132 TDP 140W, memory 7.5W, hard disk 10W*

HCA card	FDR	Infiniband Mellanox ConnectX®-3 HCA card, single port QSFP, FDR IB *Power consumption estimation:9W*
Switch	GbE switch	10/100/1000Mb/s，24 ports Ethernet switch *Power consumption estimation:30W*
	FDR-IB switch	SwitchX™ FDR InfiniBand switch, 36 QSFP port *Power consumption estimation:130W*
Cable	Gigabit CAT6 cables	CAT6 copper cable, blue, 3m
	Infiniband cable	Infiniband FDR optical fiber cable, QSFP port, cooperating with the Infiniband switch for use

2. HPL and HPCG (15 points)

The proposal should include descriptions of the software environment (operating system, complier, math library, MPI software, software version, etc.), the testing method, performance optimization methods, performance estimation, problem and solution analysis, etc. In-depth analysis on HPL algorithm and the source code is recommended.

The HPL software can be downloaded at http://www.netlib.org/benchmark/hpl/.
The HPCG software can be downloaded at https://github.com/hpcg-benchmark/hpcg

Successful verification and optimization of HPL and HPCG on X86 CPU or GPU platforms are recommended. However, teams that cannot access the platform and have to use their own hardware platforms are encouraged to submit their analysis and results.

3. The RELION Test (30 points)

a) Application background

Since it's established, structural biology has played an important role in biological researches. Structural biology attempts to interpret biological processes by solving high resolution bio-macromolecular structures. Normally, there are three approaches to determine the 3D structures of bio-macromolecules, including X-ray crystallography, cryo-electron microscopy (cryo-EM), nuclear magnetic resonance (NMR). In recent years, with the technology innovations, especially the development of direct electron detection camera and sophisticated image processing algorithm, cryo-EM has become the most important tool to study the 3D structures of bio-macromolecules in near atomic resolution. The technique is also called cryo-EM single particle analysis (SPA), that starts from the cryo-vitrification of bio-macromolecular solution and needs to collect thousands of high quality cryo-EM micrographs in a high-throughput way. The subsequent image processing includes micrograph correction (motion and distortion correction, dose weighting) and evaluation, contrast transfer function (CTF) estimation, particle picking and sorting, 2D and 3D classification, orientation refinement and reconstruction, and post-processing (map sharpening). Since the limited illumination dose yields a very noisy raw image of bio-macromolecules that are embedded in

vitreous ice, ten thousands of particle images are needed to increase signal noise ratio (SNR). The basic physics and principle of image processing has been fully described in Reference [1].

Currently, the most popular open-source software to process cryo-EM SPA data is RELION that utilizes the maximum likelihood and Bayesian statistics to overcome the ambiguity of orientation determination of each particle raised by low SNR. The mathematical principle and algorithm of RELION have been fully described in References [2] and [3]. The latest version of RELION is **2.1 Stable** and its source code available in Reference [4]. More information about installation and usage of RELION can be found from Reference [5].

Here we provide a cryo-EM SPA dataset of human apo-ferritin that is a protein complex storing iron in the cell. The raw particles have been picked out from original raw micrographs and stacked into difference MRC files (*.mrcs). The imaging condition and estimated CTF parameters of each particle has been listed in a STAR file (particles.star). An initial low resolution structure of the human apo-ferritin is given (run_ct24_class001.mrc). This challenge is to try to use RELION 2.1to perform image analysis including 2D (step 1), 3D classification (step 2), and the final 3D reconstruction (step 3). Please refer to the supplemental instructions for detailed information.

