Fall 2017

Cyber-physical manufacturing cloud: An efficient method of building digital twin for 3D printer by adapting MTConnect protocol

Liwen Hu

CYBER-PHYSICAL MANUFACTURING CLOUD: AN EFFICIENT METHOD OF BUILDING DIGITAL TWIN FOR 3D PRINTER BY ADAPTING MTCONNECT PROTOCOL

by

LIWEN HU

A THESIS

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

in

MECHANICAL ENGINEERING

2017

Approved by

Ming Leu, Advisor
Frank Liou
K. Chandrashekhara

ABSTRACT

The common modeling of a virtual machine is using an information model to describe the physics of machines. The integration of digital twins into productive cyber-physical cloud manufacturing (CPCM) systems imposes strict requirements such as reducing overhead and saving resources for the systems. In this paper, we investigate a new method for building cloud-based digital twins (CBDT), which can be adapted to the CPCM platform. Our method helps reduce computing resources in the processing center and guarantees the fastest speed of the interactions between the human users and physical machines. We introduce a knowledge resource center (KRC) built on a cloud server for information intensive applications. An information model for one type of 3D printers is designed and integrated into the core of the KRC as a shared resource. Several experiments and results are provided to show the performance of the CBDT compared to traditional methods.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude towards my advisor, Dr. Ming C. Leu, for his support, advice and guidance throughout my graduate studies. I would like to thank Dr. Frank Liou and Dr. K. Chandrashekhara for spending time and effort as my committee members and helping me through the course of studies. I would like to thank Dr. Ngoc Tu Nguyen for his help in the research project.

I would like to thank Dr. Hai Tsai, who have helped me greatly during my course study, project work and research work. I would like to thank Dr. S. Balakrishnan, Donald Wunsch and Robert Roe for their help through the course of studies. I would like to thank Shuang Wu, Wenjin Tao, Ze-hao Lai, Wenbin Li, Chia-Hung Hung and Yinqi Li for their help as a friend. I would like to thank Dr. Amir Ghazanfari, Austin Sutton, Leah Mason and Devin McMillen for their help during my graduate study.

Finally, I am deeply indebted to my parents for their support, encouragement and all my friends who stood by me during the entire duration of my studies at Missouri University of Science and Technology.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Figure Page

1. INTRODUCTION

Since the traditional diagnosis approach cannot provide a comprehensive view on the system's condition during working progress of manufacturing machines, the digital twin appears as the best solution that supports technicians to get in touch with physical machines overcoming the challenge of geographical distance. To improve the efficiency of remote monitoring physical machine, there is much effort to adapt the digital twin to cyber-physical cloud manufacturing (CPCM)systems. The CPCM system is one of such smart technologies uses the cloud server as a central coordination. Since thousands of applications are required to be processed simultaneously, the erratic demands can lead to the number of requests through the cloud server increasing beyond the computing capacity ([1]), which leads to the failure of the whole system. This encourages each cloud application to reduce the occupation of computing resources as much as possible by using the shared resources to decrease the total demand and achieve a smooth collaboration with each other. Otherwise, it is a tragic overloading capacity when each application takes a singular request to a system. In this context, most of the existing research works such as published by [2, 3, 4] rely on extracting data from a physical machine by impacting physical parts such as attaching sensors on physical components to collect data, resulting in increasing unnecessary overhead and wasting energy. To overcome the above challenges, we introduce a knowledge resource center (KRC) that helps share the resources for every application inquiring information in the CPCM system.

It is worth mentioning that there exists a method called web-service based virtual machine (WBVM) that claims to be successful in remote monitoring. However, in building a virtual machine process, ceasing communication or providing discontinuous information from machines happens when the congestion of Internet traffic exists in the systems. The web-service based virtual machine system is built over the Internet based on the website platform, resulting in no guarantee of continuous data transition. In addition, each part in the traditional web-service based virtual machine method that is conducted in a closed loop cannot share the resources with other applications, and the virtual machine components corresponded to sensors one to one, which results in increasing the overhead of computation processing. These properties make the traditional web-service based virtual machine method no longer suitable within the CPCM system.

