Scalable Workflow-Driven Hydrologic Analysis in HydroFrame

Shweta Purawat[1], Cathie Olschanowsky[2], Laura E. Condon[3], Reed Maxwell[4], and Ilkay Altintas[1]

[1] San Diego Supercomputer Center, UC San Diego
[2] Boise State University, Boise
[3] University of Arizona, Tucson
[4] Colorado School of Mines, Golden

Abstract. The HydroFrame project is a community platform designed to facilitate integrated hydrologic modeling across the US. As a part of HydroFrame, we seek to design innovative workflow solutions that create pathways to enable hydrologic analysis for three target user groups: the modeler, the analyzer, and the domain science educator. We present the initial progress on the HydroFrame community platform using an automated Kepler workflow. This workflow performs end-to-end hydrology simulations involving data ingestion, preprocessing, analysis, modeling, and visualization. We demonstrate how different modules of the workflow can be reused and repurposed for the three target user groups. The Kepler workflow ensures complete reproducibility through a built-in provenance framework that collects workflow specific parameters, software versions, and hardware system configuration. In addition, we aim to optimize the the utilization of large-scale computational resources to adjust to the needs of all three user groups. Towards this goal, we present a design that leverages provenance data and machine learning techniques to predict performance and forecast failures using an automatic performance collection component of the pipeline.

Keywords: Computational hydrology · Scientific Workflow · Reproducibility · Machine Learning.

1 Introduction

Hydrology aims to model freshwater on earth with the aim of facilitating large-scale and long-term impact on humanity. Resource contention and rapid growth of the human population demand a precise understanding of physics behind the flow of large water reserves and their interactions with interfacing geological systems. Computational Hydrology focuses on leveraging large-scale computational infrastructure to build hydrologic models. Hydrologic researchers are beginning to exploit the full potential of computational resources by converting hydrologic equation solvers into scalable simulations. However, to make a substantial improvement, scientists need a frictionless mechanism to perform continental-scale hydrologic experiments.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20

In the recent developments, platforms such as National Water Model(NWM) [1] and ParFlow-CONUS [2], [3],[4] have become critical toolsets on which hydrologic community depends. These platforms extend the reach of scientists in performing hydrologic simulations. However, to make a continental scale impact in the field of hydrology, there is a need for an efficient mechanism that matches the needs of hydrology with advancements in Computer Science. In recent years, the field of Computer Science has grown tremendously delivering capabilities such as cloud computing, parallelization mechanisms such as map-reduce, ultra-fast I/O hardware that facilitate efficient data movement, and Graphical Processing Units to perform large scale model development. Isolated developments in hydrology have not exploited these computational developments to the fullest, mainly due to the rapid rate of change in computer science techniques. Hydrologic researchers' main problem-solving abilities should focus on their domain expertise. To bridge this gap between hydrology and computer science, we are building a framework that provides abstractions to allow domain scientists to unleash the underlying capabilities of computer science, with the same ease as driving a car without knowing the internal workings of an internal combustion engine. We challenge ourselves to develop abstractions that make hydrologic research transparent to computational complexity, while still exploiting the benefits of computational developments.

The key challenges that hinder hydrologic research at the continental scale are experiment reproducibility especially due to the advent of distributed computing, code-portability, efficient data storage and movement, fault tolerance, and functionality to auto-scale hydrologic simulations. Our framework provides a graphical user interface oriented approach based on Kepler Scientific Workflow's modular code-encapsulation and execution structure. In addition, we leverage SmartFlows Suite to provide hydrology experts the benefit of using Machine Learning for automated failure prediction, resource allocation, dynamic fault tolerance, and 24x7 anytime anywhere control of simulations [5].

Kepler: The Kepler Scientific Workflow System provides domain scientists a simplified graphical front end, and a sophisticated workflow execution engine for backend computation [6],[7]. Hydrologic scientists can design large-scale simulations by connecting blocks of code on a canvas, and execute their experiments on local or remote hardware. Kepler provides two critical components for building new workflows: (a) Actor and (b) Director. An *Actor* encapsulates a piece of code and can be connected to another actor through a directed edge. The directed edge signifies the dataflow in the graph. A *Director* orchestrates the execution of the entire graph, based on parameters set by the user. A hydrologic researcher can formulate a complex experiment as a directed acyclic graph in Kepler, by building new actors or by re-using existing actors that encapsulate hydrologic code. Such an actor-oriented approach promotes code-reusability and rapid prototyping while the director performs heavy-lifting tasks of executing the instructions on distributed hybrid computational architecture.