References:

[1]. Orlova, E. V. & Saibil, H. R. Structural analysis of macromolecular assemblies by electron microscopy. Chem Rev 111, 7710-7748, doi:10.1021/cr1003[illegible]2011).
[2]. Scheres (2012) J. Mol. Biol. (PMID: 22100448)
[3]. Scheres (2012) J. Struct. Biol. (PMID: 23000701)
[4]. Relion source code: https://github.com/3dem/relion/releases (version 2.1 Stable)
[5]. Relion wiki: http://www2.mrc-lmb.cam.ac.uk/relion/index.php/Main_Page

b) Introduction to the test command

In this challenge, all participants are encouraged to complete the computation, obtain the right rights and make efforts to reduce the computational needs, both in time and resources. The proposal document should include descriptions of the software environment (operating system, complier, math library, MPI software and RELION version, etc.), the testing method, performance optimization methods, performance estimation, problem and solution analysis, etc. In-depth analysis into RELION's algorithm and source code is highly encouraged. The detailed tasks and requirements of this challenge are listed below.

1. Compile and install RELION and run the program against the given data according to the instructions from Step 1 to 3. Please submit files generated at the final iteration, screen output of each step and command-line of each step. which should be compressed into a tar.gz file(See the table below). To be noted that, the files produced in the final iteration of Step 3 do not come along with 'ite' marks.

	Compressed file name	contents
Step 1	Step1.Class2D.tar.gz	_it025_classes.mrcs _it025_data.star _it025_model.star

		_it025_optimiser.star _it025_sampling.star Command line file(*.sh) Screen output(*.log)
Step 2	Step2.Class3D.tar.gz	_it040_class001_angdist.bild _it040_class001.mrc _it040_class002_angdist.bild _it040_class002.mrc _it040_class003_angdist.bild _it040_class003.mrc _it040_class004_angdist.bild _it040_class004.mrc _it040_data.star _it040_model.star _it040_optimiser.star _it040_sampling.star Command line file(*.sh) Screen output(*.log)
Step 3	Step3.Refine3D.tar.gz	_model.star _sampling.star _data.star _class001_angdist.bild _class001.mrc Command line file(*.sh) Screen output(*.log)

The requested RELION files should be uploaded into Baidu SkyDrive or Microsoft OneDrive and copied the download links with MD5 codes of upload files into your proposal (with password if needed).

2. Submit a description and summary into the proposal with what kind of computational resources (configuration and architecture) you use to perform the computation, and how long it takes for each step (submission of a log file is encouraged). You may also describe how you compile the package and whether you perform some modifications of the code, how and why.

3. Describe the strategies you think (or have performed) that can reduce the computational needs of this kind of computational challenges.

4. **Notice that** each parameter with its' value we provided is necessary for ensuring the consistency of results. But parameters about application performance, you can use them according your platform, we didn't show them in this instruction.

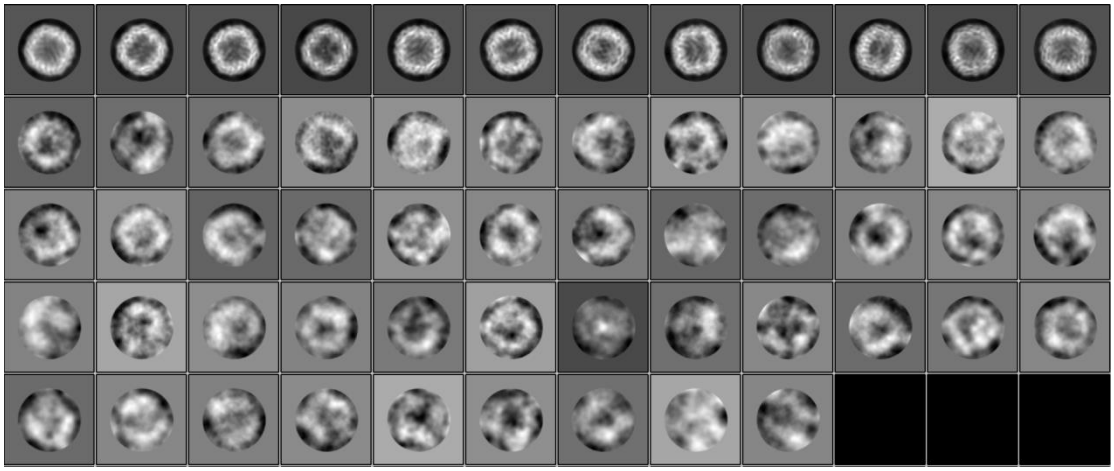
Step 1.