Building a virtual machine consumes multiple types of resources at the same time. It not only consumes the computing capacity when the CPU is not the only critical resource in cloud data centers, but also uses a lot of memory and network bandwidths, possibly causing high burden to the system. This is a formidable motivation for us to investigate a new method for building virtual machines within the CPCM system, named cloud based digital twin (CBDT) with multiple usage prediction and estimation methods to improve the demands of using resources efficiently. The CBDT aggregates the statuses of manufacturing resources through a virtualization interface that has the capability to store and tag each data point to make the data stream traceable. In this context, collaborative usage data from other protocols in the system is the best choice to take the advantage of a hybrid system like CPCM. In this work, our CBDT method is designed with a new information model to satisfy the requirements of building the virtual 3D printer on a CPCM system. Our method is executed during MTconnect protocol's operating process to reuse its shared resource based on the integration of protocols. The alternate resource utilization secures the reliable characterization of data in the system, thereby reducing both the load and the power consumption. We evaluate our building method through simulations on both model and real-world workloads. The obtained results show that the shared data resources with multiple reduce the total power consumptions and workloads of the server as well as guarantee the quality of the service of our digital twin.

The remaining sections of this paper are organized as follows. A summary of the related works is in Section 2 to give the reader a whole picture on the virtual machine. The concept of CPCM system is introduced in Section 3 to help the reader understand our research situation. The details of the proposed new method to construct digital twin are presented in Section 4 and evaluated in Section 5.

2. RELATED WORK

The principle of the idea of using a virtual machine system to monitor the physical machines first emerged in the early 1990s and aimed to simulate, analyze, and improve the actual machine tools on the production workshop [5, 6]. In a joint effort to satisfy the strict requirements of the modern factory systems, such as low overhead and real-time communication, several research papers were published by [7, 8] over the years to discuss the new method for building virtual machines. Witnessing the growth of local virtual machine tool systems during the 21^{st}-century, many research papers [9, 10, 11, 12, 13] proposed to provide the new monitoring methods to update the machines's working status during the manufacturing process. Unfortunately, since the virtual machine system technology was at the very first stage, these methods could only provide simple modeling and simulation functions. Following the great evolution of industry and network technology, the manufacture was over any physical borders to scale up via the Internet. In this context, the above methods were no longer suitable for machine-internet monitoring. To overcome the previous system's shortcomings, some researchers such as [14, 15, 16, 17] took the advantage of using web-service based technology to present the virtual machine over the Internet. Some web-service based virtual machine systems proposed by [2, 3, 4] achieved remote monitoring physical resources through virtual machines. The authors successfully built a virtual machine to reduce the bandwidth occupation instead of using a camera to remotely monitor the behavior of the manufacturing. In their systems, the virtual machine updated the status by using retrieved data from external sensors, which were embedded in the physical machine components as shown in Figure 2.1. However, these methods came with several unacceptable drawbacks. Firstly, such systems could never achieve the "real-time monitoring" as they claimed due to the Internet traffic. This is because in the Internet environment, no protocol or system could be guaranteed real-time data delivery. Secondly,