SmartFlows: The SmartFlows Suite couples tightly with the Kepler system. It utilizes Machine Learning algorithms to predict failures, assign resources

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20

for execution, re-run workflows when faults occur, and provide an interface for anytime-anywhere control of workflow by scientists [5], [8],[9].

The SmartFlows Suite architecture is designed to provide hydrology scientists a range of services to choose from and create a customized simulation. Once implemented, the SmartFlows Suite will provide the functionality to monitor workflow execution continuously and dynamically make intelligent decisions, as per the initial parameterization of the experiment. Specifically, through the proposed SmartFlows framework, we seek to create a new paradigm where computer scientists and domain science experts work as a unified team. The unified Kepler-SmartFlows framework will act as an automated data assimilator and decision-maker, which will allow domain experts to focus their problem solving capabilities on hydrology specific challenges. Our Machine Learning driven framework will provide assurance to hydrology experts by intelligent handling of computational challenges such as parallelization and fault tolerance with minimal supervision.

In this paper, we present an architecture for an intuitive scientific toolbox that encapsulates computational complexity needed to execute large-scale hydrology codes on distributed hardware platforms. The framework modularizes hydrology algorithms. We have wired the modules of the framework to provide (a) native reproducibility support, (b) 24x7 monitoring and anytime-anywhere control feature, (c) real-time fault prediction by machine learning, (d) dynamic resource allocation based on hardware cost and system behavior, (e) ability to transparently deploy hydrology codes in containers, which facilitate code-portability, and (f) a shared platform for community engagement to upload hydrology workflows to cloud for solving the grand challenges of hydrology at scale.

We envision the proposed framework to become a critical partner that facilitates the hydrology community to unleash an era of continental-scale simulations: a vital step for solving future hydrology challenges.

The key contributions of this paper include:

- A unified architecture that proposes (i) modular hydrologic workflow development with multiple execution modes customized for hydrologic experimentation and (ii) a Machine Learning toolset that is pre-trained on hydrologic workflows' performance monitoring data.
- In-depth illustration of specific benefits the proposed framework brings to each target stakeholder group: the modeler, the analyzer, and the domain science educator.
- A scientific workflow tool that contains the widely-utilized ParFlow-CLM Model.
- An open-source containerization toolset that enables scientists to bootstrap the Kepler-ParFlow framework on their machine within minutes.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20

2 Related Work

Hydrologic researchers are actively involved in addressing global problems through innovation and continuous improvement. However, the challenges associated with water are not always bound by geographic partitioning. Most solutions that form isolated hydrologic models are approximations that deviate significantly from reality. Large-scale water bodies are essentially continental scale, and demand computational tools that operate at this scale. [10] have specifically highlighted continental scale simulations in the field of hydrology as a "grand challenge." Their work emphasizes the need to exploit massively parallel computational systems to solve hyperresolution hydrologic equations, which contain a large number of unknowns. [11] demonstrates the benefits gained by the use of parallel execution capabilities of ParFlow and the urge for the development of parallel system platforms in hydrology. [11,13,12,14,15,16] remark the requirement of adopting best practices in parallel computation from other fields, with the goal of improving infrastructure utilization when solving large-scale hydrologic problems.

There has been growing interest in the hydrology community in adapting the scientific workflow approach due to its modularity, reproducibility, code shareability, execution choices, and ease of use. [18] demonstrates the effect of automating the pre-processing step in hydrologic systems using Rule Oriented Data Management System (iRODS). However, this work in limited in scope to enhancing only a portion of the hydrology pipeline. [19,20,21] utilizes the Microsoft Trident Workflow System based solutions to demonstrate the benefits of using workflow design and execution paradigm in hydrology studies. [22] examines the challenges in hydrology related to big data, data flow, and model management, and build a preliminary adaptation of the Kepler Scientific Workflow System for hydrology applications. In contrast to these developments, the solution we present is first of its kind framework that will not only provide modularity, choice of execution, reproducibility but also dynamically lookout for predicting failures and course correct the execution trail to improve chances of successful workflow execution.

The framework presented in this paper utilizes the Kepler Scientific Workflow System [6] as a core pillar. Kepler has evolved over the last decade to serve the requirements of domain scientists from multiple research areas, due to its extensive repository of features, and a robust execution engine that can orchestrate complex workflows on distributed infrastructure. bioKepler [7], a workflow system for Biological Scientists, is a prominent example of Kepler's flexibility to adapt its features to a domain orthogonal to Computer Science. There are multiple works [24,25,29,26,27] that leverage Kepler's Workflow Engine to amplify and accelerate scientific discovery rate, and broaden the impact of research with improved shareability and provenance. Specifically, [26] deploy Kepler to perform single-cell data analysis using the Kepler System. [30,31] leverage the Kepler System to improve disaster management of Wild Fires by conducting real-time data-driven simulations that forecast a wildfire's trajectory, and visualizing wildfires to design mitigation response strategies. In addition, [28] provides an open-source platform, BBDTC, that allows domain science educators

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20

to build and share knowledge with a larger community. [32] develop fundamental techniques that allow precise prediction of workflow performance using Machine Learning techniques.

