

Copyright ASTM International. All rights reserved, date, time. Downloaded by (Prakashan Korambath, Hari S. Ganesh, Jianwu Wang, Michael Baldea, and Jim Davis), pursuant to Author/Copyright Owner Agreement. No further reproduction authorized.

Korambath, Prakashan, Hari S. Ganesh, Michael Baldea, Jianwu Wang, and Jim Davis. "Use of On-Demand Cloud Services to Model the Optimization of an Austenitization Furnace." *Smart and Sustainable Manufacturing Systems* 2, no. 1 (November 13, 2018): 165–79.
<https://doi.org/10.1520/SSMS20180024>.

<https://doi.org/10.1520/SSMS20180024>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

Please provide feedback

Please support the ScholarWorks@UMBC repository by emailing scholarworks-group@umbc.edu and telling us what having access to this work means to you and why it's important to you. Thank you.



Smart and Sustainable Manufacturing Systems

Prakashan Korambath,¹ Hari S. Ganesh,² Jianwu Wang,³ Michael Baldea,²
and Jim Davis⁴

DOI: 10.1520/SSMS20180024

Use of On-Demand Cloud Services to Model the Optimization of an Austenitization Furnace

VOL. 2 / NO. 1 / 2018

Prakashan Korambath,¹ Hari S. Ganesh,² Jianwu Wang,³ Michael Baldea,² and Jim Davis⁴

Use of On-Demand Cloud Services to Model the Optimization of an Austenitization Furnace

Reference

Korambath, P., Ganesh, H. S., Wang, J., Baldea, M., and Davis, J., "Use of On-Demand Cloud Services to Model the Optimization of an Austenitization Furnace," *Smart and Sustainable Manufacturing Systems*, Vol. 2, No. 1, 2018, pp. 165–179, <https://doi.org/10.1520/SSMS20180024>. ISSN 2520-6478

ABSTRACT

This article describes a smart manufacturing framework, comprising an on-demand, cloud-based deployment of a modeling application in a manufacturing operation. A specific use case, the optimization of the austenitization of steel parts, is presented. The framework uses a Kepler workflow as a cloud service to orchestrate and manage the data and computations required to implement a run-time model-based control and optimization approach on Amazon Web Services (AWS) resources. Austenitization is an energy intensive heat treating process commonly employed to harden and strengthen ferrous metals, such as steel. Pre-finished steel parts are heated to a specific temperature in a continuously operating industrial austenitization furnace without oxidizing the surface. The steel parts are then rapidly cooled or quenched in an oil bath. There is significant potential to optimize energy productivity by managing the energy usage needed to achieve the properties of the metal part instead of managing to operational process settings. Models of this process, which predict the furnace energy consumption and temperatures of parts as a function of time and position in the furnace and map temperatures to properties, have been previously developed; however, for operational use, the data and models need to be orchestrated for run-time operation, access to infrastructure, scalability, security, and support. A cloud-based approach is an alternative to the on-premise approach, in which an infrastructure for data, computational, and security needs to be built and maintained to support the application. A workflow service makes it

Manuscript received March 9, 2018; accepted for publication September 21, 2018; published online November 13, 2018.

¹ Institute for Digital Research and Education, University of California, Los Angeles, 5308 Math Sciences, Los Angeles, CA 90095, USA (Corresponding author), e-mail: ppk@idre.ucla.edu, <https://orcid.org/0000-0003-0531-7384>

² McKetta Department of Chemical Engineering, 200 E. Dean Keeton St Stop C0400, Austin, TX 78712, USA

³ Department of Information Systems, University of Maryland, Baltimore County, Baltimore, MD, USA

⁴ Institute for Digital Research and Education, University of California, Los Angeles, 5308 Math Sciences, Los Angeles, CA 90095, USA

possible to combine and sequence simulation and optimization software applications developed in several distinct MATLAB (The Mathworks Inc., Natick, MA) model configurations that are needed for various data-based calculations. The final output of the computation is the optimal operating condition of the furnace that minimizes the fuel consumption without violating the part target specifications. The workflow can be triggered on demand by an operator of the furnace or run at periodic intervals. All the computational resources required are instantiated and run at the start of the workflow and shutdown at the end of the workflow.

Keywords

cloud, smart manufacturing, workflow, real-time data, austenitization

Introduction

Smart manufacturing [1,2] is the business, technology, infrastructure, and workforce practice of optimizing manufacturing through the use of engineered systems that integrate operational technologies and information technologies (OT and IT, respectively). It is industry-wide terminology denoting the interoperable application of OT and IT technologies in the form of sensor, data, modeling, control, and actuation applications used to increase unit operation performance, value, and supply chain productivity and precision with runtime product qualification. Previous works [3,4] describe how we have used Kepler (The Kepler Project, UC Davis, UC Santa Barbara, UC San Diego, CA) scientific workflow [5] software to automate the process of collecting input data, running models, and visualizing output data to increase the performance of a commercial industrial-scale steam methane reforming furnace with improved management of the energy distribution in the furnace. In these, application workflows were implemented on compute resources that were statically deployed with all software installed locally and with public Internet Protocol addresses known in advance. OT modeling was limited to a fixed set of infrastructure resources and run as a single application (one-off) on the dedicated resources. By contrast, this article describes a deployment strategy in which the workflow, modeling, and compute resources take advantage of the instantaneous on-demand compute capabilities found in cloud computing. Similar application workflows, now augmented with cloud infrastructure capabilities, are used for the optimization and control of an austenitization furnace to minimize energy consumption without compromising product quality. For this use case, OT refers to the modeling, control, and management of the austenitization furnace, and IT refers to digital and cloud technologies that support or enable these OT functions.