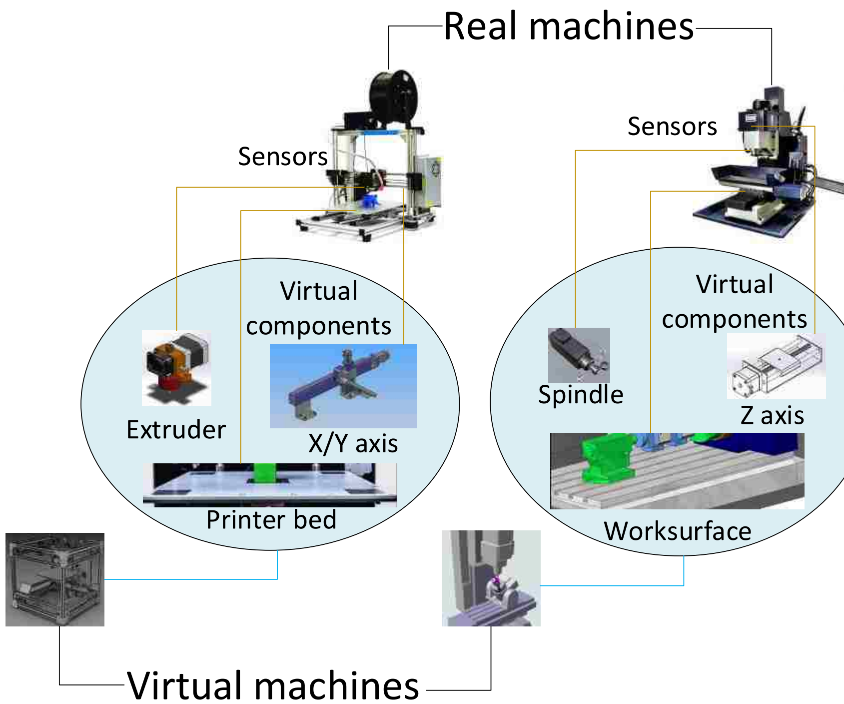


Figure 2.1. The System Architecture of web-service based Virtual Machine Constructing method.

the system is built with no intermediate intelligence center to coordinate the data streams or to manage the data from the physical components and modules; thus, every part in the virtual machine uses the fixed communication channel. Therefore, it requires tremendous effort to collect data whenever the system scales up with new types of machine. Moreover, the more external sensors that are deployed, the more extra energy is consumed, causing the unnecessary overhead by machine processing and management.

Far away from the disadvantages of previous works, the authors [18, 19] adapted the MTConnect protocol [20] to a manufacturing factory for data acquisition and tried to use the method as an alternative and effective solution to using sensors. The systems were significantly flexible when the MTconnect protocol brought efficiencies in the reducing overhead of data acquisition and guaranteed the least delay time in communications. Continuing the successes, references [21, 6, 22, 18, 19] adopted the MTconnect protocol as well. However, none of them had success in taking the virtual machine into the cloud manufacturing system. This is because there were a lot of limitations in the methods for building virtual machines that were not able to satisfy the strict requirements of cloud services, such as the requirement of continuous interactions of applications to the cloud server.

This is a formidable motivation for us to present a new method for building virtual machines, in which we combine using MTConnect to get the data with using a new information model. Our method helps reduce a lot of computing resources for the processing center as well as guarantee the fastest speed of the interactions between the users and physical machines.

Web-service based virtual machine method is one of the most popular methods for building virtual machines. The idea of using sensors as an aggregation tool to update the machine's information status for building a virtual machine was proposed early by [2, 4] and has grown in popularity over the years. As shown in Fig. 2.1, each machine has to carry external sensors on its components. The status of these components are then updated and reported by the sensors to the virtual machine building system. In this way, the virtual parts can be updated for any changes according to the reports from the embedded sensors. Finally, the virtual machine is built piecemeal during the manufacturing process in this case. In this method, the success of building a virtual machine is completely associated with the way of updating information from the sensors. Each virtual component requires a singular data stream to keep updating the status of physical components. Building the virtual machine can do very well if the system is with a few machines. However, hundreds of machine involved in a factory with the relentless requests is a common phenomenon in the modern factories. That makes the web-service based virtual machine method no longer suitable for the large-scale systems.