Run 2D classification of the raw particle images. Command can be like the following.

"which relion_refine_mpi" --o Class2D/job007/ --i particles.star --ctf --iter 25 --tau2_fudge 2 --particle_diameter 150 --K 100 --flatten_solvent --zero_mask --strict_highres_exp 8 --oversampling 1 --psi_step 12 --offset_range 5 --offset_step 2 --norm --scale

反向的类分布

The output should be like the following (displayed by reverse sort of class distributions):

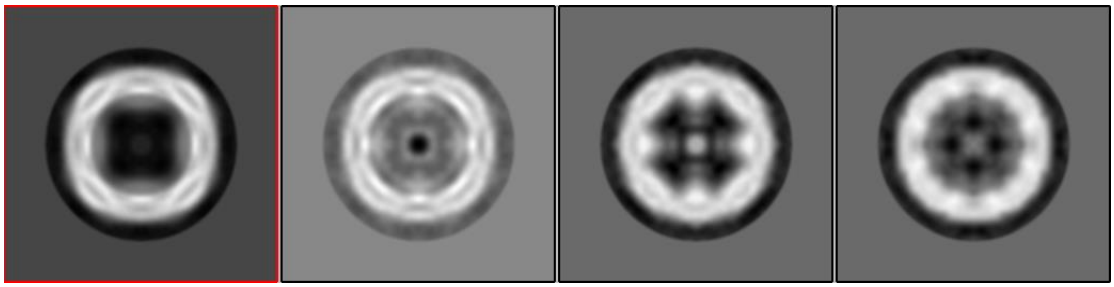


Step 2.

Run 3D classification of the raw particle images starting from an initial reference model. Command can be like the following.

"which relion_refine_mpi" --o Class3D/job007/ --i particles.star --ref run_ct24_class001.mrc --firstiter_cc --ini_high 40 --ctf --iter 40 --tau2_fudge 4 --particle_diameter 150 --K 4 --flatten_solvent --zero_mask --strict_highres_exp 10 --oversampling 1 --healpix_order 1 --sigma_ang 0.3 --offset_range 5 --offset_step 2 --sym O --norm --scale

The output should be like the following (displayed by reverse sort of class distribution):

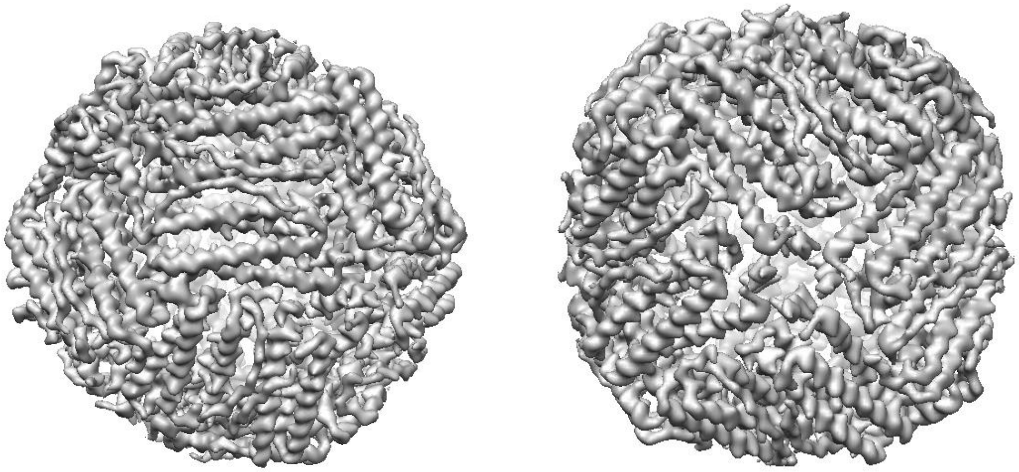


Step 3.