In contrast to isolated and partially focused development efforts, this paper presents an end-to-end solution that couples the Kepler System and the Smart Flows toolset to provide Hydrology researchers a framework critical for tackling the grand challenges of hydrology. This framework aims to bridge the gap between Hydrology Research and Computer Science. In coordination with Hydrology experts, we plan to build a solution that meets the needs of all three hydrology user categories: the modeler, the analyzer, and the educator. The framework will promote code-shareability, enhance data preparation and model development pipeline, provide multiple execution choices, automatically capture performance metrics, and leverage Machine Learning to dynamically forecast failures and adapt execution trails to reduce the impact of hardware faults.

3 HydroFrame Target User Groups

The design of the presented framework is driven by the quest to provide meaningful answers to every individual interested in conducting a hydrology study. To ensure large scale adoption of the framework, we've partitioned our potential consumers into three orthogonal categories shown in Fig. 1 (i) HydroFrame Target User groups. This organization enables us to write user-specific expectations and map the framework's capabilities to the demands of our users.

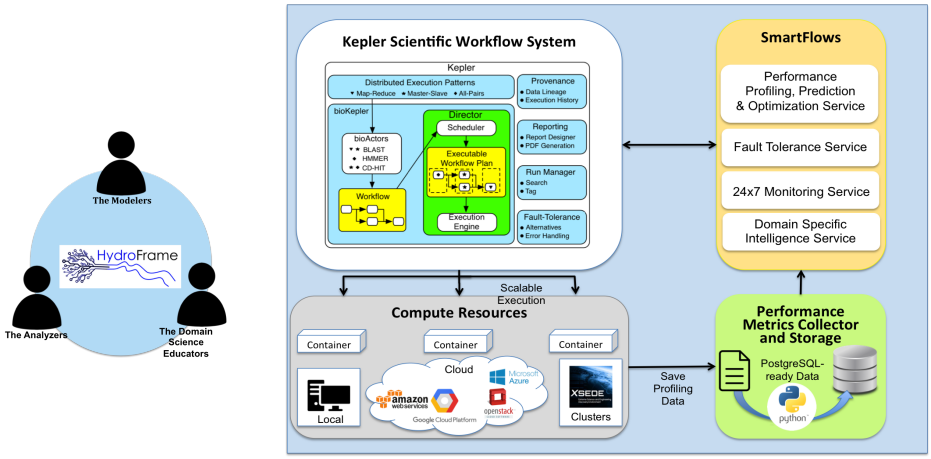

Fig. 1: (i) HydroFrame Target User groups (ii) End-To-End-Architecture of Scalable Workflow-Driven Hydrologic Analysis

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20

The first category represents 'the modeler.' A modeler is a researcher interested in the mechanics of a hydrology model and needs to control the inputs to a model for research. The technical bottleneck faced by the modeler is the investment of time required to gain an understanding of computational concepts that enable large scale distributed processing. Our proposed framework obviates such requirements by providing intuitive abstractions that act as powerful levers. Using our framework's graphical user interface, domain experts can leverage distributed data-parallel execution patterns to solve complex hydrologic equations. The modelers can design the flow of hydrologic data through an array of modular operations, choose from multiple execution patterns, and execute hydrologic instructions remotely on compute resources (e.g., a Kubernetes cluster), without necessarily writing lengthy programs with thousands of lines of code.

The second category represents the direct consumer of model outputs - 'the analyzer.' An analyzer is an applied researcher or scientist, who consumes a model's output, but not necessarily interested in changing the input parameters. The analyzer wants to visualize and derive a meaningful understanding of the complex hydrologic phenomenon, and impact the output of a project or create new downstream solutions based on the insights extracted. To benefit this community, our portal can facilitate model and output shareability and ensure reproducibility.

The last critical consumers are the educators who play a pivotal role in building the next generation hydrology researchers and scientists. These users are passionate about creating educational material that sparks community interest in hydrology research and enables students to understand the grand challenges. The educators want simplified visual outputs that effectively communicate the importance of solving the challenges of today's hydrologists. Our framework will enable these educators to export the model output, share model design, and input data in a streamlined manner with other educators and students. Our framework will provide K-12 educators the ability to give feedback and drive continuous improvement of the framework's usability.