3. CYBER-PHYSICAL CLOUD MANUFACTURING SYSTEM CONCEPT

Cyber-physical systems (CPSs) are an integration system of computation, networking, and physical processes. CPSs are being used increasingly in manufacturing systems to improve the machine connectivity and intelligence. On the other hand, more and more domains of research are nowadays being involved with the Internet and cloud based system organization. Cloud Manufacturing (CMfg) offers services for remote management of a manufacturing process by delivering production capabilities to clients as web services. By being visualized and positioned on the cloud, it is possible to share different types of manufacturing machines for fulfilling collaborative manufacturing demands. The addition of the cloud to additive manufacturing (AM) services has bought about a new dimension of distant manufacturing production and distribution, but virtualization of machine characteristics and services is not sufficient, real-time monitoring and complete control over the Internet is still missing. With the view to achieve that, the combination of CMfg and CPS is of extreme importance. Implementing a CPCM system for 3D printers has the promise of reforming the existing manufacturing industry. To monitor and diagnose 3D printing devices through the Internet is a challenging task as there is no particular common communication protocol for network 3D printers. The scope of CPCM's application is recently widened by the developments of Industry 4.0. A CPCM system is designed to provide manufacturing services as well as allow their machining tools to be monitored and operated directly from the cloud. The potential use of CPCM in manufacturing processes also requires a specific design of system and work flow.

Figure. 3.1 shows how remote manufacturing is conducted within CPCM. A web-browser gateway for users can access the digital twins to monitoring the machine remotely during the manufacturing process. The requests of access by the users are immediately connected to the private cloud and then coordinated. The private clouds are scattered in

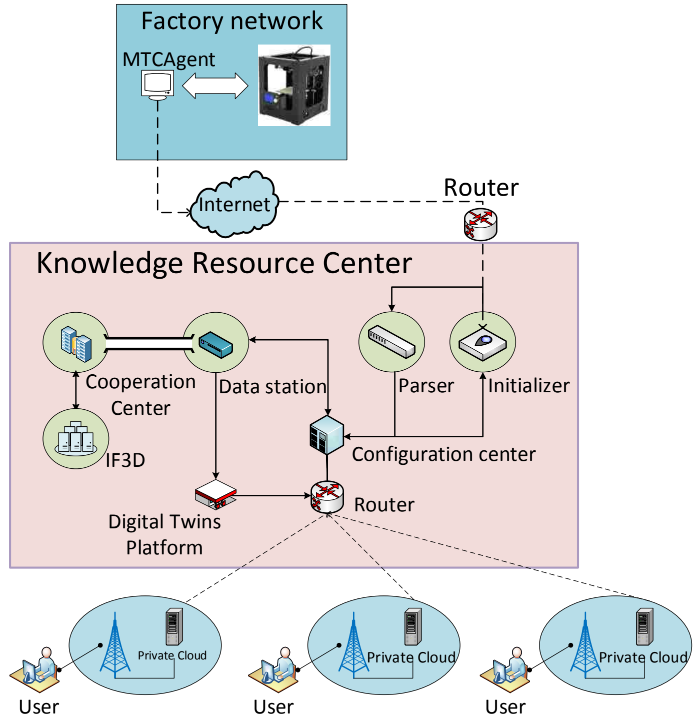


Figure 3.1. System architecture of cloud based digital twins of manufacturing machines

specific regions. Furthermore, a main station that works as a knowledge resource center (KRC) connects all private clouds together. In another aspect, the MTconnect protocol is embedded to supply the KRC with data from the physical machines. Since thousands of applications are required to be processed simultaneously, on the behalf of a processing center of a cloud server, the KRC requires strong processing in the inner core with a specific design of the relevant modules.

4. CLOUD BASED DIGITAL TWIN OF MANUFACTURING MACHINE

In this section, we first discuss the motivation and challenges of our research in Section 4.1. The details of the proposed method for constructing digital twin in the CPCM systems are introduced in Section 4.2.

4.1. THE MOTIVATION

The CPCM system is a kind of hybrid system in which multiple manufacturing factories are connected to a cloud server via the Internet and thousands of manufacturing machines are involved in the CPCM system. Every approach to the physical machines is conducted from the cloud. Thousands of services are intergraded in the inner core of cloud systems that support all activities of the applications. The relentless requests placed on the cloud server become a stretched condition. A minor flaw in a single part can trigger a sequence of cascading failure of all systems. Because of this vulnerability, it is required to minimize at most the use of bandwidth and computing resources. For these points, it cannot be conducted by the traditional web-service based virtual machine method.