Manufactured steel that is cooled at a slow rate is quite soft and cannot be used in common applications. Hence, the metal is usually heat-treated by a process called austenitization to improve the hardness and strength. During austenitization, steel parts are heated to a specific temperature (typically above 1,000 K) without oxidizing the surface to transform the metal microstructure into austenite. The parts are then rapidly cooled in an oil quench bath to induce the desired metallurgical properties such as hardness, toughness, tensile strength, etc. The manufacturing process considered in this work is a continuously operating commercial austenitization furnace, as presented in “Operational Demonstration of an Austenitization Process.” Heng et al. [6] developed several mathematical models of the furnace and its operation, each constructed in MATLAB (The

Mathworks Inc, Natick, MA) [7], to predict the furnace energy consumption and temperature distribution of the metal parts. The models were extended by Ganesh, Edgar, and Baldea [8] and Ganesh et al. [9] to also capture microstructure changes and thus metallurgical properties of the parts during heating. By integrating these models into a workflow that orchestrates the data and the models in an ordered manner, it is possible to relate the furnace operation to microstructure properties and then the operational data and properties to calculate the optimized furnace set points. These publications describe how the individual models are combined into an overall workflow to optimize the furnace operation based on a response surface methodology and in so doing, increase the energy productivity of the system.

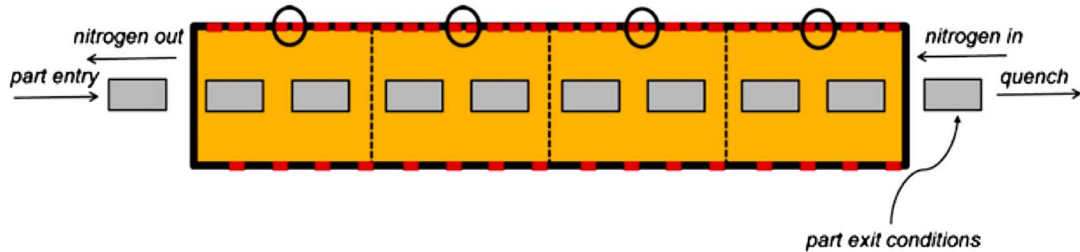
This article extends this earlier modeling work to describe a novel on-demand cloud workflow service for orchestrating sensor, data, model, and optimization workflows in an operational setting. The particular use case demonstrates a smart manufacturing workflow orchestration that optimizes an austenitization furnace operation by relating in situ property qualifications to the furnace operating conditions. Key contributions include (1) a demonstration of an industrial OT use case automatically executing several mathematical models that are orchestrated in a scientific workflow (e.g., some of the outputs of one program are provided as the inputs to the next program) and (2) an on-demand cloud workflow service (IT), which instantiates virtual machines (VMs) only when needed for model calculations. Initial deployment and tests show the OT use of the workflow can support operators with real-time monitoring and decision-making to improve the energy productivity and performance of the austenitization process. Because the workflow captures the entire sequential operational process, a factory operator can focus on the function and operational activity of the workflow, and the enabling IT remains embedded. While IT and cloud technologies are described in this article, at the operational level, the need-to-know about such IT matters like Amazon Web Services (AWS) [10] interfaces, the running MATLAB software, etc. is minimized.

The article is organized as follows: “Operational Demonstration of an Austenitization Process” describes the austenitization furnace and the operational use case. “The Operational Technology” describes the model based optimization strategy, i.e., the OT. “Setting up the IT as a Cloud Computation Architecture” summarizes the IT side of the computational resources and architecture on which the calculations are performed, as well as how the Linux images are created. This section is intended to bring out the depth of the IT and what is embedded and hidden from operational consideration. “Joining OT and IT with Kepler Workflows” gives a detailed description of the Kepler workflows that were configured to do the data transfer and computations. In “Results from the Modeling Calculation”, we show some graphical results from the MATLAB-based modeling calculations that can be visualized by the operator or the workflow itself to illustrate a working system with novel and embedded IT underpinning.

Operational Demonstration of an Austenitization Process

The overall energy consumption in the United States is 100 quadrillion Btu (or quads) per year [11]. The metal processing industry accounts for about 2 % of the total energy demand [12]. For a typical forging and heat treatment process such as austenitization, around 60–80 % of the energy is consumed as fuel for reheating and heat treating furnaces [13].

FIG. 1 A schematic of the two-dimensional model of the austenitization furnace developed in Ganesh, Edgar, and Baldea [8]. The parts, represented by grey rectangles, enter from the left and exit from the right. Nitrogen, the inert blanket gas, enters from the right and exits from the left. The conveyor belt is neglected in the model. The furnace is divided into four temperature control zones. The dotted lines represent the boundaries between the zones. The temperature of a zone is the temperature of the top middle insulation surface (circled) in it. The conditions of the parts exiting the furnace and before getting quenched are recorded for optimization purposes.



Metal production and processing are not only energy intensive but also one of the major contributors to CO₂ emissions [14–16]. There is significant opportunity for improved energy productivity with improved operation and control. In the austenitization process, metal parts are frequently overheated because of the difficulty to sense or predict the temperature of the part during operation, especially the interior of the part. Overheating is the result of operating conservatively to avoid any cold regions that would lead to a defective part.

Fig. 1 shows a schematic of a typical Holcroft [17] austenitization furnace (AFC-Holcroft US, Wixom, MI). The furnace considered in this work operates in a continuous manner and is heated indirectly by natural gas-fired radiant tube burners on the ceiling and floor of the furnace in an inert gas environment (nitrogen) to prevent surface oxidation of parts. Metal parts are loaded in a tray on a conveyor belt that runs the entire length of the furnace. At the exit, there is an oil bath for quenching. While the parts are predominantly heated by radiation, the blanket gas is heated by convection and adds some heating effect to the part. Because it is difficult to sense and control the part temperatures (especially the interior) directly, part temperatures are indirectly controlled by controlling the furnace temperature distribution and dwell time that the part is in the furnace. To provide a level of control, the furnace is divided into four temperature control zones; the temperatures within each of the four zones are precisely controlled with feedback controllers that adjust the fuel flow rates to the burners. The temperature-controlled zones also indirectly control the microstructural changes that are dependent on the internal heat exchange and conduction at the surface and within the part. The temperature distribution within the part must be controlled for the product to be structurally sound and without defects. An empirical relation reported by Anelli [18] is used to predict the austenite grain size, a key microscale property as a function of temperature and history of heating. The reader is referred to the work by Ganesh, Edgar, and Baldea [8] and Heng et al. [6] for detailed discussion of a two-scale model and solution.