4 A Reference Architecture for Measurement-driven Scalable Hydrologic Workflows

In the architecture illustrated in Fig. 1(ii) End-To-End-Architecture of Scalable Workflow-Driven Hydrologic Analysis, we build four tightly coupled systems, which will orchestrate a system that continues to operate efficiently, in the face of faults, and completes hydrology workflow through the cost-effective utilization of computational hardware. In addition, the architecture is designed with ease-of-use as a central theme, in order to facilitate community engagement among Earth System Science communities.

The core component of the architecture is the Kepler Scientific Workflow System, which leverages modularity and parallel execution strategies to transparently execute instructions from hydrology equation solvers. Continental-scale hydrology can generate massive amounts of data in a short duration, presenting

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20

opportunities for parallel execution. Earth Science Researchers can delegate parallelization and containerization to Kepler's execution engine. Kepler Workflow Engine provides native support to measure and log critical execution steps of hydrology workflows via the 'Provenance Module.' Kepler workflow continuously monitors and records performance statistics into a relational database. This data can be consumed by Machine learning algorithms for preemptive resource allocation decisions. Automated provenance collection by Kepler ensures real-time measurement of performance data and dynamic course correction.

The SmartFlows component plugs into the Kepler Scientific Workflow System and consumes critical data streams to perform real-time fault prediction. The performance prediction module of SmartFlows consumes hardware utilization data of hydrology workflows and provides real-time resource allocation strategies to meet performance goals in a cost-effective manner. In addition, the SmartFlows enhances the Earth Science Researchers' capability to monitor and change the course of a real-time workflow by providing anytime anywhere access to workflow execution on multiple devices. Through it 24x7 monitoring services, the SmartFlows component can inform the modeler of any critical change in expected behavior, or notify the analyzer when results become available. The domain-specific intelligence service can be used to record hydrology-specific execution patterns, which allow the SmartFlows module to make predictions with higher accuracy. For example, Earth Science Research Community can share execution histories of multiple hydrology workflows into the cloud and have a dedicated Machine Learning model that specializes in predicting hydrology specific workflows.

To enhance the ability of modelers and analyzers to repurpose existing workflow modules, and design complex workflows with sophisticated debugging capabilities, we aim to provide 'containerization' as a built-in option for each actor. Containerized actor execution can enhance fault tolerance by leveraging the Kubernetes framework [33]. Containerization promotes standardization, rapid code-portability, and scalability. Kubernetes containers will ensure replication of software stack, hence providing the same initial conditions for equation solvers. Once the software stack variation has been eliminated, any deviation in outcome can be traced to reasons such as algorithmic randomization or data distribution.

The framework design keeps automation at the core of each component. The seamless provenance collection feeds the machine learning algorithms and continuously makes them more accurate with each workflow execution. The containerization and modularization of code increase code portability and further facilitates performance metrics distribution among hydrologic experts, thus enabling performance prediction machine learning algorithms to learn at a much faster pace. This shared-training of domain-specific models can lead to the rapid maturity of the hydrology domain-specific intelligence service module in the SmartFlows toolbox.

The modular nature of code, and transparent execution of containers on the hardware of scientist's choice, will enable domain science educators to engage and build scientific community interest in hydrology at large scale. Hence, our

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20

architecture keeps not just enhances the computational scalability of hydrologic workflows, but also enables knowledge dissemination of domain to a wider audience in a seamless manner.

Keeping up with the rate of innovation in Computer Science: In the future, as the computer science community continues to innovate and build new hardware and software systems, the hydrologic domain experts will not have to go through the training cycles of new computational techniques. The Kepler Workflow Engine abstracts the computational complexity and provides a simplified interface for domain experts. In the future, the mechanism to orchestrate containers will continue to evolve; Kepler will leverage the advancements in this area while maintaining the simplicity of the interface to end-users. We envision the Kepler-SmartFlows architecture to become an auto-upgrading abstraction for hydrology research. The architecture will ensure that hydrology scientists continue to innovate by leveraging advances in parallelization, resource allocation, fault tolerance, reproducibility, provenance, and performance forecasting, without the need to catch up with ever-evolving computer science vocabulary.

5 Example: A Scalable Workflow Tool for ParFlow Hydrologic Model Simulations

As progression on the above-described architecture, in this section, we present a Kepler workflow tool that performs end-to-end hydrology simulations by integrating data ingestion, preprocessing, modeling, analysis, and performance profile collection steps in one automated tool. The workflow uses the Parflow CONUS model for hydrologic simulations. The workflow employs an in-built provenance feature of Kepler to record execution history. The workflow is built to support three execution choices – i) Local Execution, ii) Remote Cluster Execution, and iii) Kubernetes Execution. In this section, we will illustrate the workflow tool and different execution styles in detail; describe different features of the tool and how it can be repurposed to the requirements of the three target HydroFrame user groups – the Modelers, the Analyzers, and the Domain Science Educators.

