Data Management and Simulation Support Accelerating Carbon Capture through Computing

You-Wei Cheah, Joshua Boverhof, Abdelrahman Elbashandy, Deb Agarwal
Lawrence Berkeley National Laboratory
Berkeley, CA

Jim Leek, Thomas Epperly
Lawrence Livermore National Laboratory
Livermore, CA

John Eslick, David Miller
National Energy Technology Lab
Pittsburgh, PA

***Abstract*—The Carbon Capture Simulation Initiative (CCSI) project has developed and deployed scientific infrastructure called the CCSI Toolset. The CCSI Toolset provides state-of-the-art computational modeling and simulation tools to accelerate the commercialization of carbon capture technologies from discovery to development, demonstration, and ultimately the widespread deployment to hundreds of power plants. Carbon capture technologies have the potential to dramatically reduce the carbon emissions from power plants. The CCSI Toolset provides end users in industry with a comprehensive, integrated suite of leading-edge, scientifically validated models with simulation, uncertainty quantification, optimization, risk analysis and decision making support. The CCSI Toolset has at its core an integrated framework that enables execution of simulations and workflows including optimization and uncertainty parameter sweeps using a wide variety of computing platforms including desktops, clusters, Clouds, and HPC systems. The integration framework enables the running of a variety of commercial process simulation packages as well as custom simulators. Moreover, the framework enables scientists to run and manage thousands of concurrent simulations to perform optimizations and uncertainty quantification. Components of the CCSI Toolset are connected through the use of a data management system that stores data to a repository and enables the tracking of provenance for each simulation as well as its associated components. The data management system tracks all the configurations, models, simulations, and results created during the design of a carbon capture system and supports the design life-cycle as well as decision making. The primary contribution of this paper is thus the design and implementation of the integration framework within the CCSI Toolset, which provides both data management and simulation support for CCSI. This integration framework has been deployed and is in use by several groups of researchers and commercial entities.**

***Keywords*—carbon capture, cyberinfrastructure, e-Science, data management, parallel simulation**

I. Introduction

Carbon capture has the potential to significantly reduce greenhouse gas emissions from power plants. Development of cost effective carbon capture processes that can operate at the plant scale (typically around 650MW) is required for the widespread deployment of these technologies. Historically, the energy sector takes up to 15 years [1] to move from the laboratory to pre-deployment and on the order of 20 to 30 years for industrial scale deployment [2]. The U.S. Department of Energy initiated the Carbon Capture Simulation Initiative (CCSI) in 2011 with the goal of developing a computational toolset that would enable industry to more effectively identify, design, scale up, operate, and optimize promising concepts (Miller et al., 2014). The CCSI project is a partnership among national laboratories, industry and academic institutions that is developing and deploying the computational modeling and simulation tools. The goal of the project is to deploy state-of-the art scientific computational capabilities to accelerate carbon capture technologies from discovery to development, demonstration, and ultimately the widespread deployment to hundreds of power plants.

The CCSI Toolset supports particle scale (computational fluid dynamics, CFD), device scale (process simulators), and plant scale (superstructure simulators) modeling and simulation. It enables multi-scale modeling and the features needed to evaluate, optimize and scale the models and to make decisions based on the results. By developing the CCSI Toolset, a comprehensive, integrated suite of validated science-based computational models, CCSI provides simulation tools that will increase confidence in designs, thereby reducing the risk associated with incorporating multiple innovative technologies into new carbon capture solutions. The scientific underpinnings encoded into the suite of models will also ensure that learning will be maximized from successive technology generations.

The requirements of the CCSI Toolset were developed through extensive interaction with the CCSI Industry Advisory Board (IAB) and active testing by industry partners during the development phase. User Experience design techniques were utilized during the project to solicit input on priorities and processes from the industry participants and to improve usability of CCSI Toolset modules [3], [4]. One of the results of our early interaction with the industry participants was the recognition that they already have extensive experience and libraries of models in existing process simulation packages including AspenPlus, Aspen Custom Modeller (ACM), Simulink, gProms, and custom simulation packages including some that are Excel-based. In addition, the users in industry are primarily using Windows systems for process simulations. Although users expressed a willingness to consider changing simulation environments, it was clear from the early interviews

that any change would require building up process model libraries and validation of these models.

Realizing the potential of CCSI depends on having an *integration framework* to provide the simulation and data management support for the multi-scale simulations, uncertainty quantification, decision support, and optimization of scientific models. The integration framework enables easy interoperability of the wide variety of process simulation packages in use in the industry with other components of the toolset and the ability to run 1000s of simulations and 100s of iterations. Simulations can be run on desktops, Cloud environments, clusters, or high-performance computing environments. The choice of simulation execution resources is based solely on the computational resources available to the user and the user's preferred runtime environment. Tracking all of the experiments, models, simulations, and results along with conclusions requires data management support to enable decisions. The primary contribution of this paper is the presentation of the design and implementation of the CCSI integration framework.

The remainder of the paper is organized as follows. We first discuss related work in Section II and present the CCSI Toolset architecture for computational tools in Section III. This is followed by an in-depth discussion of the implementation of these computational tools in Section IV. The implications of the tools and adoption are discussed in Section V. Lastly, we present our conclusions and future work in Section VI.

II. Related Work

There are several scientific workflow frameworks with integrated data management and simulation capabilities. However, there is relatively little past work to implement either capability for process simulation environments. In this section, we will focus our discussion on related work for the CCSI Toolset's data management and parallel simulation support.

A. Data Management

Data management plays a key role in science cyberinfrastructure [5]. In the commercial industrial simulation space, ANSYS Engineering Knowledge Manager (EKM) provides an integrated solution providing simulation-based process data management capabilities based entirely on ANSYS tools [6]. It does not allow integration of simulators from other companies. The CCSI project must be able to also support AspenPlus, ACM, and SimuLink simulations as well as custom simulations. Various data management systems have been developed for scientific projects such as the ASCEM Data Management component [7] which relies on a combination of Velo [8], a package based on semantic wiki and Alfresco CMS capabilities, and traditional database technologies [9] for data management and Vistrails an open-source scientific workflow and provenance management system [10]. Earth System Grid is a peer-to-peer distributed data system supporting climate science simulations [11]. The high-energy physics experiments at CERN's Large Hadron Collider have also developed data management solutions. For example, the Atlas experiment uses Don Quixote 2 [12] and Panda [13]. Although a few of these infrastructures have found acceptance beyond their specific domains, it is rare since each contains many customizations to make it useful to the original domain.