When we study the building virtual 3D printer issue within a CPCM system, many new challenges are realized and compared to that in the web-service based virtual machine method. We summarize these main challenges as follows.

- **C1:** unlike that in the web-service based virtual machine methods where we can only acquire the overall information of a network step by step using a singular request for the virtual machine resulting in CPCM, we can only schedule data transmissions with limitation. It follows that thousands of applications are required to be processed simultaneously. The total demand of applications and the erratic demands can lead to the number of requests through the cloud server beyond its computing capability, which leads to the failure of the system. Therefore, how to design an effective shared data resource for the applications in the CPCM is a big challenge.

- **C2:** the communication between the layers of architecture is critical due to the diversity of the technical characteristics of each layer such as using MTConnect, Transmission Control Protocol (TPC/IP), and REpresentational State Transfer (REST) protocol, which are able to share data to each other. Thus, how to extract the data from each protocol and reuse it by others are important concerns when designing framework for the CPCM system.

- **C3:** following the challenges C1 and C2, the third challenge is how to theoretically analyze the framework and build virtual 3D printers on both model and real-world workloads. Since the data extraction manner works in a dependent manner, it is difficult, sometimes even impossible, to know the exact time a data transmission happened, as well as the time duration of a data transmission. Hence, obtaining the achievable smooth working capacity not only requires the Internet, but also depends on the carefully designed information model for physical machines.

To address these challenges, we proposed a new method to construct a new framework to build virtual 3D printer in CPCM systems, named cloud based digital twin (CBDT). We implement our proposed method through simulations on both digital models and real-world workloads. The main contributions of this paper are summerized as follows:

- We study the method of using an appropriate information model to build the digital twin for 3D printers, termed the cloud based digital twin of manufacturing machines (CBDT).

- We introduce a knowledge resource center (KRC) in order to redefine the machine's structure as well as coordinate all data streams flexible during building the digital twin process.

- Our method is executed during the MTconnect protocol process to reuse the shared resource based on the integration of the protocols.
- Our method deducts the total overhead by using shared resources from KRC to achieve a smooth performance of the digital twin system.

4.2. THE KRC FRAMEWORK

The KRC is designed with six modules, including the initializer, parser, information model, data station, digital twins platform, configuration center. All interactions between the modules in the KRC are connected and configured by the configuration center. The initializer works as a gateway of KRC to support all communications between the physical machines and the KRC. While the data in MTconnect protocol standard is organized by the XML's structure, the parser helps to retrieve the necessary data from MTconnect protocol standard. Afterward, the extracted data is forwarded to the data station, where the data is stored in the server for building digital twins. The configuration center also helps bridge the information model and data station and provide memory cells to store processed data. In the last stage, the digital twins platform is set to present the completed digital twins to the users.

As mentioned in Section 2, the traditional method uses manual connections to supply data to construct the virtual machine. These manual connections consume a lot of resources, resulting in the failure of system when building virtual machines for different types of manufacturing machines at the same time. In contrast, the CBDT uses the IF3D as a central resource in the KRC and serves as a repository and shareable resource for every input of any requests of building digital twin. These features plus functional modules allow the connection of physical machines to the KRC without any manual impacts. The CBDT is described from Section 4.2.1 to 4.2.4.

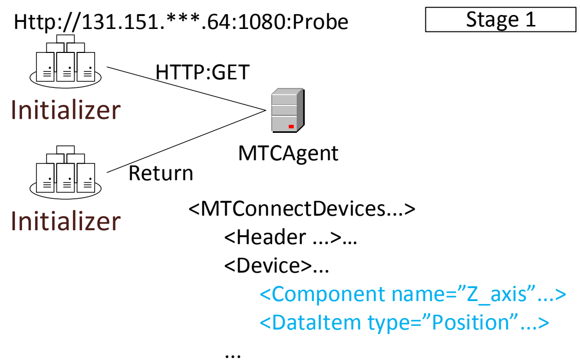


Figure 4.1. Stage 1: The probe command and MTCAgent response.