The OT

The overarching objectives of the modeling approach are to identify the optimal operating conditions of the furnace that ensure the properties of all the parts during the austenitization process while optimizing energy usage. In this section, we present a short

description of the optimization strategy based on a surrogate modeling approach developed in Ganesh, Edgar, and Baldea [8] and Heng et al. [6] to identify zone temperature set points. More specifically, the energy per individual part is minimized when parts are heated just past the temperature threshold required for developing uniform properties through the volume of the part.

While defining the part (output) variables in the optimization problem, we consider the simulation results of only the constant or steady-state input/output operating regime of the furnace, ignoring the beginning and end of each batch. The furnace can process a maximum of eight parts at any given point in time. Therefore, the first and last eight parts of a batch are processed under the “unsteady-state” operating regimes (i.e., startup/shutdown) of the furnace wherein there are lesser heat sinks in the furnace compared to the steady-state operation. To ignore the dynamics of reaching a new steady-state, one and a half cycles are neglected before recording the part exit conditions. For processing a batch of 40 parts sequentially, the output conditions are averaged over the middle 16 parts during the steady-state operating regime of the furnace and are referred to as the part output conditions. The part surfaces are heated by radiation, and the interiors are heated by conduction. The inhomogeneous temperature distribution within each part is calculated by solving the two-dimensional unsteady-state heat equation, a partial differential equation, by Crank–Nicolson finite-difference method. The heat duties of the part surfaces determined by solving the heat balance relations of the furnace are then used to define Neumann-type boundary conditions for the heat equation to compute the part temperature profile.

The optimization problem can be represented mathematically as follows:

$$\begin{aligned}
 &\text{minimize Fuel Input Part} = f(T_{sp}) \\
 &T_{sp} \\
 &\text{subject to} \\
 &1,000K \leq T_{sp,i} \leq 1,300K \\
 &T_{sp,i} + T_{diff} \leq T_{sp,i+1} \\
 &T_{part,exit} \geq 1,100K \\
 &\frac{\sigma_{part,exit}}{\mu_{part,exit}} \leq 0.05 \\
 &d_{part,exit} \leq 90\mu\text{m}
 \end{aligned} \tag{1}$$

where $f(T_{sp})$ is the energy input per part processed, $T_{sp,i}$ is the temperature set point of zone i , T_{diff} is the minimum allowed temperature difference between consecutive zones in the direction of part movement, and $T_{part,exit}$ is the minimum temperature of the part at the exit of the furnace. $\sigma_{part,exit}$ and $\mu_{part,exit}$ are the standard deviation and average exit part temperature, respectively, and $d_{part,exit}$ is the maximum grain size of the part at exit. Note that the part conditions are averaged quantities. The temperature set points must be in the region where the austenite phase is stable. Regardless of shape, size, and volume, parts must be heated past the required threshold with minimum variance in the part’s temperature distribution. Grain sizes must be within the upper bound to ensure toughness of quenched product. Larger grain sizes that make the product brittle must be avoided.

A physics-based furnace model provides the fidelity to address radiation interactions but is necessarily computationally intensive and iterative in calculating gradients and determining a temperature profile. This presents a considerable impediment for optimization because the gradients are required in the optimization solvers. In the approach presented, we have opted for using surrogate models that mimic the physics-based model in the region where the optimal solution is likely to be found. The physics-based model is used to identify surrogate model coefficients, and the surrogate models are then used

during run-time for optimizing the part temperatures and energy usage. The approach converges the physics-based model predictions together with the identification of the surrogate model parameters. It remains computationally expensive, but it produces a surrogate model for operational use. Computationally, the convergence approach requires execution of the physics-based model at each iteration, which takes about 15 minutes per iteration on a computer with an Intel central processing unit (CPU) (Intel Inc., Santa Clara, CA) of 2.6 Hz frequency and 6 gigabytes (GB) of random access memory (RAM). Identification of the surrogate model coefficients takes about twelve hours. The cloud resources, such as from AWS, charge according to the CPU type and RAM requested, multiplied by the number of hours. So, it helps to know the types of images that are required in this kind of modeling to have an approximate knowledge of cost in cloud computing. The surrogate model, when used for fast simulation and optimization, takes less than a minute to identify the optimal solution.

The particular structure of the surrogate model was chosen to capture local convection and zone-to-zone radiation effects.

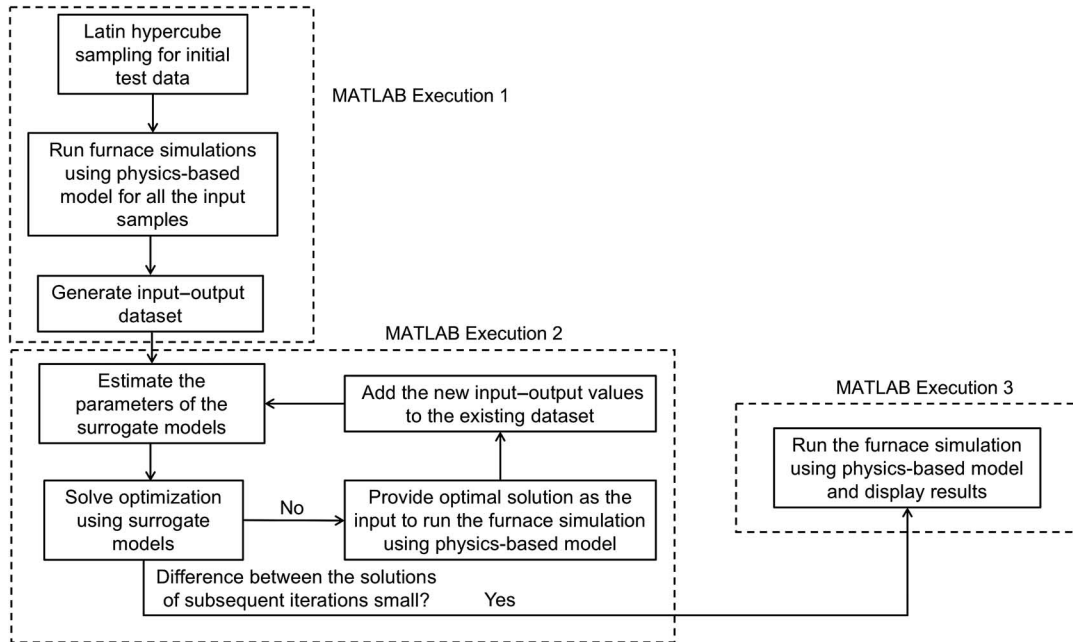
$$\begin{aligned}
 z_k = & \beta_1 T_{sp,1} + \beta_2 T_{sp,2} + \beta_3 T_{sp,3} + \beta_4 T_{sp,4} + \beta_5 T_{sp,1}^4 + \beta_6 T_{sp,2}^4 + \beta_7 T_{sp,3}^4 \\
 & + \beta_8 T_{sp,4}^4 + \beta_9 T_{sp,1}^3 + \beta_{10} T_{sp,2}^3 + \beta_{11} T_{sp,3}^3 + \beta_{12} T_{sp,4}^3 + \beta_{13} T_{sp,1} T_{sp,4} \\
 & + \beta_{14} T_{sp,2} T_{sp,4} + \beta_{15} T_{sp,3} T_{sp,4} + \beta_{16} T_{sp,1} T_{sp,3} + \beta_{17} T_{sp,2} T_{sp,3} + \beta_{18} T_{sp,1} T_{sp,2} \quad (2)
 \end{aligned}$$