Perform 3D reconstruction and refinement starting from an initial reference model. Command can be like the following.

```
"which relion_refine_mpi" --o Refine3D/job007/ --auto_refine --split_random_halves --i particles.star --ref run_ct24_class001.mrc --firstiter_cc --ini_high 40 --ctf --particle_diameter 150 --flatten_solvent --zero_mask --oversampling 1 --healpix_order 1 --auto_local_healpix_order 5 --offset_range 5 --offset_step 2 --sym O --low_resol_join_halves 40 --norm --scale
```

The output is the final reconstructed 3D structure of human apo-ferritin (.mrc file) with a resolution of ~3.3 angstrom, which can be visualized by using UCSF Chimera (https://www.cgl.ucsf.edu/chimera/) with the threshold value of 0.03 (see below).



The RELION workloads can be downloaded from the following websites:
Microsoft OneDrive: https://1drv.ms/f/s!AkxzxKN8axg2xCptYHmT1QMN_iQ-
or
Baidu SkyDrive: https://pan.baidu.com/s/1nwpZFHv **Password:** vsyh
(Data on the two websites are the same. You can choose anyone you want to download.)

4. Answer Prediction for Search Query (30 points)

Task

Building intelligent agents with the ability for reading comprehension (RC) or open-domain question answering (QA) over real world data is a major goal of artificial intelligence. Such agents can have tremendous value for consumers because they can power personal assistants such as Cortana, Siri, Alexa, or Google Assistant, all of which have been facilitated by recent advances in deep speech recognition technology. As these types of assistants rise in popularity, consumers are finding it more convenient to ask a question and quickly get an answer through voice assistance as opposed to navigating through a search engine result page and web browser. Intelligent agents with RC and QA abilities can also have incredible business value by powering bots that automate customer service agents for business found through messaging or chat interfaces.

MSMARCO is a large scale real-world reading comprehension dataset that target for the tasks above. The questions in the dataset are real anonymized queries issued through Bing or Cortana and the documents are related web pages which may or may not be enough to answer the question. For every question in the dataset, we have asked a crowdsourced worker to answer it, if they can, and to mark relevant passages which provide supporting information for the answer. If they can't answer it, we consider the question unanswerable and we also include a sample of those in MSMARCO.

The answer is strongly encouraged to be in the form of a complete sentence, so the workers may write a longform passage on their own. MS MARCO includes 100,000 questions, 1 million passages, and links to over 200,000 documents. The task need the user find the best algorithm that can find the best answer (passages) from all candidates (passages) or compose the best answer (passage) by itself.

Dataset

Format

Dataset formatted using JSON, each example has five fields below:

Field	Definition
Query	Question query real users issued to the Bing search engine.
Passages	Top 10 contextual passages extracted from public Web documents to answer
Document URLs	URLs for the top documents ranked for the query. These documents are the sources for the contextual passages.
Answer(s)	Synthesized answers from human judges for the query, automatically extracted passages and their corresponding public Web documents.
Segment	QA classification tag. E.g., tallest mountain in south America belongs to the ENTITY segment because the answer is an entity (Aconcagua).

Note: the task ONLY target for the description queries (example that with "query_type": "description")

The challenge will use two different datasets for preliminary and finals. Please note that only Training and Dev sets have labels.

Phase1 (Preliminary)

Training Set	https://msmarco.blob.core.windows.net/msmarco/train_v1.1.json.gz
Dev Set	https://msmarco.blob.core.windows.net/msmarco/dev_v1.1.json.gz
Test Set	https://msmarco.blob.core.windows.net/msmarco/test_public_v1.1.json.gz

Phase2 (Finals)

Training Set	Will announce at the day of finals
Dev Set	Will announce at the day of finals
Test Set	Will announce at the day of finals

Result Submission

A: Each team should submit a file named output.json that contains the answer (passages) for each description query on the Test Set. The test results should use JSON and follow the format as below:

{"query": " what are the benefits of fossil fuels ", "query_id": 0, "query_type": "description", "answers": ["The biggest advantage of using fossil fuels is that they can be easily stored and transported from one place to another. Large reserves of coal are therefore taken from the coal mines to the industries which are acres away from the mines. "] }

Field	Definition
query	Question query that same to the query at Test Set.
query_id	Should same to the query_id at Test Set
query_type	Optional
answers	Should only contains ONE best answer output by your algorithm

B: For both preliminary and final phases, each team should also submit a folder that contains CNTK code plus model that could reproduce the test results.