4.2.1. Initializer and MTConnect Communication. Any requests of users sent to the cloud sever are intermediately coordinated by the private cloud to forward to the configuration center in the KRC. Once receiving the requests, the configuration center activates the initializer as the beginning of the digital twins building process. The initializer works as an anchor to connect the KRC to a physical machine in the factory network through the internet. This connection process is divided into two stages.

In the first stage, because the KRC needs to create a guide model, that requires virtual components, the initializer therefore sends the requests to get the descriptive data from the physical machines to let KRC determine what virtual components should be included in the guide model. All responses from the physical machines to the KRC through the initializer are aggregated to the XML file. These requests in the first stage are defined by a specific type of command known as the *Probe* command. The *Probe* provides a mechanism for extracting the information of a machine's structure from the physical machines. This information is then used to construct the data for the responded XML file.

In the second stage, the KRC requires updated information of the physical machine's components, such as, the position, rotation, and speed to the corresponded components in the digital twins. In this stage, the initializer sends the type of *Current* command to the factory MTCAgent and inquires the current status of the components in the physical

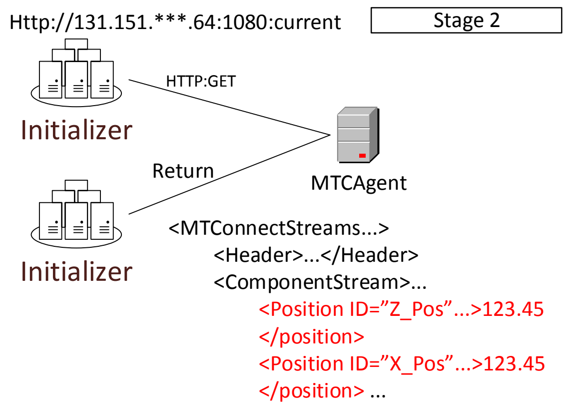


Figure 4.2. Stage 2: The current command and MTCAgent response.

machine. All communications between the initializer and the MTCAgent are conducted and transported by Hypertext Transfer Protocol (HTTP). The details of two types of command [20] uses by the initializer to communicate with the MTCAgent are presented as follows:

- *Probe* to retrieve the components' information and the data items for the system. Returns an MTConnectDevices XML document.
- *Current* to retrieve a snapshot of the data item's most recent values or the state of the device at a point in time. Returns an MTConnectStreams XML document.

When the data streams between the initializer and the MTCAgent are transported by the Hypertext Transfer Protocol (HTTP), the MTCAgent works as an HTTP server and supports both types of command, *Probe* and *Current*, and the initializer works as an anchor to connect the KRC to the physical machines and sends the requests as an HTTP client. As shown in the Figure 4.1, the initializer receives the extracted information of the machine's structure from the physical machines after sending the *Probe* command to the MTCAgent. Figure 4.2 shows the details of information received by the initializer after it sends the *Current* command to the MTCAgent to inquire the current status of the components in the physical machine.

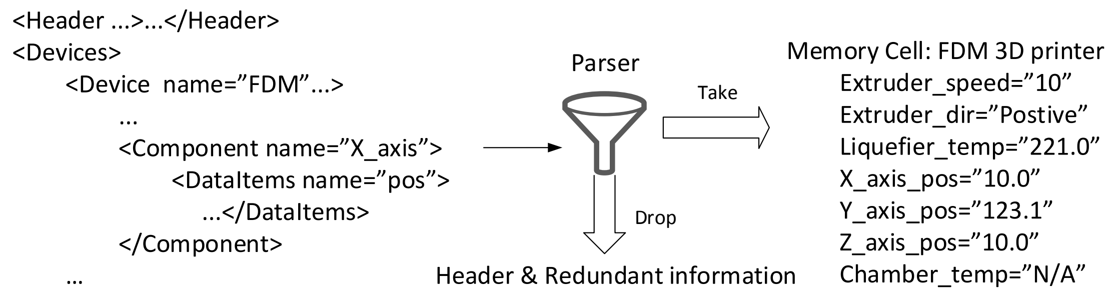


Figure 4.3. The parser filters out components value and put into the Template.