where z_k is the k th output variable, $T_{sp,i}$ is the temperature set point of zone i , and $\beta_1 - \beta_{18}$ are model coefficients, which are estimated by an iterative procedure described next.

Fig. 2 shows the logical steps involved in estimating the surrogate model coefficients and finding the optimal operational settings using the surrogate models. We first use a Latin hypercube sample [19,20] to generate 100 sample points of the 4 zone temperature set points. Latin hypercube sampling spreads the sample points more evenly across all possible values compared with uniform random sampling for a fixed (usually small) number of sample points N . It partitions the input space into N intervals of equal probability and selects one sample point from each grid. The detailed furnace model is then run to calculate the averaged output quantities: minimum part temperature, average part temperature, standard deviation of part temperature distribution, maximum grain size at the exit of the furnace, and energy input to the system per part processed. Note that the form of the surrogate model for each of the output variables remains the same (Eq 2) but the coefficients are different. Using the input–output data, we identify the surrogate model coefficients for each of the output variables using least squares regression based on the temperature data. Then, we interactively solve for the optimum operational settings (Eq 1) using the surrogate models to produce a new set of input and output values. In **Fig. 2**, this is the data generation task that is called MATLAB Execution 1.

MATLAB Execution 2 is an iterative computation in which the surrogate model and the physics-based model are used to simultaneously converge on the surrogate model coefficients and the optimum operational temperature conditions. The optimum operational conditions are those that ensure the properties of the metal parts with minimum energy usage. As shown, MATLAB Execution 2 takes the input–output data from Execution 1, estimates the surrogate model coefficients, and then solves the surrogate model for the optimum operating conditions. The estimated solution is input into the detailed model, and the surrogate model coefficients are reidentified using the updated input–output data. This procedure is repeated until the difference between the solutions

FIG. 2 Flow chart of the iterative algorithm using the physics-based furnace model for simultaneous parameter estimation of surrogate models and optimization of the furnace operation.



of subsequent iterations is within a predefined tolerance. This iterative approach to the optimum solution uses the MATLAB models in combination with the interior point optimizer “IPOPT” [21] solver.

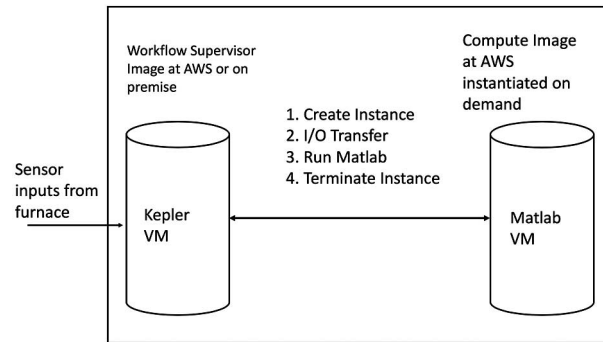
Setting Up the IT as a Cloud Computation Architecture

In constructing the workflow shown in the flow chart in Fig. 2 as an operational workflow, we provisioned the cloud IT tools and environment to make it an on-demand, autonomous computational workflow using operational data from the austenitization furnace. It is currently set up to be manually triggered by the operator but is architected in such a way that it could be triggered by an event, change in conditions, etc. The code details of the workflow are discussed in “Joining OT and IT with Kepler Workflows.”

The advantage of using cloud infrastructure is that dedicated infrastructure does not need to be built and supported for the computationally intensive furnace simulations and updating of the surrogate models. Also, a cloud-based automated workflow makes it possible for users to access complex models and infrastructure in a matter of minutes; here, AWS [10] have been used. Additionally, users will save considerable time because they don’t have to manually set up and support the compute environment each time they want to run calculations. At the University of California Los Angeles, even after operating the cluster for the last 16 years or so, we still take an average of 3 months of time to procure hardware, install software packages, etc., in order to deploy them in production. This time frame includes the time required to receive competitive quotes from vendors

FIG. 3

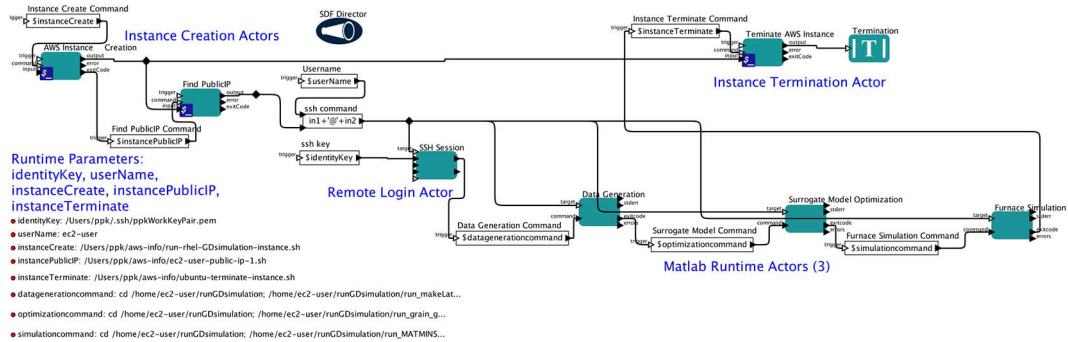
Compute architecture for running the austenitization furnace Kepler workflow at AWS.



and the ordering of replacement parts for those that arrive damaged or inoperable. In this section, we provide an architectural overview of how the computational workflow was implemented as a cloud service to convey what is embedded and hidden from the user.