5.1 End-to-end Hydrologic Simulation tool

The workflow Fig. 2 combines five major stages of ParFlow-CLM(Common Land Model) hydrologic simulation in one automated tool: data ingestion, data preprocessing, Parflow-CLM model simulation, data analysis, and performance profile collection steps. The stages are further broken down into modules. The modular design of the workflow supports repurpose and customization to create workflows for specific user groups. Each of the modules is autonomous and can be reused to create custom workflows with steps relevant to that user group's study.

Data Ingestion Stage: The Data Ingestion stage setups input files, dependent on execution style selected by a user, for subsequent processing and simulation steps. For local execution, it will specify the path of input directory containing

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20

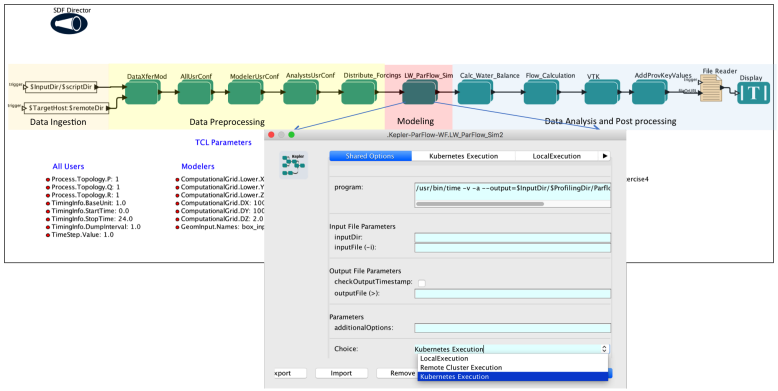

Fig. 2: A Scalable Workflow tool for ParFlow Hydrologic Model Simulations.

CLM (Community Land Model) input files, ParFlow input files, and TCL scripts. For Remote execution, the Kepler actor will copy files securely to the remote cluster.

Data Pre-processing Stage: The Data Pre-processing stage includes three different UserConf actors and a Distribute_Forcing actor. The UserConf actor facilitates users to configure frequently used hydrological model parameters through a graphical user interface (GUI), command-line, or web interfaces. The design ports model parameters as top-level workflow parameters. The UserConf modules programmatically update model parameters and other dependent files with user-specified values. The workflow contains three different actors 'ModelerUsrConf', 'AnalystsUsrConf,' and 'AllUsrConf,' focused on the needs of the three user groups – the Modelers, the Analyzers, and Common to all users, respectively. We ported the most frequently changed parameters from the TCL scripts to the top-level of the workflow, so users do not need to open TCL scripts every time to perform hydrologic simulation or analysis. The 'Distribute_Forcings' actor executes Dist_Forcings.tcl script that distributes the 3D meteorological forcings used by the ParFlow model.

Model Simulation Stage: This module simulates the Little Washita real domain(41km x 41km) using ParFlow (Parallel Flow) coupled with the Common Land Model (CLM) plus 3D meteorological forcing. Parflow is an open-source simulation platform that solves the Richards solver equation in 3D. This Kepler module runs the Tcl/TK script that provides data to the ParFlow model and runs the models.

Data Analysis Stage: This step performs an analysis of the simulation outputs obtained in the previous step. This stage performs water balance calculation, flow calculation, VTK output files generation. The 'Calc_Water_Blance' actor estimates changes in subsurface water storage, surface water storage, and total surface runoff over the simulation time. The 'Flow_Calculation' actor calculates the pressure and flow at the outlet over the simulation time. The 'VTK' actor runs VTK_example.tcl script that loads pfb or silo output and generates 3D

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20

vtk files. The vtk files are stored in the project subdirectory and are used for visualization using VisIt.

Containerization: We built a Kepler-ParFlow Docker image. It is a lightweight package that includes ParFlow, Kepler software, dependencies, and everything needed to execute the Kepler-ParFlow workflow tool. The workflow tool plus docker container provides a standardized, reproducible, and shareable unit to quickly set up the environment required to perform hydrologic simulations and analysis. The combination ensures quick and reliable code-portability from one computing resource to another. The Modeler and the Analyzer user communities can use the Kepler-ParFlow docker to shorten the environment set up time and concentrate on advancing hydrology science. The ease-of-use of the workflow and docker container will empower Domain Science Educators to effortlessly use in education and training of students and budding hydrologists.