Prior research has investigated the properties needed in a science metadata system, eScience data management systems should possess a number of requirements. It is crucial that metadata is maintained for data objects [14] and their provenance needs to be tracked [15].

Versioning has been demonstrated to be useful in scientific workflows [10], [16]. In addition, data management systems should facilitate the sharing and collaboration amongst scientists. Hence, capabilities such as searching and version control are desirable for data management systems. A nice overview of much of the current work in data provenance and annotation can be found in [17]. In the CCSI Toolset, the Data Management Framework (DMF) fulfills the role for data management by incorporating the requirements for metadata, data provenance, versioning and also the facilitation of sharing and collaboration.

B. Parallel Simulations

Over the years, much research has been done on running simulations in parallel. With emergent trends such as the Cloud, studies have evaluated the effectiveness of deploying simulation runs on Cloud environments [18], [19]. When the project first started, the main distributed technologies involved MapReduce and Hadoop at that time. These technologies have proven to be useful for scaling up simulations [20]. Container managers such as Docker [21] and rkt [22] are relatively new to the scene but have revolutionized the development and deployment of large-scale simulations [23].

As technology has evolved, so has science. Many scientific domains are shifting towards running simulations in parallel. In material science, we have systems such as Fireworks [24], which is designed for high throughput applications. Similarly, GROMACS [25] is a system that does high-throughput molecular simulations. Qdo [26] is another system that supports batch execution of tasks. More recently, Apache Spark [27] and Amazon's AWS Flow Framework [28] provide support related to the CCSI Toolset's Turbine Science Gateway capabilities. However, they were not available when the project begun. The CCSI Toolset is set to change the scientific landscape of carbon capture by allowing scientists to run simulations in parallel on distributed environments or Cloud resources.

III. CCSI Toolset Architecture

The CCSI Toolset's concentration is on providing the capabilities needed to enable effective design and scale-up of carbon capture processes for installation in a power plant. A typical design process starts with bench-scale experiments in order to find the most promising capture process (chemical reaction), or in some cases CFD simulations for understanding how chemical reactions would operate in a device. This is normally followed by building a small device to test basic concepts, which is then followed by using process simulation

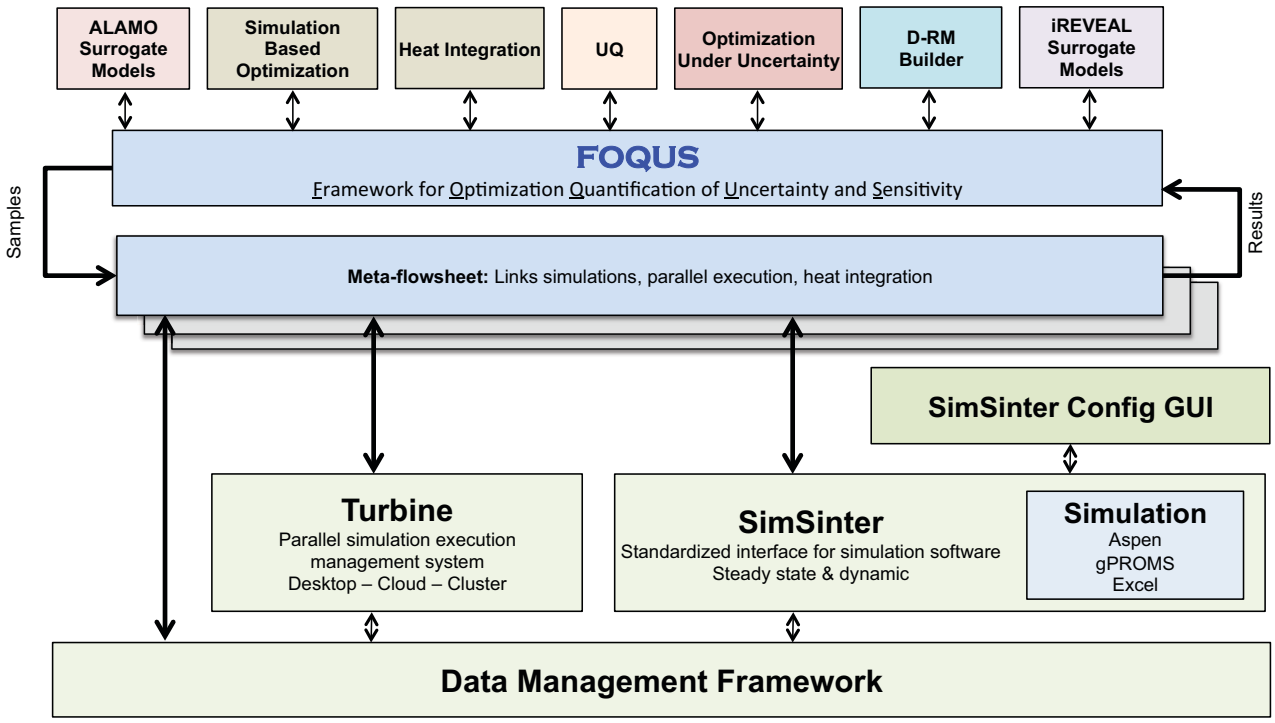

Fig. 1. Architecture of CCSI Toolset

to build successively larger devices to predict expected performance during the scale-up process. The design process like this typically involves many iterations at each scale.

The intent of the CCSI Toolset is to utilize the latest scientific computational tools and understanding of the design process to enable thorough study of the design before a physical device is built. This approach is expected to reduce the number of physical prototypes and increase the likelihood that devices built at each scale succeed the first time. As a result, this will maximize the learning at each level. In order to accomplish this goal, the existing process simulation and CFD capabilities need to be augmented with an integrated ability to perform uncertainty quantification and optimization. In addition, the process needs to be studied at all scales in the context of the design process and the plant.