With reference to [Fig. 3](#), the overall workflow management architecture is composed of a “workflow supervisor” and the surrogate/physics MATLAB execution workflow, as discussed. As cloud services, these are two distinct workflow resources, both of which exist as VM images that can be run on demand as application instances. The “Kepler VM” acts as the workflow supervisor, running Linux as a VM image. This supervisor can be run on a cloud service or locally on an on-premise computer. The Kepler VM oversees a second compute resource, shown as “MATLAB VM,” which runs Red Hat Linux (Red Hat Inc., Raleigh, NC) images that underpin a pre-installed MATLAB runtime library and other development tools. The MATLAB models described in “The OT” are encoded in MATLAB and instantiated as individual MATLAB Red Hat Enterprise Linux (RHEL) configurations. These RHEL resources constitute the on-demand modeling resources and are run only when called upon. For this use case, the size of the RHEL runtime image, including operating systems, is around 5 GB, of which around 3 GB is the MATLAB runtime library. We have given the size information of the image here because cloud service vendors charge according to the size of the image multiplied by the number of hours in usage. So, it is better to optimize the size to minimal, and small size images take less time to get deployed. The supervisor portion is constructed by installing AWS command line python interface [\[22\]](#) application tools using the commands given at the AWS website [\[22\]](#).

The cloud compute environment follows strict security guidelines recommended by the provider. In this case, using AWS, the details are available at the AWS site [\[22\]](#). This two-part architecture enables the Kepler supervisor node to execute any AWS instructions, such as “start the instance” or “terminate the instance,” without having to input login credentials during the execution. This embedded security is an important requirement for a workflow that runs without human intervention. The supervisor node, where the workflow starts, retains all the cloud login credentials in a secure location so that the on-demand computational workflow knows where to access them. The computational workflow also knows how to bypass or answer any interactive queries where human interactions are typically required. The computation workflow is set up to run from different supervisor nodes and can be started based on different triggers.

FIG. 4 Workflow to run austenitization furnace MATLAB model at AWS on demand.

The supervisor VM uses a Java (Oracle Inc., Redwood Shores, CA) runtime environment to accommodate the Kepler scientific workflow software, which is written in Java. As shown in Fig. 3, the supervisor node performs four important functions: (1) create the “instance,” (2) transfer needed data as input or output, (3) send a Secure Shell encrypted network protocol (ssh) remote execution instruction to start the MATLAB model on a MATLAB VM on the newly created instance, and (4) terminate the instance at the end of the workflow. Because the commands are executed remotely, we needed to use a remote access protocol to send the execution commands. Windows operating system (OS) uses remote desktop protocol, and Linux OS uses ssh protocol. We used ssh instructions, the most commonly used remote execution command protocol on Linux. The workflow shown in Fig. 4 uses a graphical user interface (GUI) to display the code and to execute these functions. Kepler calls each of the main boxes “GUI actors.” Each actor is an execution packet of JAVA code that executes Linux commands inside the Kepler workflow. A detailed description of an actor is given in the next section. This particular workflow is set up so that the only optional actor is the input/output (I/O) transfer actor required when there is a need to transfer input data to the compute VM and retrieve output files back to the supervisor VM. This is an operational decision limiting optional actors, which increases autonomy and minimizes user involvement but decreases flexibility.

The transfer of input data from the factory sensors to the supervisor node can be done in two ways: (1) selected data can be periodically pushed to a database or historian in the cloud service, or (2) the supervisor node can directly pull the data from the factory historian. This article does not address those two operational situations specifically because all the required input data for running the model was available in advance.

The MATLAB compute resources are shown collectively in Fig. 3 as MATLAB VM, which, in our cases, are images established in AWS. To build this VM, one can start with a standard Linux image and install the MATLAB runtime library, which is available at the Mathworks website [6]. We have included all of the required MATLAB files inside the Linux image in advance. The entire MATLAB furnace model code is precompiled using the MATLAB compiler “mcc” and converted into a runtime execution file and saved inside this image. Details of how to save a Linux image, recall it at runtime, and other details are available at the AWS website [22]. The workflow just recalls this specific image and executes the computation.

Joining OT and IT with Kepler Workflows

Kepler [5] is an open-source scientific workflow software that is constructed from other open-source software. Specifically, its software base is Ptolemy II (The Ptolemy Project, UC Berkeley, Berkeley, CA) [23], an open-source software in Java. As mentioned, Kepler is designed for actor-oriented workflow construction. Actors are software components that execute concurrently and communicate messages with each other through interconnected ports. Kepler inherits modeling and design capabilities from Ptolemy including the GUI, workflow execution, and scheduling capabilities. The workflow orchestration itself is done through a Kepler component called a “Director.” A Director controls the execution of the workflow, whether it is in a pre-executed sequence, such as synchronous data flow (SDF), or executed in parallel, such as process network (PN), in which one or more components run in parallel and different components communicate at the same time. The SDF director is used when only one actor is executing at a time in a single thread. The PN director is used when workflow is driven by data availability and multiple actors may be executing at the same time. To simplify the workflow construction process, Kepler provides an intuitive GUI and an execution engine to help edit and manage the workflows and their execution. In the Kepler GUI, actors are dragged and dropped onto the screen, where they can be customized, linked, and executed.