5.2 Features of the Kepler-ParFlow Workflow tool

Multiple Execution Choices: Kepler "Build once – run many times" policy allows users to operate the same workflow tool in three different execution ways depending on the requirement, input data size, or available compute resources. We used the Kepler execution choice actor and built multiple sub-workflows to offer different alternatives to execute the same workflow, as shown in Fig 3. The Kepler workflow tool operates in three modes:

- Local Execution: This is the default mode of operation; it uses External execution Kepler actor to execute TCL/TK scripts on the local machine. To execute the workflow, users can install ParFlow and Kepler on their machine and follow the README guide to execute the workflow. Alternatively, users can download and use the Kepler-ParFlow Docker image from the Docker hub to speed up the development activity. The docker image includes ParFlow, Kepler software, and all dependencies needed to execute the workflow tool. This mode of execution is well suited for exploration, limited-scale hydrologic simulations, education, and training.
- Remote Cluster Execution: This mode of operation enables execution of ParFlow tools on a remote cluster, such as BlueM at Colorado School of Mines, an HPC resource. The subworkflow copies input files to a remote cluster using Kepler SCP(secure copy) actor; creates, submits and manages job on a remote cluster using Kepler SSH Session and GenericJobLauncher actors; and copies back the results to the local machine.
- Kubernetes Execution: This execution choice facilitates to run and manage different steps in the workflow tool in docker containers on a remote cluster managed by Kubernetes. Fig. 3 illustrates Containerized scalable execution with Kubernetes Execution Choice. The Kepler abstraction layer provides an easy-to-use interface on top of Kubernetes hiding all the underlying complexities. Specifically, Kepler actors create all the YAML files and implement all the steps required by Kubernetes to create persistent volume, transfer data from your machine the remote cluster, create pods, submit and manage the

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20

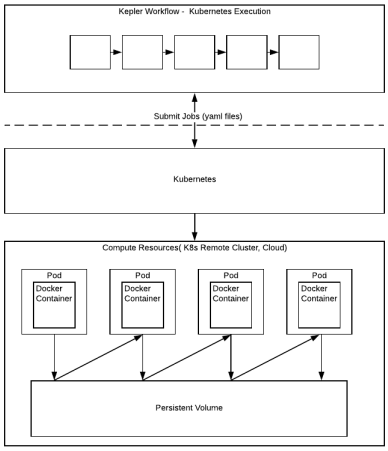

Fig. 3: Containerized Scalable Execution with Kubernetes Execution Choice.

job on the remote cluster and scalability. Each actor in the Kepler-ParFlow tool can be executed within a docker container on a pod. The docker containers are created using the Kepler-ParFlow Docker image. Kepler-ParFlow docker image packages all software and dependencies needed to run the application in a standard unit.

Provenance, Reproducibility, and Reporting: The workflow employs an in-built provenance feature of Kepler to record execution history. We advanced the provenance to collect and save the System and Software information for the execution for complete reproducibility. The system and software version information are gathered and recorded in the provenance database using the 'AddProvValues' actor. The builtin reporting suite in Kepler assists users to generate reports from workflow runs. Users can execute Kepler workflow within Jupyter Notebook using Kepler Magic functions https://github.com/words-sdsc/Jupyter_Kepler_Integration and create reports in Jupyter Notebook.

Automated Performance Metrics Collection: We integrated the Linux time application profiler in the hydrologic workflow that enables systematic collection of application performance diagnostic features: execution time, memory, and io measurements. The collected data will be utilized by machine learning algorithms in the SmartFlow services to predict Performance, in Fault Tolerance, and to optimize utilization of large-scale computational resources dynamically, as the experiment progresses.

6 Conclusion

In this paper, we present the initial developments of our long-term vision to produce an integrated framework that enables the Hydrology research community

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20

to leverage advancements in Computer Science, such as distributed computing, fault-tolerance, reproducibility. We developed a Kepler Scientific Workflow that allows scientists to execute the Parflow-CLM model on large-scale hybrid infrastructure. The workflow consists of five modules that involve data preparation, ParFlow model simulation, analysis, and visualization. To facilitate performance measurements, we have added functionality to measure and collect performance metrics such as CPU usage, memory usage, I/O rates within the Parflow-CLM Workflow. This will enable scientists to record both hardware and software configurations automatically, enhancing the reproducibility of the workflow. In addition to providing the core functionality, the workflow will enable the Hydrology community users such as modelers, data analyzers, and educators to configure modeling parameters through both command line and graphical user interface, without the need to edit code. Through continuous development, we aim for the presented framework to become a critical workbench for Hydrologists for building modeling simulations and sharing knowledge.

