AI4IO: A Suite of AI-based tools for capturing IO patterns

Michael Wyatt[1], Stephen Herbein[2], Kathleen Shoga[2], Todd Gamblin[2], Michela Taufer[1]

[1]University of Tennessee Knoxville [2]Lawrence Livermore National Laboratory

1. Problem overview and community needs

Resource managers like SLURM manage nodes and on-node resources, like processors and GPUs but fail to consider the wider set of resources in HPC systems, including the Parallel File System (PFS). The PFS can be a major bottleneck for HPC applications on large-scale systems: peak IO bandwidth is limited by power consumption constraints; concurrent executions compete for IO bandwidth (both on a machine and between different machines connected to the same PFS); and the rate of growth in PFSes to store the data is outpaced by the growth of HPC systems to generate data. Ultimately, HPC application jobs exhibit loss in performance and, in many cases, fail due to contention associated with partial or total loss in the PFSes' IO bandwidth.

Because there is an intrinsic complexity in predicting the occurrence of IO contention patterns (and failures) to access PFSes, current resource managers run short in accurate predictions. Consequently, resource managers often do not make efficient use of HPC resources when dealing with an heterogeneous ensemble of HPC jobs with a broad range of IO usage. When dealing with IO-intensive applications, in the best scenarios, the applications waste scarce node-hours by running their jobs slowly on congested resources; in the worst scenario, entire application allocations are wasted when their jobs fail because of the IO do not complete. Jobs that do very little IO may not be affected at all. If we can infer information about jobs, such as expected runtime and IO usage and which jobs are most affected by IO contention, we can inject the knowledge into the resource managers and ultimately mitigate the loss in performance for the impacted applications.

The community is in need of tools that provide HPC stakeholders with a priori knowledge for managing the wider set of HPC resources. These tools must be able to: (1) identify and exploit temporal and system-specific patterns in resource management; (2) augment resource managers to be resource-aware; and (3) avoid inconveniencing HPC users. Machine Learning (ML) based tools are a promising solution because: (1) ML can capture the trends unique to individual systems or utilities and leverage this for making predictions; (2) ML predictions can be pipelined into HPC resource managers to make schedulers resource-aware; and (3) ML can use the data that is already submitted to HPC machines as input and require no additional information from users.

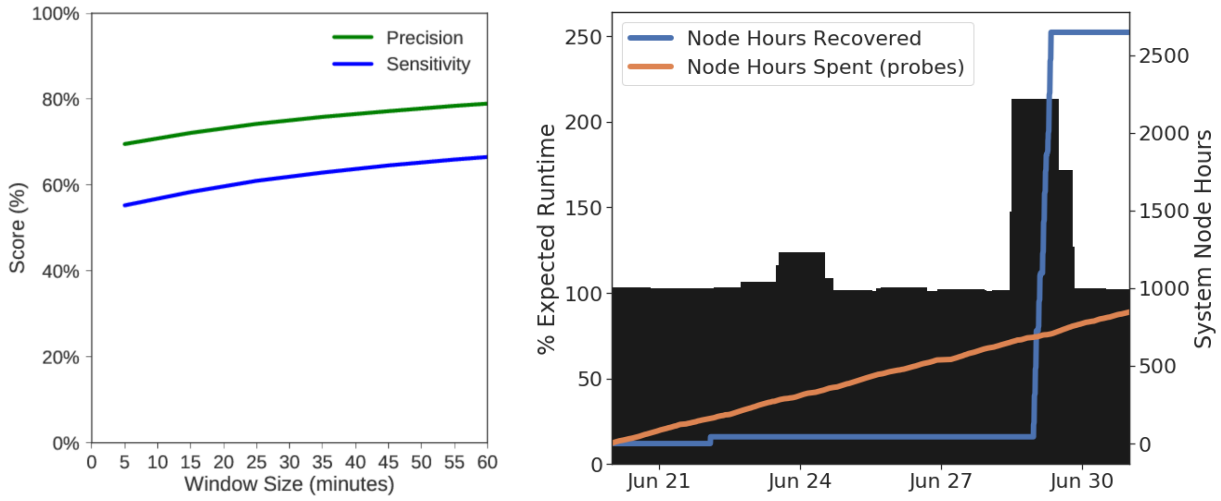
2. Our approach to tackle the problem

We are tackling the overall problem by designing AI4IO, a suite of ML-based tools that capture trends in historical IO data and make predictions on IO patterns that resource managers can use to improve the overall system resource utilization. Our suite currently consists of two tools: PRIONN and CanarIO.

These two tools approach the problem orthogonally to provide complimentary predictions: PRIONN works to **prevent IO contention** by providing resource managers with accurate predictions of individual job runtime and IO usage. We leverage PRIONN predictions to avoid scheduling IO-intensive jobs together, a common cause of IO resource contention and degradation. CanarIO works to **detect and mitigate IO contentions** by predicting the effects of IO degradation in real-time. We leverage CanarIO predictions to detect when IO degradation is occurring and which jobs are IO-sensitive and IO-resilient. We mitigate the contention effect by replacing IO-sensitive jobs with IO-resilient jobs in the execution queue during IO degradation, ultimately recovering wasted system resources.

We evaluate each tool using real HPC data and simulating the execution jobs with a resource manager. Figure 1a shows the precision and sensitivity for which we are able to predict future IO bursts based on PRIONN predictions. We are capable of predicting (and preventing) over 55% of IO bursts within 5 minutes of the actual event. Likewise, Figure 1b shows the node hours spent for data collection to train

CanarIO and the node-hours recovered during an IO degradation event on the system. In just a 10-day window, we observe over 1500 node-hours of saved system resources using predictions from CanarIO.



(a) (b)

Figure 1: Precision and sensitivity for which we are able to predict future IO bursts based on PRIONN predictions (a) and node hours spent for data collection to train CanarIO and the node-hours recovered during an IO degradation event (b).

3. Challenges in using ML for HPC

The accuracy of ML-based tools such as PRIONN and CanarIO heavily relies on the existence of pipelines collecting relevant data and making predictions actionable. In the development of our suite, the major challenge comes from the need for more reliable and accurate data collection infrastructure. Specifically, we observe that the resources being devoted to data collection on HPC machines is insufficient for the deployment of our tools on production machines. PRIONN and CanarIO rely on datasets collected from several sources, including user job scripts, SLURM logs, PFS logs, and system monitoring tools. When one or more of these datasets is unreliable (i.e., missing data or contains incorrect data) the ability to provide accurate predictions to the resource manager diminishes. For this reason, it is necessary to improve data collection infrastructure on HPC machines to support the tools' backbone ML algorithms. In addition to the data-collection infrastructure necessary to support our suite of tools, we observe that innovation of HPC resource managers is necessary. Current resource managers, such as SLURM, do not provide direct and native support for predictions from our tools. Next-generation schedulers should adopt a design model that allows for easy integration and augmentation of the job schedule with tools such as PRIONN and CanarIO.

4. Future directions

We envision future additions to our suite to include tools for monitoring and modeling additional on-node resources, such as GPUs, with constraints for system resources, such as power. Similar to our current tools, these tools will leverage the power of ML to provide predictions to resource managers that enable better scheduling for the wider set of HPC resources under many system- and node-level constraints. In order to develop these tools, new data sources will need to be identified and collected on HPC systems.