To support the computation models and processes described in “The OT” and illustrated in Fig. 2, the layered Kepler workflow, shown in Fig. 4, was constructed. As shown, the workflow orchestrates the three MATLAB executions for data generation, surrogate model optimization, and furnace simulation as sublayer in the right-hand corner. The highest-level workflow consists of “local execution” actors, which are named in Fig. 4 as “Instance Creation Actor” and “Instance Termination Actor.” The local execution actors are constructed from the same Kepler actor construct, but each executes a different command. This is an example of how a single actor can do multiple functions in Kepler. Fig. 4 also shows a “Remote Login Actor” to establish a remote ssh connection to three MATLAB Runtime Actors. These three actors are also the same Kepler actor construct but executing three different MATLAB commands, shown in the diagram as “MATLAB Runtime Actors.” Finally, the Instance Termination Actor terminates the running instance on AWS and release all the resources back to AWS. The section shown in Fig. 4 as “Runtime Parameters” can be altered at runtime so that we don’t need to reconstruct the workflow for usually variable parameters. This, again, shows the flexibility of Kepler workflow to reuse the actors for different actions. The three MATLAB execution actors are executed one after the other as output from one MATLAB execution task is used as input for the next. Kepler waits for each process to complete before it starts the next execution. There are several display actors to presenting the output from each of the execution actors. We use the SDF Director in this workflow because each step is carried out sequentially. All the runtime parameters, such as “datagenerationcommand,” can be altered as runtime inputs to the workflow. This gives the flexibility to reuse the workflow when different execution commands need to be run.

The details of the instance creation commands to create the virtual instances are directly available from AWS website [22] or can be made available upon request. Similarly, the details of the MATLAB execution commands are available at the MATLAB site [6].

In general, the workflow can be very complex by adding Input/Output (I/O) actors for input/output transactions between the supervisor node and MATLAB instances. This workflow can also be reused as a template to run other applications such as Mathematica

(Wolfram Research Inc., Champaign, IL) or R (The R Foundation for Statistical Computing, Vienna, Austria) by changing only the MATLAB commands with Mathematica or R commands that point to installed images of Mathematica or R.

Results from the Modeling Calculation

The on-demand cloud workflow provides the same information to a furnace operator as an application implemented in an on-premise structure. In this section, we show some example results to illustrate. In this cloud service implementation, when the workflow completes, the operator can display the results as plain text as well as graphical plots.

For this application, the heated metal parts exiting the furnace must meet the target austenitization specifications. Fig. 5 shows an example graph of the predicted exit conditions for a batch of 40 parts processed sequentially in the furnace. The parts have nonhomogeneous spatial temperature distributions because of the radiative and convective heat transfer effects between the part surface and furnace core and the conductive heat transfer within the parts. Line 4 represents the minimum part temperature at the exit of the furnace, Line 2 represents the average part temperature, and Lines 1 and 3 represent the standard deviation in part temperature. The energy input to the system for processing a part is shown by the vertical bars. The minimum part temperature must be above the threshold to ensure complete transformation to the austenite phase, and the standard deviation should be small enough to ensure part temperature uniformity. This particular chart is used by the operators to reduce total enthalpy while maintaining minimum part temperature.

As a second example of operator oriented results is shown in Fig. 6, in which the grain size distribution is plotted as a function of time. Steel is a polycrystalline material composed of many crystallites or grains. The grain size distribution in the austenite phase determines the metallurgical properties of the quenched product. Larger grains make the product brittle, making it unsuitable for the intended application. Fig. 6 shows the grain size distribution as a function of time for Part Number 20, a part in the middle of the batch.

These example plots provide information to the operator to manage the part quality along with energy input to the system. Behind these plots, from a computational

FIG. 5 Enthalpy change in the system and part temperature distribution at the exit of the furnace.

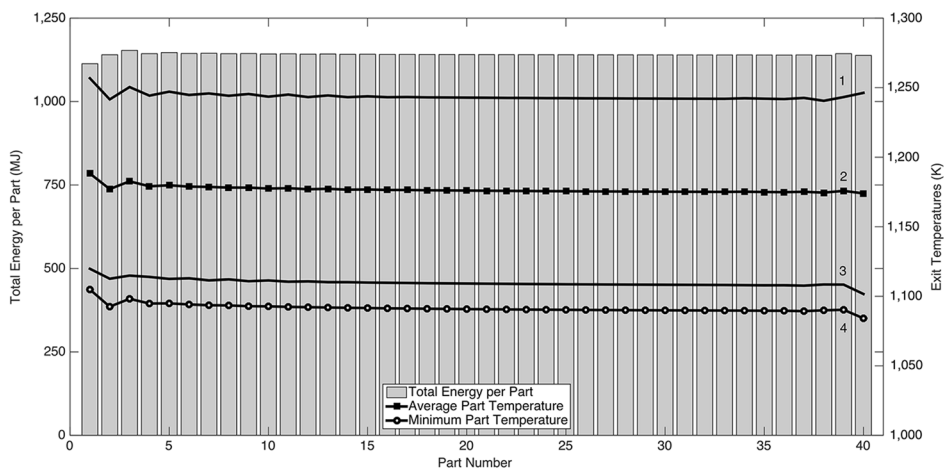
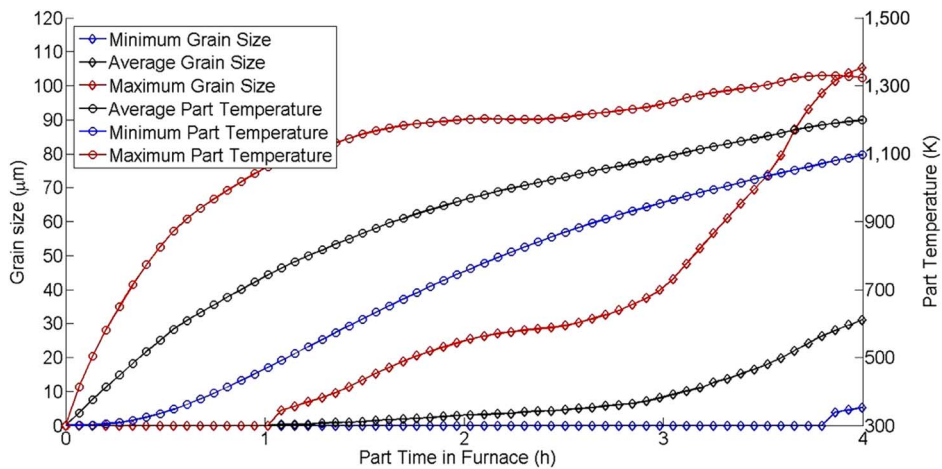


FIG. 6 Plot of grain size distribution as a function of process time in furnace.



standpoint, are simulations that take about 15 minutes to predict the energy consumption of the system and the part exit conditions for 24 hours of plant operation. When tied to the real-time measurements, these predictions can be used to solve an on-line optimization problem to identify the optimal operating condition of the furnace that minimizes energy consumption without compromising product quality. Moreover, about 10–15 % of the parts fail in the actual system because of insufficient heating and other microstructural defects, which is a direct loss of the part and the fuel consumed to make it. The availability of operational model predictions assists the operators in taking corrective actions to reduce the part failure rate.