To address these goals and to provide the required functionality, the CCSI Toolset is designed to be modular with a unified interface and support framework. The design leverage existing software components such as process simulators and CFD capabilities and allows the team to concentrate on building the new functionality required. Figure 1 shows a schematic of the components built for the toolset. The top layer identifies modules providing the core capabilities such as process scale modeling using surrogate models (ALAMO), optimization of device configurations, uncertainty estimation to understand sensitivities of models (using PSUADE [29]), dynamic reduced model development to enable understanding of behavior during changing conditions (D-RM Builder), and surrogate model development to enable CFD models to be embedded in process simulations (iREVEAL). The next layer down is the Framework for Optimization Quantification of Uncertainty and Sensitivity (FOQUS), which is the brains of the Toolkit. The FOQUS layer orchestrates the execution of the core capability modules on behalf of the user. FOQUS relies on the integration framework components (the heart of the toolkit) to run simulations and store results. The process simulation framework consists of three components: 1) Turbine Science Gateway for simulation management, 2) SimSinter to connect process simulators to the framework through standard interfaces, and 3) the Data Management Framework to store and track all of the activities and data in the system.

The details of the core capability modules are beyond the scope of this paper, but an overview will be provided here to provide context and to motivate the process simulation framework capabilities. There are five major modules, they are: 1) Simulation-based optimization, 2) Uncertainty Quantification (UQ), 3) Optimization Under Uncertainty, 4) Heat Integration, and 5) D-RM Builder. Simulation-based optimization is a technique where models are evaluated by running 100's to 1000's of iterations of the process simulation software and using genetic algorithms to evaluate an objective function and also determine the optimal configuration of a device. The UQ module encompasses a rich selection of mathematical, statistical, and diagnostic tools for application users to perform UQ studies on simulation models. These studies are carried out by treating the simulation as a black box and varying the inputs while running 1000's of simulations and studying

their impact on the outputs. Optimization Under Uncertainty extends the simulation-based optimization by including studies of the contribution of model parameter uncertainties in the objective function. The Heat Integration tool maximizes heat utilization within the entire carbon capture process. And lastly, the D-RM builder enables the automatic, systematic generation of data-driven dynamic reduced models from high-fidelity dynamic models.

In addition to the five major modules, FOQUS also bundles surrogate models. Since CFD and process simulations are time consuming and may fail to converge, surrogate models are simplified (reduced) models that are meant to be executed in a much shorter time. Examples of these simplified model modules are ALAMO and iREVEAL. The ALAMO module is used to create algebraic surrogate models for supporting large-scale deterministic optimization. The iREVEAL module is an automated tool to create reduced models from CFD simulations and export them in a form that process simulators can use.

FOQUS provides a graphical user interface and uses a flowsheet as a means of describing a process. The flowsheet is a core concept of FOQUS and is akin to a workflow. Simulations are initiated through FOQUS and executed on the Turbine Science Gateway, the parallel simulation execution management system for the CCSI Toolset. SimSinter is a library that interfaces with the wide variety of process simulators to allow models to be executed using a standardized interface from Turbine Science Gateway through SimSinter. A graphical user interface is provided with SimSinter to allow configuration of simulations for use with SimSinter. The last of the integration framework tools is the Data Management Framework (DMF). It interfaces with all of the computational tools and acts as a repository for storing, organizing, and sharing of data and models in CCSI. In addition, the DMF tracks the provenance associated with data and models.

IV. Implementation

In this section, we discuss implementation details of the integration framework tools in the CCSI Toolset, namely, FOQUS, Turbine Science Gateway, SimSinter, and the Data Management Framework. These tools provide the data management and simulation support for accelerating carbon capture in CCSI.

A. FOQUS

FOQUS (The Framework for Optimization and Quantification of Uncertainty and Sensitivity) is an end-user application that serves as the primary computational platform in the CCSI Toolset. It provides a graphical user interface and standard platform for several CCSI tools (Figure 2).

For processes such as post-combustion carbon capture from a coal fired power plant, there are many very different subsystems involved. The boiler, steam cycle, pollution controls, carbon capture, and compression systems are all distinct parts and simulation of some of these parts may be better done separately in different software by experts in each system. A critical feature of FOQUS is its ability to interface with and run commonly-used chemical engineering process modeling software. Through FOQUS, models constructed by many different people and using a variety of process simulation packages can be combined into a larger composite model referred to as a *flowsheet* and flowsheets can be combined into a *meta-flowsheet*. The meta-flowsheet allows connections between various flowsheets. For instance, the meta-flowsheet can be used to combine all of the carbon capture related system flowsheets into one large system, which can have recycle streams. FOQUS utilizes the tools SimSinter and the Turbine Science Gateway to run models using the external process simulation software. FOQUS provides a graphical interface to make linking simulations on Turbine into the flowsheet or meta-flowsheet relatively simple. The user can draw out their flowsheet where nodes represent simulations or calculations, and directed edges represent the flow of information between nodes. If a set of interdependent nodes is created (a flowsheet with recycle), FOQUS can automatically find an optimal set of tear edges (streams) and solve the problem iteratively.

The UQ portion of FOQUS provides an interface for the PSUADE tool, which does the underlying calculations. FOQUS first calls PSUADE to generate sets of samples to be run. This is then followed by FOQUS using Turbine and SimSinter to execute the simulations. FOQUS can take advantage of Turbine Science Gateways parallel execution capabilities to run flowsheets and meta-flowsheets in parallel; this provides a significant increase in speed when performing uncertainty quantification analysis. Once the results of the simulations are obtained, FOQUS can then use PSUADE to analyze results and generate relevant plots.

The simulation based optimization portion of FOQUS uses a plugin system that allows various derivative free optimization solvers to be added to FOQUS. The end user may write or modify derivative free optimization methods, or write FOQUS plug-in wrappers for libraries they have available. Many derivative free optimization solvers can benefit from running several simulations in parallel. For example, in evolutionary methods, an entire generation must usually be run before moving to the next iteration. Significant time savings can be achieved by running the simulations in parallel using Turbine Science Gateway.

The meta-flowsheet in FOQUS plays an important role in simulation based optimization. The optimum of a whole system may be different than independently optimizing sub systems. Since many of these models may not be compatible with any one process simulation package or may use a process simulation package that does not provide robust optimization methods, FOQUS's ability to link simulations and models built in different process simulators and run them efficiently is critical for uncertainty quantification and optimization of process-scale and plant-scale carbon capture process design.