4.2.2. Parser. The data streams that get into KRC through the initializer gateway cannot be digested without processing because the data is packed by XML structure ([23]) and encrypted by HTTP protocol ([24]). A parser is considered to help extract data from the complex XML structure. Note that the data in XML is organized by architecture in the floor network level and has to be sequential traverse to approach any specific portions. In addition, those packets that are brought by HTTP are embedded protocol's scripts, encryption, and identification code. These reasons make the data stream from the initializer too complicated for direct data comprehension. To overcome these tangles, the parser provides a special mechanism for the data streams from the initializer to the data station in the KRC, shown in Figure 4.3. Each data stream is passed by a service port in the parser, and a sequential traverse mechanism is immediately constituted to hop into the data structure. On the other hand, the processing center in the parser is ready to hone the raw data provided by the service port on the symmetrical processing mechanism. The symmetrical processing mechanism is conducted based on the simultaneous two-way handling of service port and processing center in the parser. The data is popped out from service port and immediately put into and processed by the processing center. Because of this point, the KRC can guarantee the fastest processing speed even at the first stages. In the latest steps in the processing of parser, the data is quickly transferred to the data station for the next stages. All interactions of services in the KRC are coordinated by the configuration center, including the communication between the parser and the data station.

4.2.3. Data Station and Information Model. A data station is built with a set of memory cells to store the data processed by the parser in the previous stage. Likewise, it also works as a transit station to provide the data for the information model for 3D printers (IF3D). The data includes the current status of the physical machines such as the components' position, temperature, and direction, etc. These data are stored in the memory cell and controlled by the configuration center. These data sources form parts of input for any digital twins and it determines all the parameters of the digital twins' components. By collaboration with the IF3D, the data station creates a closed working loop in which the IF3D provides a set of standard information models for digital twins and the data station supplies the needed parameters of the physical machines' condition. As with the process of the parser, the collaboration is conducted on the symmetrical processing mechanism. Thus, a smooth work sequence is reached to prepare all necessary materials for building digital twins.

The IF3D is a center of the specific information model, which describes the 3D printer's structure by our CBDT method. There has been a lot of effort aimed at building a virtual CNC machine using an information model [16, 2, 25]. However, no previous research can provide a solution of using the information model to build the virtual 3D printer within the CPCM systems. Even though some additive manufacturing machines (3D printers) have similar structure, such as X/Y/Z axis with CNC machine tools, the way 3D printers handling materials and work piece is different from CNC machines. In addition, the CNC machine is constructed by discrete parts that work independently with sparse modules and are responsible for independent functionality. The machine's functionality in the virtual CNC machine therefore can be built by using the traditional information model with the simple description of mechanical elements. In contrast, the 3D printer is a complicated

