Scalable Assessment of Progression-Free Survival in Metastatic Breast Cancer using HPC Clusters

Mike Mikailov, Subrata Mukherjee, Qian Cao, Fu-Jyh Luo, Kenny Cha

U.S. Food and Drug Administration, CDRH/OSEL/DIDSR, Silver Spring, MD, USA 20903

Abstract

In the assessment of progression-free survival (PFS) for metastatic breast cancer (mBC) drug approvals, the analysis of **extensive medical image datasets is a critical and complex task**. These analyses involve identifying, tracking metastatic lesions across longitudinal studies through advanced image registration processes, a task well-suited for High-Performance Computing (HPC) clusters. In collaboration with the U.S. Food and Drug Administration (FDA) and Novartis Pharmaceuticals, the FDA's Center for Devices and Radiological Health HPC team employs innovative scaling techniques to manage these challenging computational requirements. Traditional methods often relied on manually matching lesions for tracking, a process fraught with potential errors and inefficiencies. This presentation showcases a workflow with new strategies for efficient data partitioning and large-scale computational techniques on HPC clusters, developed for an automated lesion correspondence algorithm that accurately monitors both target lesions (actively monitored mBC lesions) and non-target lesions (abnormalities not currently the primary focus) in longitudinal studies. This workflow dramatically improved computational efficiency, cutting processing time from **7.6 months to less than two days**.

Introduction

This poster presents work from a collaboration between the U.S. Food and Drug Administration and Novartis Pharmaceuticals Corp, where we retrospectively utilize anonymized data from Phase III clinical trials, MONALEESA-2, 3 and 7, to investigate novel prognostic and predictive factors for metastatic breast cancer patients. Large CSV files packed with patient data (with multiple records per patient) and references to CT scan images taken at different time points present a significant processing challenges[1]. Processing a single CSV file takes **more than five days**.

Even massively parallel HPC clusters may struggle when faced with the demands of sequential processing. Conventional parallelization techniques, such as OpenMP and POSIX multithreading, are tailored for shared-memory systems limited to a single computing node, making them unsuitable for distributed computing environments like HPC clusters. The most dominant software parallelization technique, Message Passing Interface (MPI) used on HPC clusters is rigid: a) cannot exceed the maximum capacity of HPC clusters; b) MPI requires all necessary resources to be available simultaneously in order for the jobs to begin, which may lead to job "starvation" - jobs that require substantial resources might never be scheduled or could experience excessive delays in the queue, even if their priorities are high and continue to increase over time; c) no checkpointing by default - as the number of CPU cores requested by a large-scale application grows, the likelihood of encountering faulty computing nodes also rises, potentially leading to critical failures that cannot be recovered from; d) the newly available resources cannot be assigned to running MPI applications; e) performance bottleneck because of the tasks synchronization ; f) limited scalability.

We introduce a scalable workflow with an innovative efficient data partitioning and software scaling techniques to overcome the challenges.

Disclaimer

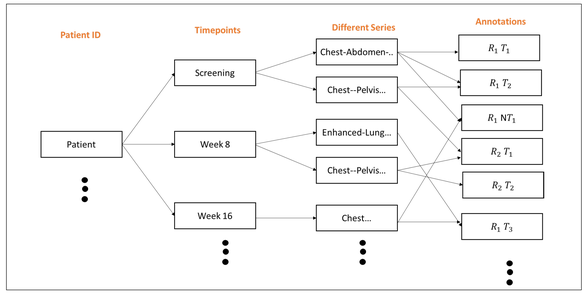
The mention of commercial products, their sources, or their use in connection with material reported herein is not to be construed as either an actual or implied endorsement of such products by the Department of Health and Human Services. This is a contribution of the U.S. Food and Drug Administration and is not subject to copyright.

The information in this presentation represents the opinions of the authors and does not necessarily represent FDA's position or policy.

[1] S. Mukherjee et al., "Image registration based automated lesion correspondence pipeline for longitudinal CT data," 2024 IEEE International Conference on Prognostics and Health Management (ICPHM), Spokane, WA, USA, 2024, pp. 184-192, doi: 10.1109/ICPHM61352.2024.10627649. https://ieeexplore.ieee.org/document/10627649.

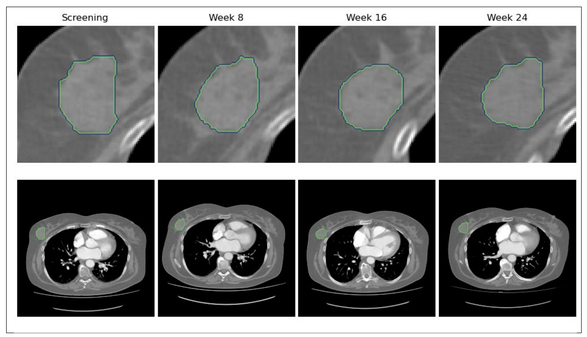
Materials and methods

Data we have



Complicated set of lesion annotations across images, patients and timepoints that requires data wrangling