B. Turbine Science Gateway

Turbine Science Gateway is a generic solution that can be extended to process modeling and simulation packages, and

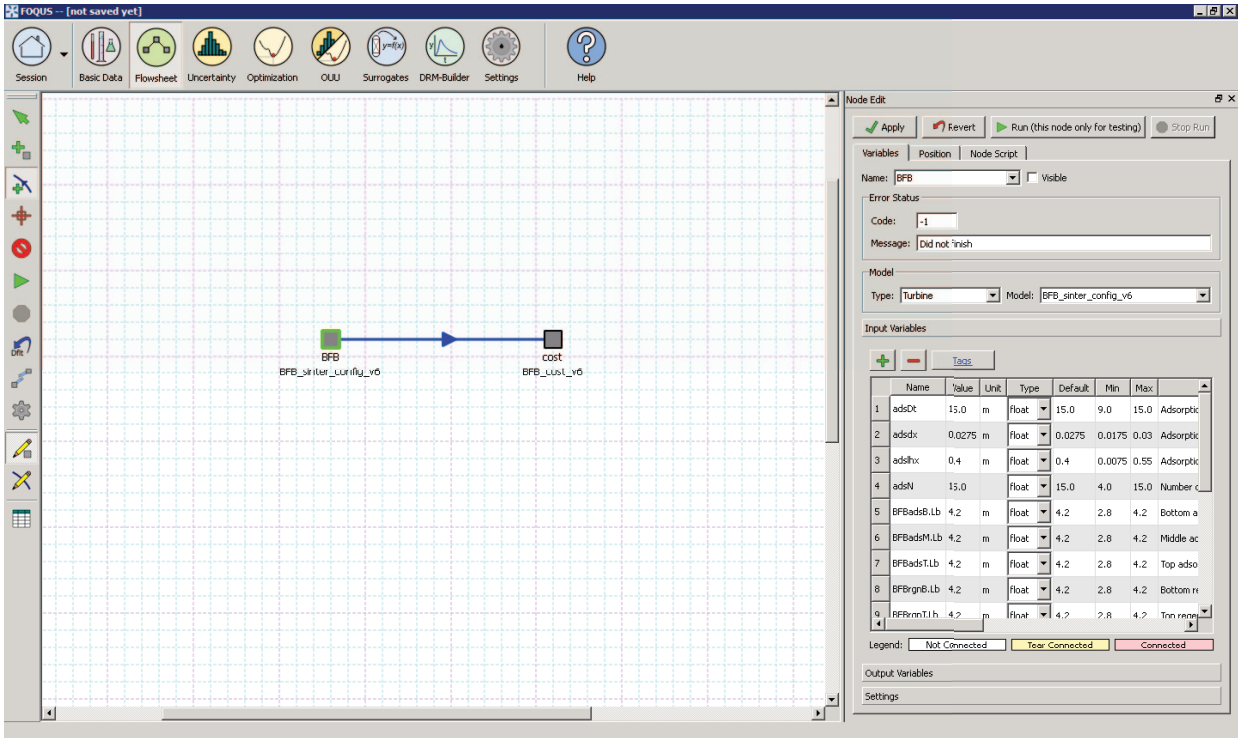

Fig. 2. Framework for Optimization and Quantification of Uncertainty and Sensitivity (FOQUS) with example Flowsheet construction

is essentially a batch system that provides staging of input and output files. The Turbine Science Gateway is composed of four components; a Turbine Web application, a Turbine Server providing an execution environment for running and managing scientific applications, a Turbine Database for storing results, and a Turbine Client interface that provides the local machine a connection to the other components. The Turbine Web allows users to retrieve, monitor, run, and manage scientific applications running on the back-end. The Turbine Server back-end execution framework can be configured for single machines, networked workstations, cluster, or Cloud computing resources such as Amazon Web Services EC2. The cluster and EC2 deployments allow for parallel application executions, where on EC2, the CCSI team has successfully executed Aspen Plus simulations using deployments of 400 concurrent instances. When deploying with EC2, the Turbine Science Gateway harnesses Amazon EC2 spot instances allowing simulation executions to happen using low costs. Through the use of spot instances, running simulations in the Cloud now become more economical than having a cluster of machines. In our experience, parallelization dramatically increases application throughput and decreases the time to solution from weeks to hours and months to days.

Turbine Science Gateway is designed to operate primarily in Windows since that is the platform of the process simulations and users. The Turbine Client, on the other hand, is a python library that is platform independent. Turbine Clients interact with the Turbine Web through a RESTful Web interface or directly through HTTPs. The RESTful Web API defines 5 kinds of Web resources: Application, Simulation, Job, Consumer, and Session. Short descriptions of each Web resource is provided below:

1) Application: The Application resource is read-only, provides descriptions of the supported scientific applications, and specifies the files required for an execution.

2) Simulation: A Simulation resource references the target application, and contains an appropriate application model and sinter configuration file (see IV-C). These are specified as required staged-in files in the Application resource.

3) Job: Each simulation run is represented by a Job resource. This is essentially a single execution request. A Job must reference a Simulation resource and contains metadata concerning the progress of the job through the system (status, timestamps, etc), input data set, and output data set specified by the sinter configuration file. In addition, each Job resource contains a reference to the Consumer resource which identifies the computational resource on which the Job was executed.

4) Consumer: The Consumer resource represents workers that are running the scientific application. Consumers are each associated with a computational resource where each consumer process must be registered before running Jobs.

5) Session: Each Session resource is a set of Job resources that can be controlled as a group.

In addition to the REST API, Turbine Web also provides a Python library that is designed to be scriptable, returning structured JSON [30] output that can be easily consumed by other tools. When the Turbine Web receives a request to run a simulation or flowsheet it stores the request in the Turbine

database. It also triggers Turbine Server to launch the tasks.

The Turbine Server includes a Turbine Worker component which executes and directly manages each underlying simulation process through SimSinter. A Turbine Worker also requires a configuration string to connect to the database, which is set up during installation. Each supported process simulation package (i.e. Aspen Plus, ACM, gPROMS, Excel) has a corresponding Turbine Worker application that manages the scientific application through SimSinter.