The operational model can then be used for optimization (see “The OT”), control, and heat integration of the furnace to minimize energy consumption. We did these calculations in the cloud for fewer number of iterations than those required for accurate results for demonstration purposes to save computation time in the cloud and thus, cost. Nevertheless, the full simulations were run on desktop computers, and the results were published. Heng et al. [6] calculated the optimal zone temperature set points that would minimize the energy consumption of the system without violating the product temperature and its uniformity constraints. Operating under optimal conditions resulted in energy savings of 4.8 % compared with the heuristic operating conditions suggested by the operators of the plant. Ganesh, Edgar, and Baldea [8] included the constraint on metallurgical property (toughness), a value comparable with that in the heuristic case, along with the temperature constraints in the optimization problem. The solution resulted in energy reduction of 3.5 % compared with the heuristic case. Ganesh, Edgar, and Baldea [8] calculated the optimal set points that attempt to improve the toughness of the quenched product from the heuristic operation mode. In this case, the energy consumption increased by 12.57 %, indicating that more energy is needed to improve the product quality. Additionally, model predictive control, as a supervisory control strategy to control the product temperature, resulted in energy savings of 5.3 % compared with the heuristic case [24]. Finally, heat integration of the furnace using radiant recuperators to transfer some of the heat wastefully discharged in the burner exhaust to preheat inlet air to

the burners resulted in energy savings of 15.93 % [25,26]. The previous results have been compiled and presented in the Department of Energy (DOE) project report [27].

Limitations

On-demand cloud computing, at least in theory, should not have any limitations on the availability of the resources, but in practice, there may be delay in instantiating the images and getting up and running. Also, cloud resource vendors charge for data transfer bandwidth. So, if the model has to transfer large amount of data, the cost of computing may rise. In this example, the size of the data is small, but that may not always be the case. If the application uses licensed software packages, then the number of simultaneous instances that are running at the same time will be limited by the available licenses. The stability of this procedure is dependent on reliable network connectivity between the Kepler VM and MATLAB VM. Because we don't checkpoint our calculations in this example, in the event of network disruption, the workflow will need to be restarted.

Conclusions and Ongoing Work

The main purpose of this article is to demonstrate how the manufacturing industry can take advantage of on-demand cloud computing technologies through a scientific workflow service architecture. In this demonstration case, cloud technologies are used to run multiple austenitization furnace models without having to deploy a high-performance computing cluster in their local environment. Additionally, on-demand cloud technologies are used to run application software in an operational setting only when needed, as determined by the operators. We demonstrated the capability to view the computational results through visual plots within decision-making time frames to reduce fuel usage without compromising part target specifications. This operational demonstration also brings out how on-demand cloud computing can lower cost and improve security, not only by reducing infrastructure investment, but also by starting and stopping instances. The complete operational optimization calculations of the furnace were run in desktop computers. The operation under optimized conditions results in significant energy savings (up to 16 %) compared with the heuristic case.

With respect to the next steps with cloud technologies, we are in the process of deploying the same workflows using Docker Containers (Docker Inc., San Francisco, CA) [28,29] in a cloud vendor agnostic environment.

ACKNOWLEDGMENTS

We acknowledge the DOE grant DE-EE0005763 "Industrial Scale Demonstration of Smart Manufacturing Achieving Transformational Energy Productivity Gains." We also acknowledge an education credit for computing time on AWS resources by Amazon Web Services.

References

- [1] Davis, J., Edgar, T., Graybill, R., Korambath, P., Schott, B., Swink, D., Wang, J., and Wetzel, J., "Smart Manufacturing," *Ann. Rev. Chem. Biomol. Eng.*, Vol. 6, 2015, pp. 141–160, <https://doi.org/10.1146/annurev-chembioeng-061114-123255>