In the future, we plan to add Machine Learning modules that leverage the collected configurations and performance data to dynamically predict hardware usage. We plan to add an auto-deploy feature, which will initiate a new instance of the workflow, once the probability of failure reaches a threshold, preset by the modeler. These functionalities will be encapsulated in the SmartFlows Toolset, which will allow modelers, analyzers, and domain science educators to control and monitor workflow progress from any connected device, through SmartFlow's 24x7 monitoring module.

Software Availability for Community Outreach and Education:

- https://cloud.docker.com/repository/docker/spurawat/kepler-parflow
- https://github.com/hydroframe/Workflows.git

Acknowledgements: This work is supported by NSF OAC CSSI 1835855, and DOE DE-SC0012630 for IPPD.

References

1. Gochis, D. J., Yu, W., D.N. Yates (2013). The WRF-Hydro model technical description and user's guide, version 1.0. NCAR Technical Document. Boulder, CO, National Center for Atmospheric Research: 120.
2. Ashby, S. F. and Falgout, R. D. (1996). "A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations." Nuclear Science and Engineering 124(1): 145-159.
3. Jones, J. E., and C. S. Woodward (2001), Newton-Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems, Adv. Water Resour., 24(7), 763–774, doi:10.1016/S0309-1708(00)00075-0.
4. Kollet, S. J., Maxwell, R. M. (2006). Integrated surface-groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model. Advances in Water Resources, 29(7), 945–958 doi.org/10.1016/j.advwatres.2005.08.006.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20

5. Altintas, I., Purawat, S.,Crawl, D., Singh, A. and Marcus, K., "Toward a Methodology and Framework for Workflow-Driven Team Science" in Computing in Science Engineering, vol. 21, no. 04, pp. 37-48, 2019. doi: 10.1109/MCSE.2019.2919688
6. Ludaescher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M. M., Lee, E.A., Tao, J. and Zhao, Y., "Scientific Workflow Management and the Kepler System," Concurrency Computation Practice: and Experience, Vol.18, pp. 1039–1065, 2006.
7. Altintas, I., Wang, J., Crawl D. and Li, W. (2012). Challenges and approaches for distributed workflow-driven analysis of large-scale biological data: vision paper. Proceedings of the 2012. Joint EDBT/ICDT Workshops, ACM.
8. A. Singh, A. Rao, S. Purawat, and I. Altintas, A Machine Learning Approach for Modular Workflow Performance Prediction, in Proceedings of the 12th Workshop on Workflows in Support of Large-Scale Science, New York, NY, USA, 2017, p. 7:1–7:11. https://doi.org/10.1145/3150994.3150998
9. A. Singh, M. Schram, N. Tallent , and I. Altintas, Deep Learning for Enhancing Fault Tolerant Capabilities of Scientific Workflows in IEEE International Workshop on Benchmarking, Performance Tuning and Optimization for Big Data Applications, at the IEEE Big Data 2018 Conference, Seattle, WA
10. Wood, E. F., et al. (2011), Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth's terrestrial water, Water Resour. Res., 47, W05301, doi:10.1029/2010WR010090.
11. Kollet, S. J., Maxwell, R. M., Woodward, C. S., Smith, S., Vanderborght, J., Vereecken, H., and Simmer, C. (2010), Proof of concept of regional scale hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources, Water Resour. Res., 46, W04201, doi:10.1029/2009WR008730.
12. Bierkens, M. F. P. (2015), Global hydrology 2015: State, trends, and directions, Water Resour. Res., 51, 4923– 4947, doi:10.1002/2015WR017173.
13. Clark, M. P., et al. (2015), A unified approach for process-based hydrologic modeling: 1. Modeling concept, Water Resour. Res., 51, 2498– 2514, doi:10.1002/2015WR017198.
14. Maxwell, R. M. (2013). "A terrain-following grid transform and preconditioner for parallel, largescale, integrated hydrologic modeling." Advances in Water Resources 53: 109-117.
15. Maxwell, R. M., Condon, L. E. and Kollet, S. J. (2015). "A high-resolution simulation of groundwater and surface water over most of the continental US with the integrated hydrologic model ParFlow v3." Geoscientific Model Development 8: 1-15.
16. Maxwell, R. M. and Condon, L. E. (2016). "Connections between groundwater flow and transpiration partitioning." Science 353(6297): 377.
17. Hutton, C., Wagener, T., Freer, J., Han, D., Duffy, C., and Arheimer, B. (2016), Most computational hydrology is not reproducible, so is it really science?, Water Resour. Res., 52, 7548– 7555, doi:10.1002/2016WR019285.
18. Mirza M. Billah, Jonathan L. Goodall, Ujjwal Narayan, Bakinam T. Essawy, Venkat Lakshmi, Arcot Rajasekar, Reagan W. Moore, Using a data grid to automate data preparation pipelines required for regional-scale hydrologic modeling, Environmental Modelling Software, Volume 78, 2016, Pages 31-39, ISSN 1364-8152, https://doi.org/10.1016/j.envsoft.2015.12.010.
19. Fitch, Peter; Perraud, Jean-Michel; Cuddy, Susan; Seaton, Shane; Bai, Qifeng; Hehir, David. The Hydrologists Workbench: more than a scientific workflow tool. In: Sims J, Merrin L, Ackland R and Herron N, editor/s. Water Information Research and Development Alliance: Science Symposium Proceed-