A Turbine worker will start and connect to the database and query for a matching simulation request or job. For example, an ACM Turbine Worker will query for an ACM job. When the worker finds a job, it creates a working directory and then downloads and stores the input files there. It then utilizes SimSinter to manage the scientific application. One of the staged files placed in the working directory is the SimSinter configuration JSON formatted file. This file is required to use SimSinter, it contains information for executing the underlying scientific application, like the simulation file name (eg. ACMF). This file also specifies input parameters the user can change for a particular simulation, and output parameters which are retrieved from the underlying application after the job has completed. The worker manages most of the state changes of a job, moving it sequentially from submit, setup, running, finished or error. Additional workers can be added to form a Turbine Cluster by installing the Turbine Worker component on additional machines.

C. SimSinter

SimSinter is a standard interface library for driving single-process Windows based process simulation software. Through the use of SimSinter, the CCSI Toolset is able to provide extensible support with various simulation tools. Within the CCSI Toolset, SimSinter serves as a connector between the Turbine Science Gateway and each individual process simulation tool. SimSinter achieves this by providing a standardized execution request interface to Turbine Worker and uses a custom interface to interact with process simulation tools. It is designed to run simulations on both large-scale computers, such as on a cluster or in the Cloud, or on users local machine.

Since SimSinter supports commercial simulations, it must be executed on a machine that has the simulator and simulator licenses installed. For example, SimSinter can run Aspen Plus simulations on a desktop computer that has Aspen Plus installed and the necessary Aspen Plus licenses. In the case where users do not already have a local license copy, users can still use SimSinter remotely via the Turbine Science Gateway. However, the sinter configuration file will need to have already been created. The main purpose of a sinter configuration file is to identify simulation input and output variables that are available to users in FOQUS. Sinter configuration files are written in the JSON format and is typically created by the model creator. Outputs for SimSinter are also written using the JSON format. SimSinter can be called via the Turbine Worker, through Microsoft Excel, or standalone tools that are command-line based. As part of simplifying the configuration process for users, a graphical user interface is bundled with SimSinter, called the SinterConfigGUI.

SimSinter leverages existing custom interfaces of process simulators and is based on .Net and COM. In current implementations, SimSinter currently supports four kinds of commercial simulators: Aspen Custom Modeler, Aspen Plus, gPROMS, and Microsoft Excel. A brief description of these simulators and how they are supported are provided below.

1) Aspen Custom Modeler: Aspen Custom Modeler (ACM) is an equation-oriented chemical process simulator developed by AspenTech. ACM provides a Microsoft COM interface for driving simulations. SimSinter uses the COM interface to launch ACM, inserts input variables according to the paths provided in the sinter configuration file, run the simulation, read out the output variables according to their paths, and finally closes the simulation.

2) Aspen Plus: Aspen Plus is a sequential-modular chemical process simulator also developed by AspenTech. Aspen Plus provides a Microsoft COM interface for driving simulations. SimSinter uses the COM interface to launch Aspen Plus, inserts the input variables according to the paths provided in the sinter configuration file, run the simulation, read out the output variables according to their paths, and closes the simulation at the end.

3) gPROMS: gPROMS is an equation-oriented chemical process simulator developed by Process Systems Enterprise (PSE). gPROMS provides a simulator called gORUN_XML that takes inputs and outputs from an XML file. SimSinter is able to support gPROMS by taking in an input JSON file and creating a gORUN_XML readable XML file, which is then executed using gORUN_XML. When the execution completes, SimSinter reads the results from the XML file and converts back to JSON. The simulation file provided to SimSinter must be is the .gENCRYPT format, as gORUN_XML will only run encrypted gPROMS simulations. SinterConfigGUI cannot currently be used with gPROMS, so the sinter configuration file currently must be written by hand.

4) Microsoft Excel: Microsoft Excel is a commonly used spreadsheet application. It is widely used by scientists for simple calculations such as costing. Excel provides a COM interface that SimSinter uses to access spreadsheet cells. Macros inside Excel are also supported by SimSinter.

D. Data Management Framework

The Data Management Framework (DMF) serves as a framework for tracking all files related to a carbon capture investigation. The framework provides support for collecting CCSI data files. Additionally, it also manages and stores metadata associated with these files. Furthermore, the Data Management Framework maintains data provenance and provides the necessary tools and interfaces to allow scientists to quickly identify outdated simulations using provenance traces. Consequently, simulations can be rerun when a model is updated and the time-consuming operations such as optimization can be performed using the same configurations as were used on the previous version. This capability enables