The folder structure should like:

team_xyx/ /* Root directory */

team_xyz/output.json /* a JSON file of test results */

team_xyz/script/* /* CNTK source code here */

team_xyz/model/* /* CNTK model here */

Note:

1. Each team should submit only ONE test results file (output.json) and corresponding source code and model.
2. The committee may run the submitted CNTK code to ensure the legitimacy and effectiveness of the submitted test results.
3. The committee will ignore any submission that doesn't adherer to the rules. Please do check your result file with the scoring script before submission.

Metric

Task uses ROUGE-L as metric for the ranking model quality and use the code below to score and output ROUGE-L score.

ROUGE-L is Longest Common Subsequence (LCS) based statistics according to the research paper (http://www.aclweb.org/anthology/W/W04/W04-1013.pdf). ROUGE-L firstly finds out the longest common subsequence (LCS) between a candidate passage (X) of length *m* and a reference passage (Y) of length *n*.

Define $\mathrm{R} = \frac{\mathrm{LCS(X,Y)}}{\mathrm{n}}$ and $\mathrm{P} = \frac{\mathrm{LCS(X,Y)}}{m}$

Then $\mathrm{ROUGE-L} = \frac{(1+b^2)\times \mathrm{R}\times \mathrm{P}}{\mathrm{R}+b^2\times \mathrm{P}}$, which $\mathrm{b} = 1.2$

Notice that ROUGE-L is 1 when two passages is identical (X = Y), and ROUGE-L is zero when LCS (X, Y) = 0.

Code to score and output metric

https://msmarco.blob.core.windows.net/msmarco/ms_marco_eval.tar.gz

Training Framework

You must use CNTK framework (https://github.com/Microsoft/CNTK) for the deep learning part of your solution. Using or depending on any other deep learning framework will forfeit your submission.

To install CNTK, please follow the instruction in https://docs.microsoft.com/en-us/cognitive-toolkit/setup-windows-python. You can simply `pip install <cntk wheel path>` CNTK python wheel that matches your corresponding setup.

CNTK documentation: https://cntk.ai/pythondocs/

Hand on notebook tutorials that cover basic to advance deep learning task: https://github.com/Microsoft/CNTK/tree/master/Tutorials

Relevant tutorials for text processing:

1. Simple example for language understanding: https://github.com/Microsoft/CNTK/blob/master/Tutorials/CNTK_202_Language_Understanding.ipynb
2. Sequence to sequence example that convert a sentence from one language to another: https://github.com/Microsoft/CNTK/blob/master/Tutorials/CNTK_204_Sequence_To_Sequence.ipynb
3. Deep structure semantic modeling with LSTM: https://github.com/Microsoft/CNTK/blob/master/Tutorials/CNTK_303_Deep_Structured_Semantic_Modeling_with_LSTM_Networks.ipynb

Baseline code

Here a CNTK baseline training code for MS Marco dataset: https://github.com/Microsoft/CNTK/tree/nikosk/bidaf/Examples/Text/BidirectionalAttentionFlow/msmarco

The above training code is in Python, it contains converter script that converts MSMARCO dataset to CTF CNTK format that is consumable by CNTK reader (you can write your own python code to read the data and feed it to the trainer, but it will be slower), and the end-to-end training script.

Hardware requirement

It is highly recommended to run the training code on a GPU platform.

For any other related questions, please contact techsupport@asc-events.org