Accelerating R Applications Using GPUs: Open Discussion

During this interactive Birds of a Feather discussion, a team of research computing professionals and facilitators will share experiences, challenges, and perspectives on support for graphics processing units (GPUs) within the R programming language. The organizers hope to spark discussion with the audience, providing an opportunity to share experiences and perspectives on the need for massively parallel package development in R and current efforts toward GPU acceleration.

EXTENDED ABSTRACT

R is a popular interpreted language that is well known for being free, open-source, flexible, and powerful. R offers its users a wide variety of simple and complex statistical and graphical techniques, robust visualization libraries, and several other tools that make it a valuable asset in the world of data science. In fact, as of April this year, the TIOBE Index [1] shows R to be in 11th place of the most popular programming languages in the world. However, there is one area in high performance computing that R continues to lag behind, and that is the integration of graphic processing units (GPUs).

GPUs have won the world over with their capability of parallelism, taking advantage of their numerous, more specialized cores to tackle problems of high compute density, machine learning, and any method that relies on basic linear algebra subroutines. It is this GPU acceleration that has allowed other languages, such as Python, to soar beyond R and lead the way for the future of data science. There are currently only four packages [2-5] that support GPUs within R while others, such as gpuR [6], have not been updated for many years and are no longer supported by the CRAN repository. Interfaces to Python and its well-integrated GPU support, such as the reticulate package, require understanding data structures and syntactic details for efficient execution.

Currently, ASU Research Computing has an ongoing engagement with NVIDIA and the concern of lack of GPU support within R has been discussed at length. In these discussions, it was suggested that users of R at ASU Research Computing move toward utilizing Python for its well-integrated GPU support, rather than trying to replicate these efforts within R itself. This is similar feedback that fellow institutions, such as University of Arizona, have also received when assisting with machine learning applications for their R user base. However, is this the best or ideal solution in supporting researchers who utilize R at high performance computing centers?

When engaging with these researchers, there is either resistance to having to learn a new language for their research or a compromise where the researchers will use Python for one section of their analysis before returning to R for the remainder of their downstream analyses. To better support these R users, ASU Research Computing [7] recently implemented an R language wrapper around NVIDIA's CUDA, cuBLAS, and cuSOLVER to extend GPU acceleration to

numerical linear algebra function calls. This was done in hope of improving user experience by accommodating these users within their language of choice while still allowing the chance for them to accelerate and optimize their research through the utilization of GPUs.

This BoF will highlight the need for GPU accelerated packages within R, preliminary efforts at Arizona State University to integrate GPU BLAS libraries, preliminary efforts at University of Arizona to support machine learning applications and research within R, and discussion on possible future directions. Representatives from NVIDIA, RMACC, and R Ladies have also been invited to contribute and participate as panelists.

INTERACTIVE DISCUSSION OUTLINE

1. Brief introductions of panelists
2. Background of the topic and experience of panelists
3. How could users of R benefit from more GPU support?
4. How could institutions and facilitators benefit from more GPU support within R?
5. Why isn't there more GPU support for R already?
6. "Just move to Python" - Is this the right answer to support researchers?
7. Future directions
8. Open discussion to the audience

ORGANIZERS (Name, Title, Organization, Specific Contributions)

1. Rebecca Belshe - Sr Software Application Analyst, Arizona State University, leads R training and facilitation for ASU Research Computing.
2. Gil Speyer - Director of the KE Accelerator, Arizona State University, optimizes and accelerates users' code in various domains and applications.
3. Jason Yalim - Assistant Research Professor, Arizona State University, leads Python training and facilitation for ASU Research Computing.
4. Chaitanya Inumella - Masters Student Employee, Arizona State University, has been developing software to use GPUs from R.
5. Natalie Mason - Student Employee, Arizona State University, assists users with ASU Research Computing resources and R package installations.
6. Chris Reidy - Research Facilitation Manager, University of Arizona, author of Intro to Machine Learning on HPC Using R workshop.

PANELISTS

1. Rebecca Belshe
2. Gil Speyer
3. Jason Yalim
4. Chris Reidy

This BOF has not been organized previously.

EQUIPMENT NEEDS

Standard microphone

REFERENCES

[1] TIOBE Software BV. 2022. TIOBE Index. Retrieved April 29, 2022 from https://www.tiobe.com/tiobe-index/

[2] CRAN. 2022. gcbd: 'GPU'/CPU Benchmarking in Debian-Based Systems. Retrieved April 29, 2022 from https://cran.r-project.org/package=gcbd

[3] CRAN. 2022. h2o4gpu: Interface to 'H2O4GPU'. Retrieved April 29, 2022 from https://cran.r-project.org/package=h2o4gpu

[4] CRAN. 2022. rkeops: Kernel Operations on GPU or CPU, with Autodiff, without Memory Overflows. Retrieved April 29, 2022 from https://cran.r-project.org/package=rkeops

[5] CRAN. 2022. torch: Tensors and Neural Networks with 'GPU' Acceleration. Retrieved April 29, 2022 from https://cran.r-project.org/package=torch

[6] CRAN. 2022. gpuR: GPU Functions for R Objects. Retrieved April 29, 2022 from https://cran.r-project.org/package=gpuR

[7] Chaitanya Inumella, Rebecca Belshe, Gil Speyer and Jason Yalim. 2022. Accelerating the R-language with cuBLAS and cuSOLVER. [Submitted].