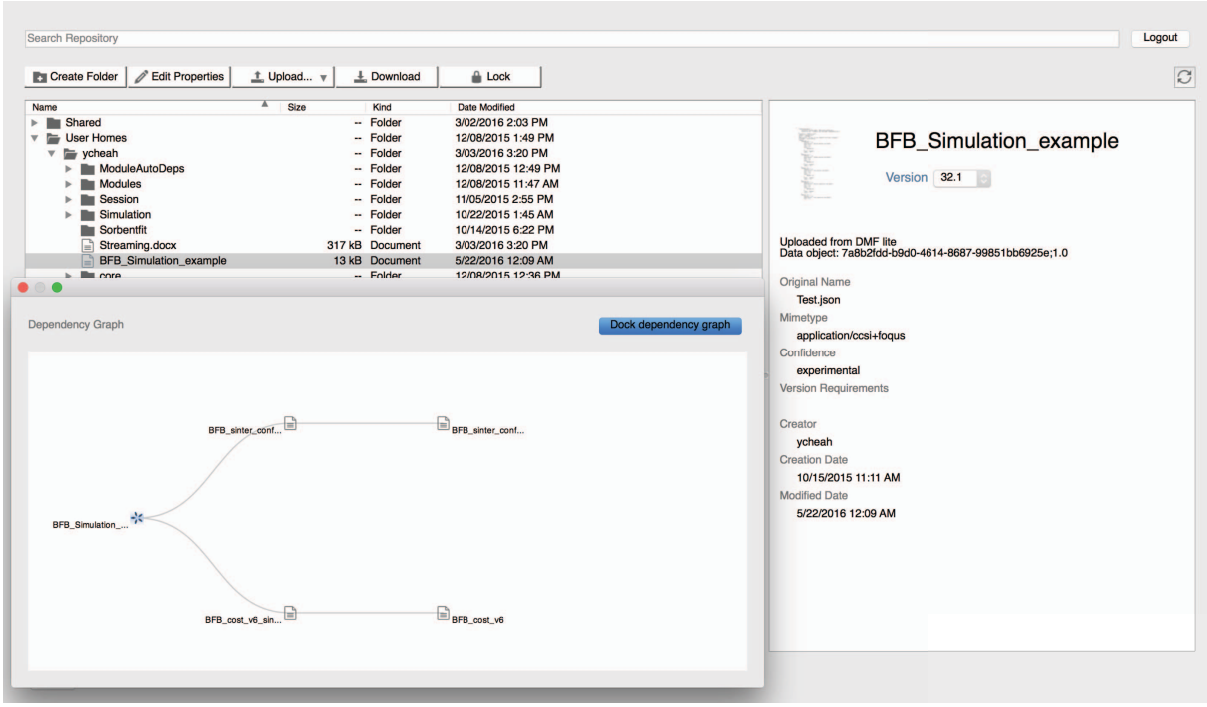

Fig. 3. CCSI Data Management Framework: DMF Browser with data provenance visualization

the design lifecycle and allows decisions to be made on information that was generated using the latest basic property and model information. The DMF backend currently comes in two flavors: *DMFServ*, a server-based feature-rich flavor and *DMF Lite*, a lightweight local machine solution. A separate client, *DMF Browser*, with a Graphical User Interface is also available and bundled with the FOQUS tool.

DMFServ is a server-based, central, network-accesible service that is suitable for deploying in an organization. This version of the Data Management Framework facillitates group interactions by providing a venue to share documents between multiple, geographically distributed individuals while tracking relationships between files. The server-based DMFServ is intended to store all the data related to a project from the gathering of experimental data, to the development of basic models, to the development of process models, and ultimately optimization and UQ studies and their results.

In addition to providing basic file sharing services, DMFServ is set up to store multiple revisions of each file in the system, allowing the DMFServ to provide a historical archive of the evolution of the project. Older versions of data and model files are stored to enable backtracking or to provide a means to compare the final product with earlier work.

Data provenance tracking is also provided with the DMFServ. As files are added to the framework, they are assigned unique IDs and versions. Users are able to specify relationships between individual files to indicate dependencies as needed. Collections of these dependency relationships help establish the provenance chain for each file in the system. Users are thus able to see at a glance when dependencies of a simulation have been modified, indicating that the simulation may not be incorporating the best available data. These allows users to rerun existing simulations to obtain updated results.

The smaller lightweight DMF Lite provides a solution for users that tend to operate on just their local machine. Like the DMFServ, it is able to store multiple revisions of each file and provide provenance tracking. In our implementation the DMF Lite lacks the capability for sharing data directly and searching over files because it is meant to be complimentary with the DMFServ. Users of the DMF Lite are able to push their data onto the DMFServ if needed and this will open up better integrated collaborative capabilities for users.

A standalone client called the DMF Browser is also provided with the CCSI Toolset. This client can be operated both with or without the use of a graphical user interface. In the case of interfacing with other CCSI Toolset computational tools, the DMF Browser is often operated without the use of a graphical user interface. Alternatively, users can operate it in graphical user interface mode (Figure 3). The graphical user interface is in the form of a file browser and shows the metadata. These metadata includes basic file metadata like the name, author, creation time, versions of files, location of the files. In addition, users can also browse data provenance dependency graphs. The rendering of provenance graphs uses the D3 javascript library [31]. Users are able to quickly identify from the provenance graphs detailed information about data dependencies, version information, and also whether file dependencies are outdated. When the DMF Browser is used with the DMFServ

backend, additional functionality such as searching over the repository and setting basic file permissions are available. The DMF Browser can be used to browse multiple DMF backends if needed. A command-line client called the *DMF Basic Data Ingester* was also developed to automate the process for basic data modelers to channel modeling data straight into the Data Management Framework.

The DMFServ uses the Alfresco Community Edition repository [32] as the backend repository. Alfresco was chosen as a backend repository due to its rich collaborative features, versioning and search features, and also support for a wide variety of data formats. In addition, Alfresco comes in open source and commercially supported versions allowing it to be deployed in large corporations and small contractors. DMFServ is implemented using a mixture of Python and Java. The DMF Lite, on the other hand, uses the Git version control system in place of Alfresco as its backend repository. The DMF Lite is written using only Python.

In the CCSI Toolset, the Data Management Framework is tightly integrated with FOQUS, Turbine Science Gateway, and SimSinter. With SimSinter, users can store their simulations right after creation or when making edits in the Data Management Framework. Users can also ingest simulations to the Data Management Framework through the FOQUS interface. When users are ready to run their experiments, the Data Management Framework allows users to choose simulations. These simulations will be synchronized with the Turbine Science Gateway before execution so that the correct versions of simulations are available before a run. When a run is completed, output files will be generated and users can choose to save the results of simulations to the Data Management Framework directly or via FOQUS after analysis.

V. Discussion

In this section, we provide further information about the user experience work in the CCSI Toolset. We also present and discuss some of the real life use cases of the integration framework including ones where the Turbine Science Gateway executed thousands of simulations.

A. User Experience Design

The Industry Advisory Board (IAB) was involved in the CCSI project from the beginning and our user data collection started with informal interviews and a survey of project industry participants to better understand current simulation tools, new equipment design processes, and the industry working environment. The Industry Advisory Board's 20–30 members represent a wide variety of aspects of the power industry including power plant operators, equipment designers, and utilities. Thus their perspectives spanned the range of use cases we were targetting. One of the first things we learned was that the simulation tools in use cover a wide spectrum including many commercial, open source, and custom tools. In addition, many of the groups have developed extensive experience and confidence as well as large model libraries in these simulation tools. This survey along with the interviews of IAB members made it clear that enabling existing process simulators was an important requirement to facilitate adoption of the CCSI Toolset by our target industry. In addition, survey results also made clear that the predominant group of users in the process simulation space were Windows users and that the CFD users were primarily using a UNIX variant. The process simulation teams also rarely had access to or experience with high performance computing platforms or even clusters. It had been our expectation before the survey that gaining access to parallel resources to execute simulations required for uncertainty and optimization studies would be easy. However, it became clear from the survey that availability of on-demand resources in the Cloud would be valuable to users with no other resources. The above examples are the key constraints learned from the users. Results from the survey and interviews also provided other more minor constraints used in designing the integration framework. In particular, the interviews provided insight into why and how CFD and process simulations are created and conducted at each industry. It also provided insight into the roles of contractors in the design process.