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20

ings; 1-5 August 2011; Melbourne, Australia. Melbourne: CSIRO; 2012. 61-69. http://hdl.handle.net/102.100.100/100717?index=1
20. Cuddy, S. M. and Fitch, P., "Hydrologists Workbench – a hydrological domain workflow toolkit" (2010). International Congress on Environmental Modelling and Software. 246.
21. Piasecki, Michael Lu, Bo. (2010). Using the Workflow Engine TRIDENT as a Hydrologic Modeling Platform. 12. 3680.
22. Guru, S. M.; Kearney, M.; Fitch, P.; Peters, C. Challenges in using scientific workflow tools in the hydrology domain. In: 18th IMACS World Congress; MODSIM 2009 International Congress on Modelling and Simulation; 2009; Cairns, Qld. 2009. 3514-3520. http://hdl.handle.net/102.100.100/111463?index=1
23. Perraud, J., Fitch, P. G., Bai, Q. 2010. Challenges and Solutions in Implementing Hydrological Models within Scientific Workflow Software. AGU Fall Meeting Abstracts 2010, H53H-06.
24. Chen, R., X. Wan, I. Altintas, J. Wang, D. Crawl, S. Phan, A. Lawrence and M. Ellisman (2014). "EPiK - A Workflow for Electron Tomography in Kepler." Procedia computer science 29: 2295-2305.
25. Gan, Z., J. Wang, N. Salomonis, J. C. Stowe, G. G. Haddad, A. D. McCulloch, I. Altintas and A.C. Zambon (2014). "MAAMD: a workflow to standardize meta-analyses and comparison of affymetrix microarray data." BMC bioinformatics 15(1): 69.
26. Qian, Y., H. Kim, S. Purawat, J. Wang, R. Stanton, A. Lee, W. Xu, I. Altintas, R. Sinkovits and R. H. Scheuermann (2015). FlowGate: towards extensible and scalable web-based flow cytometry data analysis. Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure, ACM.
27. S. Purawat, P. Ieong, R. Malmstrom, G. Chan, R. Walker, I. Altintas, and R. Amaro, A Kepler Workflow Tool for Reproducible Molecular Dynamics", Biophysical Journal, Jun 20;112(12):2469-2474, 2017. doi: 10.1016/j.bpj.2017.04.055
28. Purawat, S., C. Cowart, R. E. Amaro and I. Altintas (2017). "Biomedical Big Data Training Collaborative (BBDTC): An effort to bridge the talent gap in biomedical science and research."Journal of Computational Science.
29. Wang, J., Y. Tang, M. Nguyen and I. Altintas (2014). A Scalable Data Science Workflow Approach for Big Data Bayesian Network Learning. Proceedings of the 2014 IEEE/ACM International Symposium on Big Data Computing, IEEE Computer Society.
30. Altintas, I., J. Block, R. de Callafon, D. Crawl, C. Cowart, A. Gupta, M. Nguyen, H.-W. Braun, J. P. Schulze, M. Gollner, A. Trouve and L. Smarr (2015). Towards an Integrated Cyberinfrastructure for Scalable Data-driven Monitoring, Dynamic Prediction and Resilience of Wildfires. Proc. of the Int. Conf. on Computational Science, ICCS 2015.
31. Nguyen, M., D. Uys, D. Crawl, C. Cowart and I. Altintas (2016). A Scalable Approach for Location-Specific Detection of Santa Ana Conditions. Proceedings of the 2016 IEEE International Conference on Big Data.
32. A. Singh, M. Nguyen, S. Purawat, D. Crawl, I. Altintas, Modular Resource Centric Learning for Workflow Performance Prediction, in the 6th Workshop on Big Data Analytics: Challenges, and Opportunities (BDAC) at the 27th IEEE/ACM International Conference for High Performance Computing, Networking, Storage, and Analysis (SC15) http://arxiv.org/abs/1711.05429
33. B. Burns, B. Grant, D. Oppenheimer, E. Brewer, J. Wilkes. 2016. Borg, Omega, and Kubernetes. Queue 14(1), 2016, doi: https://doi.org/10.1145/2898442.2898444

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50371-0_20