Data we need



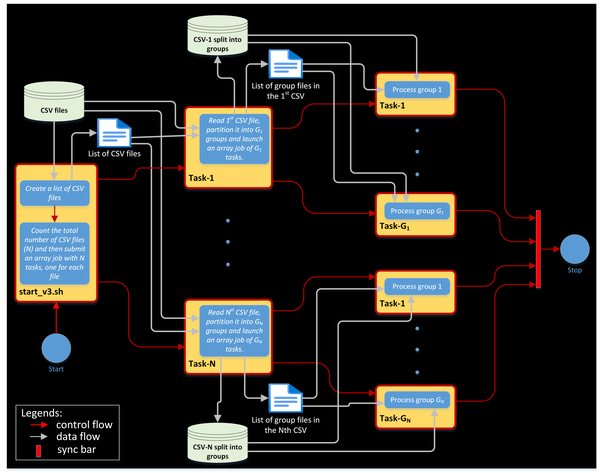
Need to track the same lesion over time

Data partitioning. Each record in the comma-separated CSV files begins with an anonymized Patient ID, followed by a time point. A succinct AWK code shown below efficiently divides a large CSV file into separate groups, with each group corresponding to a single patient at specific time points:

```
PREV=“”
awk –F, -v base=$BASE -v prev=$PREV -v title="$TITLE" ‘
    ($2 == “Screening” || $2 == “Week 8”) {
    if (prev != $1) { print title >> base”/”$1; prev = $1 }
    print >> base”/”$1
}’ “$CSV_PATH”
```

Hierarchical data-driven scaling. Hierarchical data-driven scaling is achieved through the built-in array job mechanism of job schedulers, which processes all CSV files and each patient group within those files in a fully parallel manner - without inter-task dependencies. The number of tasks in each array job is determined dynamically by the number of the CSV files and the number of groups. This approach initiates independent parallel tasks of array jobs across the distributed HPC clusters in a scalable fashion. The array job starts as soon as computational resources are available, even if it's just for a single task, thereby preventing job starvation. As additional resources become available, more tasks are automatically launched by the schedulers. Although the number of tasks in a single array job can exceed the HPC clusters' maximum capacity, this number is still limited by system implementations. The hierarchical scaling technique using the array job mechanism also addresses the system's constraint on the maximum number of tasks per array job. Additionally, array jobs offer inherent checkpointing capabilities, meaning that a system failure impacts only a subset of running tasks. Only the failed tasks need to be re-executed to recover from system failures.

Scalable data-driven workflow

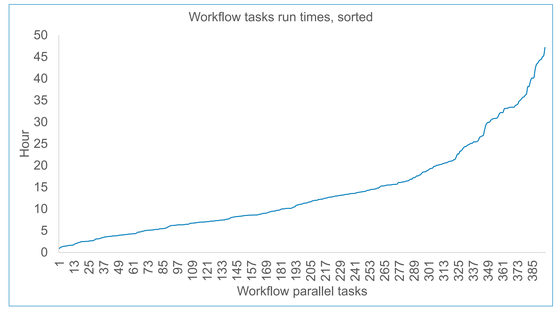


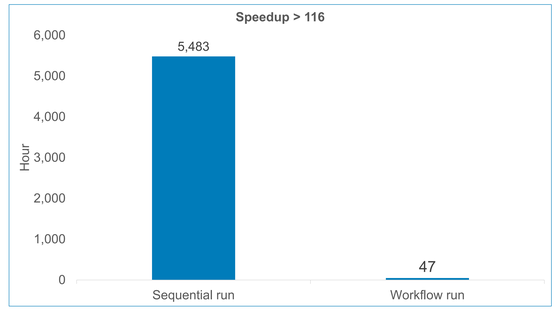
This scalable workflow streamlines processing for large datasets by combining data partitioning and hierarchical data-driven scaling. Here is how it works:

1. **User Input:** The user specifies a directory (containing N CSV files) in ***start_v3.sh*** Linux bash script.
2. **Script Execution:** The user starts ***start_v3.sh*** script which generates a list of all N CSV files in the directory.
3. **Parallel Processing (Level 1):** Then the script launches an array job with N independent tasks, each assigned a single CSV file.
4. **Task Processing:**
   - Each task using its unique task ID assigned by the job scheduler, retrieves the assigned CSV filename from the list.
   - Using provided AWK code, the task partitions the CSV file into G_i group files and creates a list of the group file names.
5. **Hierarchical Scaling (Level 2):** For each group:
   - A nested array job is launched with G_i sub-tasks, each assigned a single group.
   - Each sub-task using its unique task ID retrieves the assigned group filename from the list for parallel image processing.

Results and discussion

Using the workflow, a file from the MONALEESA-3 trials, which contains 5,296 records, and another file from the MONALEESA-7 trials, with 4,790 records, were divided into 212 and 183 groups, respectively. These groups were processed in parallel on the FDA CDRH Betsy cluster using a total of 395 parallel tasks. If executed sequentially, the entire process would have taken 5,483 hours, or 7.64 months. The longest individual task took 47 hours, or 1.97 days. The speedup is $O(G_1+\ldots+G_N)$, where N represents the number of CSV files and G_i denotes the number of partitions in the i-th CSV file. By applying data partitioning and hierarchical data-driven scaling techniques for the parallel processing of 1,580 CT images (collected at four time points for each patient) processing time was reduced from **7.6 months to under two days, achieving a speedup of 116.7x.**





Conclusion

This poster introduces a scalable workflow with an efficient data partitioning and novel hierarchical data-driven software scaling techniques on HPC clusters which was applied for parallel processing of CT images for 395 patients from the MONALEESA-3 and 7 clinical trials. The total processing time was reduced from 7.6 months to under two days, achieving a speedup of 116.7x.