Throughout the CCSI project, each graphical user interface went through multiple rounds of design with experts in Human Computer Interaction. These rounds allowed us to refocus each interface from the perspective of the developer to an interface designed to aid the user of the software and fit with their design process. The reactions to the resulting graphical user interfaces have been quite positive and several of the industry members have adopted the CCSI Toolset for use in their local environment. In our follow-on project we will be supporting that use for development of multiple large-scale carbon capture demonstration projects.

B. Parallel Simulation Use Case

The CCSI Toolset integration framework has been used extensively in the last two years. In this subsection, we pick a particular four days of heavy usage to demonstrate the performance of the framework when executing thousands of simulations. We demonstrate this through use of the Turbine Science Gateway executing a ACM Hybrid Split Optimization experiment as shown in Figure 4.

The simulations were executed spanning more than four days (from Jan 21st to Jan 26th) using 50 Amazon EC2 m3-large Virtual Machines. The x-axis shows the start time, in this case the date of the day where the simulation was started. The y-axis measures the runtime in minutes for each simulation. In the figure, the blue circles refer to simulations completed successfully and the green circles indicate failed simulations. As expected, the blue circles often reflect a shorter runtime. In comparison, the simulations represented by green circles show longer runtimes and fail eventually. Of the 74131 simulations that were executed during this period of time, 62995 of the simulations completed successfully while 11135 of them failed. Turbine Science Gateway does detect failures and attempts a single restart after a failure.

Fig. 4. Plot of simulation runtime versus start time of simulation execution

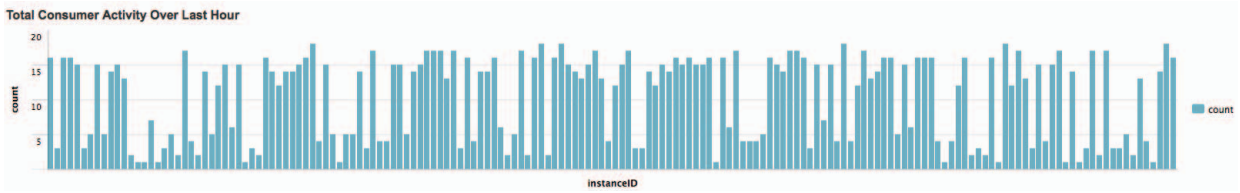

Fig. 5. Histogram of number of simulations executed by Virtual Machines

We are also able to run real-time log analysis on simulation runs. In Figure 5, we present a histogram of the number of simulations executed by Virtual Machines over an hour. Each bin in the histogram represents a single Virtual Machine (VM) and contains the number of simulations executed. This includes both simulations that succeded and those that failed. Bins that have larger simulation counts are due to the VMs executing many simulations that complete successfully. On the other hand, some bins have lower counts due to executing non-converging simulations. Non-converging simulations have a time-out set at approximately 30 minutes.

With the deployment of the Turbine Science Gateway, CCSI has seen computational performances improved drastically. An integrated mass transfer model used to require optimization times of up to 12 hours for a single iteration on a single processor machine. The same optimization executed using the Turbine Science Gateway with 4–6 consumers now only require less than a third of the time (2 hours and 45 minutes).

VI. Conclusion

Traditionally, the end-to-end process for carbon capture technologies in the energy sector consumes tens of years. The Carbon Capture Simulation Initiative (CCSI) is a joint effort between national laboratories, industry and academia with the goal of shortening the amount of time needed from discovery to deployment of carbon capture technologies. As part of CCSI, we have developed the integration framework in the CCSI Toolset consisting of computational tools with simulation models and optimization techniques. In this paper, we focus our discussion on the simulation framework and data management tools that are part of the CCSI Toolset. The simulation framework provides adapters to enable running a variety of commercial simulation packages using custom models. Moreover, the framework enables scientists to run and manage thousands of simulations to perform optimizations and uncertainty quantification. Components of the CCSI Toolset are connected through the use of a data management system that stores data to a repository and enables tracking of provenance for each simulation as well as its associated components. The CCSI Toolset has been deployed and is in use by several groups of researchers for development purposes and commercial entities for designing and deploying carbon capture equipment. The integration framework within the CCSI Toolset provides the capabilities needed to allow tasks like optimization and uncertainty quantification of carbon capture processes to be executed in hours instead of the weeks and months that were previously required. The integration framework also provides the data managment tracking of

versions and results needed to support the lifecycle of a project and decision making.

Future work will concentrate on augmenting the existing CCSI Toolset with tools to help improve decision making. With the wealth of data that the CCSI Toolset collects, it is beneficial to present these data in a manner such that trial-and-error efforts are minimized and decisions can be more informed. This will likely involve the implementation of a dashboard that is able to present and integrate existing data in an effective manner.

ACKNOWLEDGMENT

The authors would like to thank Christine Moran from INRIA, Lavanya Ramakrishnan from Berkeley Labs and the broader CCSI team for providing suggestions and feedback for the writing of this paper. Special thanks to Keith Beattie and Paolo Calafiura for managing the CCSI toolset software releases. This project was funded by the US Department of Energy under the Carbon Capture Simulation Initiative through the following contracts: LBNL DE-AC02-05CH11231, LLNL FEW0180, and NETL RES-0004000.6.600.007.002.

DISCLAIMER

REFERENCES

[1] J. Jenkins and S. Mansur, "Bridging the clean energy valleys of death," *Breakthrough Institute, November*, 2011.

