Workflow Validation: Detecting Silent Data Corruption with External Algorithmic Observers

Hadrien Croubois (ENS de Lyon), Franck Cappello (Argonne National Laboratory),
Tom Peterka (Argonne National Laboratory)

Validation of results is a pressing challenge in extreme-scale workflows. Data corruption, whether coming from bugs, attacks, or background radiation is increasingly likely in complex infrastructure and applications, and new solutions for corruption detection and tolerance are needed [1]. While addressing detection and recovery from fail-stop errors in single applications is fairly well studied, little is being done to detect and recover from systematic and nonsystematic silent data corruption (SDC), ie, from errors that do not cause obvious disruption, particularly in workflows of multiple applications. We describe below early work in error quantification that makes it possible to adapt the detection sensitivity to fit the expected accuracy of results. This model is used to detect SDCs in a pipeline of several application tasks operating on a single time step of data, without requiring a time series of multiple time steps. Such a model can be used to develop new adaptive and resilient workflow management systems (WMS).

In pipeline workflows such as the one shown in Figure 1(a), verification is commonly done using simple pipeline replication as in Figure 1(b). Comparing outputs of both replicas of the same pipeline can help detect corruptions affecting one of the replicas. The downside of this approach is the high cost of replication, and because all the replicas are identical, systematic errors affecting all replicas cannot be detected. We aim to design a new generic method that provides efficient error detection capabilities for both systematic and nonsystematic errors using an external algorithmic observer.

In order to detect systematic errors, the replication mechanism must involve different algorithms that do not share the same systematic corruption. Ideally, the replica (observer) should be less computationally expensive than the original algorithm. Further complicating the situation is the fact that most HPC applications produce approximated results. This fact implies that a simple binary difference cannot be used to compare those results, but rather that special metrics must be tailored to the expected distribution of the results. Furthermore, the distance between the two algorithms may be nonzero, even in uncorrupted cases, due to the intrinsic noise between algorithms.

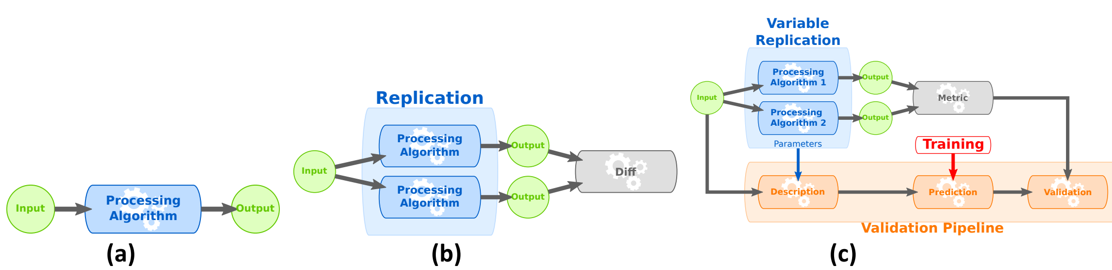


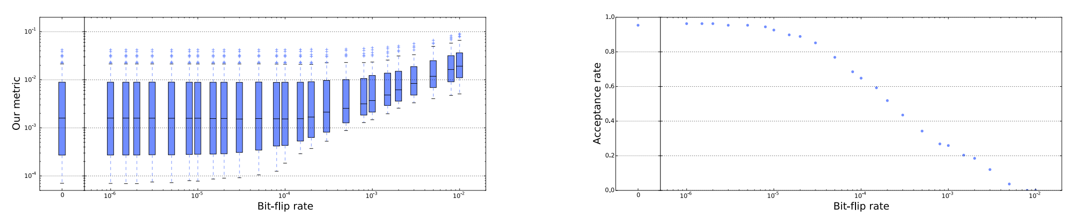
Figure 1. (a)Linear pipeline models: (a) without replication (default) (b) with simple replication, and (c) with an external observer (our method)

Detection is achieved by comparing the distance between results to the expected level of noise. This expected level of noise is used as a threshold between noise from natural variation between the methods and actual corruption. The detection is achieved by the validation pipeline, which is deployed alongside the main pipeline – Figure 1(c) – and aims at describing the context, predicting noise levels, and validating results originating from the main pipeline and detecting outliers.

The prediction step is done by building a model through machine learning and using this same model for data validation. The first step in the validation pipeline is extracting meaningful descriptors characterizing the input. Those descriptors contain both characteristics of the input data and parameters of the processing algorithms. Given those descriptors, the prediction then uses the learned model to compute the interval of expected values. The last step is to compare the result from the main pipeline with the predicted interval.

As part of the Decaf project [2], we applied this theoretical model to density estimation of cosmological dark matter. This use case is a linear pipeline producing density field images from sets of particles. Alongside a tessellation-based density estimator [3], we execute a simpler adaptive kernel density estimator [4], and compare the results using a custom metric. The complexity of the tessellation is $O(P^2)$ and the one of adaptive kernel is $O(P \log P)$. Figure 2 shows the variation of the differences between the two algorithms as well as the acceptance rate when applying the validation pipeline to our use case.

This generic method provides error detection capabilities on the order of the inherent approximation error of the underlying algorithms and the sensitivity of the difference metric to those errors. Our workflow demonstrates the theory of using an external observer in order to detect SDC. While our method cannot detect errors smaller than the approximation error of the underlying algorithms, ignoring small errors whose impact is below the expected accuracy is usually acceptable, if not desirable behavior. All in all, our early results show that adding data validation to WMSs can address the challenges of trust and validation in extreme-scale workflows; but also that much more research is needed.



(a) Distribution of differences between algorithms **(b) Acceptance rate of our workflow**

Figure 2. Sensitivity and accuracy of external observer with different amounts of memory corruption.

References:

[1] F. Cappello et al., "Improving the trust in results of numerical simulations and scientific data analytics", ANL tech report #2015, Argonne National Laboratory, 2015
[2] T. Peterka, D. Kimpe, F. Cappello, and J. Lofstead, "High-Performance Decoupling of Tightly Coupled Data Flows," 2014.
[3] T. Peterka, H. Croubois, N. Li, S. Rangel, and F. Cappello, "Self-adaptive density estimation of particle data," Submitted to SIAM Journal on Scientific Computing SISC Special Section on CSE15: Software and Big Data, pp. 1–20, 2015.
[4] E. Parzen, "On Estimation of a Probability Density Function and Mode," The Annals of Mathematical Statistics, vol. 33, pp. 1065–1076, 1962.