- [2] Edgar, T. F. and Pistikopoulos, E. N., "Smart Manufacturing and Energy Systems," *Comput. Chem. Eng.*, Vol. 114, 2018, pp. 130–144, <https://doi.org/10.1016/j.compchemeng.2017.10.027>
- [3] Korambath, P., Wang, J., Kumar, A., Hochstein, L., Schott, B., Graybill, R., Baldea, M., and Davis, J., "Deploying Kepler Workflows as Services on a Cloud Infrastructure for Smart Manufacturing," *Procedia Comput. Sci.*, Vol. 29, 2014, pp. 2254–2259, <https://doi.org/10.1016/j.procs.2014.05.210>
- [4] Korambath, P., Wang, J., Kumar, A., Davis, J., Graybill, R., Schott, B., and Baldea, M., "A Smart Manufacturing Use Case: Furnace Temperature Balancing in a Steam Methane Reforming Process via Kepler Workflows," *Procedia Comput. Sci.*, Vol. 80, 2016, pp. 680–689, <https://doi.org/10.1016/j.procs.2016.05.357>
- [5] Kepler, "The Kepler Project," <https://web.archive.org/web/20180531180451/https://kepler-project.org> (accessed 31 May 2018).
- [6] Heng, V. R., Ganesh, H. S., Dulaney, A. R., Kurzawski, A., Baldea, M., Ezekoye, O. A., and Edgar, T. F., "Energy-Oriented Modeling and Optimization of a Heat Treating Furnace," *J. Dyn. Syst. Meas. Control*, Vol. 139, No. 6, 2017, 13p., <https://doi.org/10.1115/1.4035460>
- [7] MathWorks, "MathWorks," <https://web.archive.org/web/20180531181019/https://www.mathworks.com/> (accessed 31 May 2018).
- [8] Ganesh, H. S., Edgar, T. F., and Baldea, M., "Modeling, Optimization, and Control of an Austenitization Furnace for Achieving Target Product Toughness and Minimization Energy Use," *J. Process Control* (in press).
- [9] Ganesh, H. S., Taleff, E. M., Edgar, T. F., and Baldea, M., "Simultaneous Optimization of Material Properties and Energy Efficiency of a Steel Quench Hardening Process," presented at the *2017 American Control Conference (ACC)*, Seattle, WA, May 24–26, 2017, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 2219–2224.
- [10] Amazon, "Amazon Web Services," <https://web.archive.org/web/20180531181205/https://aws.amazon.com/> (accessed 31 May 2018).
- [11] United States Energy Information Administration, "Monthly Energy Review," EIA, 2014, <https://web.archive.org/web/20180718160629/https://www.eia.gov/totalenergy/data/monthly/archive/00351412.pdf> (accessed 15 Dec. 2017).
- [12] Viswanathan, V. V., Davies, R. W., and Holbery, J., "Opportunity Analysis for Recovering Energy from Industrial Waste Heat and Emissions," Pacific Northwest National Laboratory, Richland, WA, 2006.
- [13] Thekdi, A. C., "Energy Efficiency Improvement Opportunities in Process Heating for the Forging Industry," presented at the *Forging Industry Energy Workshop*, Canton, OH, March 24, 2010, E3M, Inc., North Potomac, MD, 42p.
- [14] Worrell, E., Price, L., and Martin, N., "Energy Efficiency and Carbon Dioxide Emissions Reduction Opportunities in the US Iron and Steel Sector," *Energy*, Vol. 26, No. 5, 2001, pp. 513–536, [https://doi.org/10.1016/S0360-5442\(01\)00017-2](https://doi.org/10.1016/S0360-5442(01)00017-2)
- [15] Demailly, D. and Quirion, P., "European Emission Trading Scheme and Competitiveness: A Case Study on the Iron and Steel Industry," *Energy Econ.*, Vol. 30, No. 4, 2008, pp. 2009–2027, <https://doi.org/10.1016/j.eneco.2007.01.020>
- [16] Pardo, N. and Moya, J. A., "Prospective Scenarios on Energy Efficiency and CO₂ Emissions in the European Iron & Steel Industry," *Energy*, Vol. 54, 2013, pp. 113–128, <https://doi.org/10.1016/j.energy.2013.03.015>
- [17] AFC Holcroft, "Conveyor Furnace: Continuous Conveyor Thermal Treatment System," https://web.archive.org/web/20180531181453/https://afc-holcroft.com/images/pdfs/ConveyorFurnace_Brochure.pdf (accessed 31 May 2018).
- [18] Ettore, A., "Application of Mathematical Modelling to Hot Rolling and Controlled Cooling of Wire Rods and Bars," *ISIJ Int.*, Vol. 32, No. 3, 1992, pp. 440–449, <https://doi.org/10.2355/isijinternational.32.440>
- [19] Iman, R. L., Helton, J. C., and Campbell, J. W., "An Approach to Sensitivity Analysis of Computer Models: Part I—Introduction, Input Variable Selection and Preliminary Variable Assessment," *J. Qual. Technol.*, Vol. 13, No. 3, 2018, pp. 174–183, <https://doi.org/10.1080/00224065.1981.11978748>

- [20] McKay, M. D., Beckman, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245
- [21] Wächter, A. and Biegler, L. T., "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Math. Program.*, Vol. 106, No. 1, 2006, pp. 25–27, <https://doi.org/10.1007/s10107-004-0559-y>
- [22] Amazon Web Services, "Using Amazon EC2 Instances from Command Line to Launch, Connect and Terminate," <https://web.archive.org/web/20180531182041/https://docs.aws.amazon.com/cli/latest/userguide/cli-ec2-launch.html> (accessed 31 May 2018).
- [23] Ptolemy Project, "Ptolemy II," <https://web.archive.org/web/20180531181714/https://ptolemy.berkeley.edu/ptolemyII/> (accessed 31 May 2018).
- [24] Ganesh, H., Edgar, T. F., and Baldea, M., "Model Predictive Control of the Exit Part Temperature for an Austenitization Furnace," *Processes*, Vol. 4, No. 4, 2016, p. 53, <https://doi.org/10.3390/pr4040053>
- [25] Ganesh, H. S., Ezekoye, O. A., Edgar, T. F., and Baldea, M., "Improving Energy Efficiency of an Austenitization Furnace by Heat Integration and Real-Time Optimization," presented at the 2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, May 24–26, 2018, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 1–6.
- [26] Ganesh, H. S., Ezekoye, O. A., Edgar, T. F., and Baldea, M., "Heat Integration and Operational Optimization of an Austenitization Furnace Using Concentric-Tube Radiant Recuperators," *AIChE J.*, 2018, <https://doi.org/10.1002/aic.16414> (in press).
- [27] Edgar, T. F., Baldea, M., Ezekoye, O., Ganesh, H., Kumar, A., Wanegar, D., Torres, V. M., Davis, J., Christofides, P., Korambath, P., Manousiouthakis, V., Graybill, R., Schott, B., Megan, L., Flores-Cerillo, F., Hu, G., Vispute, T., Chup, J., Albertson, T., Cannizzaro, S., Schuster, D., Callahan, P., and Swink, D., "Industrial Scale Demonstration of Smart Manufacturing, Achieving Transformational Energy Productivity Gains," USDOE Office of Energy Efficiency and Renewable Energy, Washington, DC, 2018, pp. 80–83, <https://doi.org/10.2712/1454266>
- [28] Docker, "Docker Software Containerization Platform," <https://web.archive.org/web/20180531181918/https://www.docker.com> (accessed 31 May 2018).
- [29] Wikipedia, "Linux Containers," https://web.archive.org/web/20180531182337/https://en.wikipedia.org/wiki/Linux_containers (accessed 31 May 2018).