[2] R. S. Haszeldine, "Carbon capture and storage: how green can black be?" *Science*, vol. 325, no. 5948, pp. 1647–1652, 2009.

[3] L. Ramakrishnan, S. Poon, V. Hendrix, D. G. ter, G. Pastorello, and D. Agarwal, "Experiences with user-centered design for the tigres workflow ap i," in *Proceedings of the 10th IEEE International Conference on e-Sci ence*, Guaruja, Brazil, October 2014.

[4] C. Aragon, S. Poon, G. Aldering, R. Thomas, and R. Quimby, "Using visual analytics to develop situation awareness in astrophysics,," *Journal of Information Visualization*, 2009.

[5] A. J. Hey, S. Tansley, K. M. Tolle *et al.*, *The fourth paradigm: data-intensive scientific discovery*. Microsoft research Redmond, WA, 2009, vol. 1.

[6] ANSYS Engineering Knowledge Manager (EKM). [Online]. Available: http://www.ansys.com/Products/Platform/ANSYS-EKM

[7] M. Freshley, S. Hubbard, G. Flach, V. Freedman, D. Agarwal, B. Andre, Y. Bott, X. Chen, J. Davis, B. Faybishenko *et al.*, "Advanced Simulation Capability for Environmental Management (ASCEM) Phase II Demonstration," Tech. Rep., 2012.

[8] F. MD, G. Flach, H. Wainwright, M. Rockhold, V. Freedman, D. Moulton, P. Dixon, and J. Morse, "Applications using the advanced simulation capability for environmental management toolset," in *Waste Management Symposia*, Phoenix, Arizona, March 2016.

[9] D. A. Agarwal, B. Faybishenko, V. L. Freedman, H. Krishnan, C. L. Gary Kushner *et al.*, "A science data gateway for environmental management," *Concurrency and Computation: Practice and Experience*, p. To appear, 2015, also appeared at GCE15: The 10th Gateway Computing Environments Workshop, 9/30 to 10/1, 2015, Boulder, CO.

[10] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, and H. T. Vo, "Vistrails: visualization meets data management," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. ACM, 2006, pp. 745–747.

[11] H. I. and C. T. Ludwig, "Improving processes for user support in e-science," in *IEEE eScience 2014 Conference - Works in Progress session*, 2014.

[12] V. Garonne, G. A. Stewart, M. Lassnig, A. Molfetas, M. Barisits, T. Beermann *et al.*, "The atlas distributed data management project: Past and future," *Journal of Physics: Conference Series*, vol. 396, no. 3, p. 032045, 2012. [Online]. Available: http://stacks.iop.org/1742-6596/396/i=3/a=032045

[13] T. Maeno, "Panda: distributed production and distributed analysis system for atlas," *Journal of Physics: Conference Series*, vol. 119, no. 6, p. 062036, 2008. [Online]. Available: http://stacks.iop.org/1742-6596/119/i=6/a=062036

[14] J. Gray, D. T. Liu, M. Nieto-Santisteban, A. Szalay, D. J. DeWitt, and G. Heber, "Scientific data management in the coming decade," *ACM SIGMOD Record*, vol. 34, no. 4, pp. 34–41, 2005.

[15] S. Miles, P. Groth, M. Branco, and L. Moreau, "The requirements of using provenance in e-science experiments," *Journal of Grid Computing*, vol. 5, no. 1, pp. 1–25, 2007.

[16] R. S. Barga, Y. L. Simmhan, E. Chinthaka, S. S. Sahoo, J. Jackson, and N. Araujo, "Provenance for scientific workflows towards reproducible research." *IEEE Data Eng. Bull.*, vol. 33, no. 3, pp. 50–58, 2010.

[17] B. Ludäscher and B. Plale, *Provenance and Annotation of Data and Processes: 5th International Provenance and Annotation Workshop, IPAW 2014, Cologne, Germany, June 9-13, 2014. Revised Selected Papers*. Springer, 2015, vol. 8628.

[18] R. M. Fujimoto, A. W. Malik, and A. Park, "Parallel and distributed simulation in the cloud," *SCS M&S Magazine*, vol. 3, pp. 1–10, 2010.

[19] G. D'Angelo, "Parallel and distributed simulation from many cores to the public cloud," in *High Performance Computing and Simulation (HPCS), 2011 International Conference on*. IEEE, 2011, pp. 14–23.

[20] L. Wang, J. Tao, R. Ranjan, H. Marten, A. Streit, J. Chen, and D. Chen, "G-hadoop: Mapreduce across distributed data centers for data-intensive computing," *Future Generation Computer Systems*, vol. 29, no. 3, pp. 739–750, 2013.

[21] Docker. [Online]. Available: https://www.docker.com/

[22] CoreOS rkt. [Online]. Available: https://coreos.com/rkt/

[23] W. Gentzsch, "Linux containers simplify engineering and scientific simulations in the cloud," in *Information and Computer Technology (GOCICT), 2014 Annual Global Online Conference on*. IEEE, 2014, pp. 22–26.

[24] A. Jain, S. P. Ong, W. Chen, B. Medasani, X. Qu, M. Kocher, M. Brafman, G. Petretto, G.-M. Rignanese, G. Hautier *et al.*, "Fireworks: a dynamic workflow system designed for high-throughput applications," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 17, pp. 5037–5059, 2015.

[25] S. Pronk, S. Páll, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M. R. Shirts, J. C. Smith, P. M. Kasson, D. van der Spoel *et al.*, "Gromacs 4.5: a high-throughput and highly parallel open source molecular simulation toolkit," *Bioinformatics*, vol. 29, no. 7, pp. 845–854, 2013.

[26] Qdo. [Online]. Available: https://bitbucket.org/berkeleylab/qdo

[27] Apache Spark. [Online]. Available: http://spark.apache.org

[28] Amazon Flow Framework. [Online]. Available: https://aws.amazon.com/swf/details/flow/

[29] C. Tong, "The psuade software package version 1.0," *LLNL code release UCRLCODE-235523*, 2007.

[30] Javascript Object Notation. [Online]. Available: http://www.json.org/

[31] D3 Javascript Library. [Online]. Available: https://d3js.org/

[32] Alfresco Content Management Repository. [Online]. Available: https://www.alfresco.com