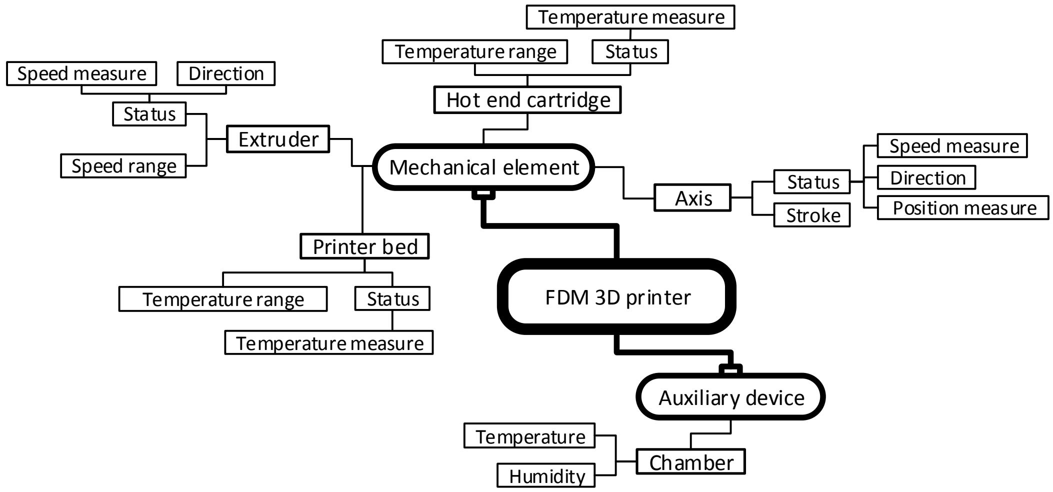


Figure 4.4. Information model for FDM 3D printer.

structure that requires a relevant information model to describe the machine's functionality. In the 3D printer, the small elements in the bottom level combine closely to each other to create the functionable structure. For example, a laser melting effect requires an LED to generate a laser beam and a galvanometer to adjust beam positions and multiple lens to change focus, beam shape, etc.

In this case, the traditional method for building a virtual CNC machine using an information model is no longer suitable to provide descriptive data for constructing a virtual 3D printer. In this paper, we propose an information model for building a virtual 3D printer. In our proposed information model, we not only construct the mechanical elements for the physical components but also define the auxiliary devices to describe the rest of the auxiliary parts with dependent functions such as the laser, chamber, and electron beam.

Figure 4.4 shows the mechanical element and the auxiliary device of FDM 3D printer with the details of their parameters. The mechanical element of FDM 3D printer includes extruder, axis, hot end cartridge, and printer bed. The modeling of axis is described by a set of variables, including stroke and status which has parameters of direction, speed

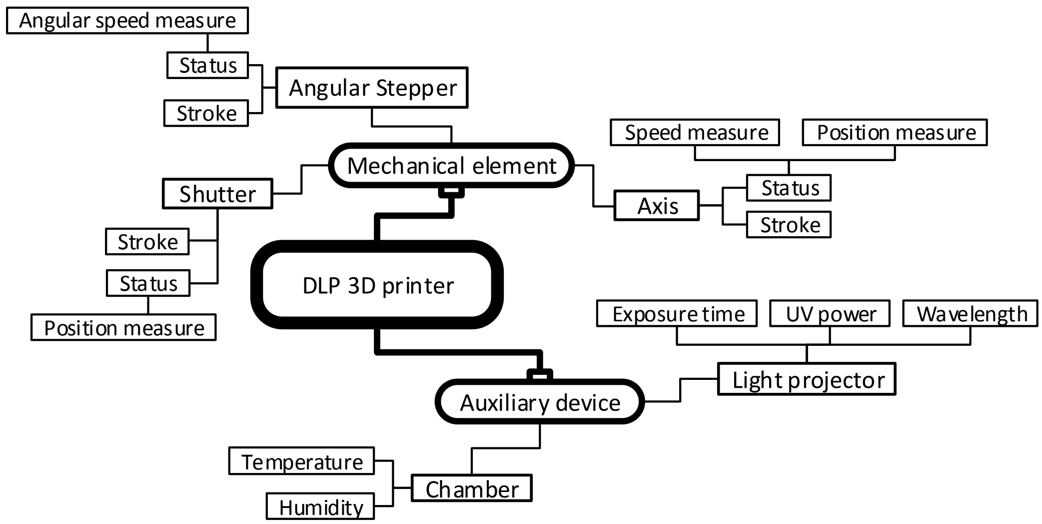


Figure 4.5. Information model for DLP 3D printer.

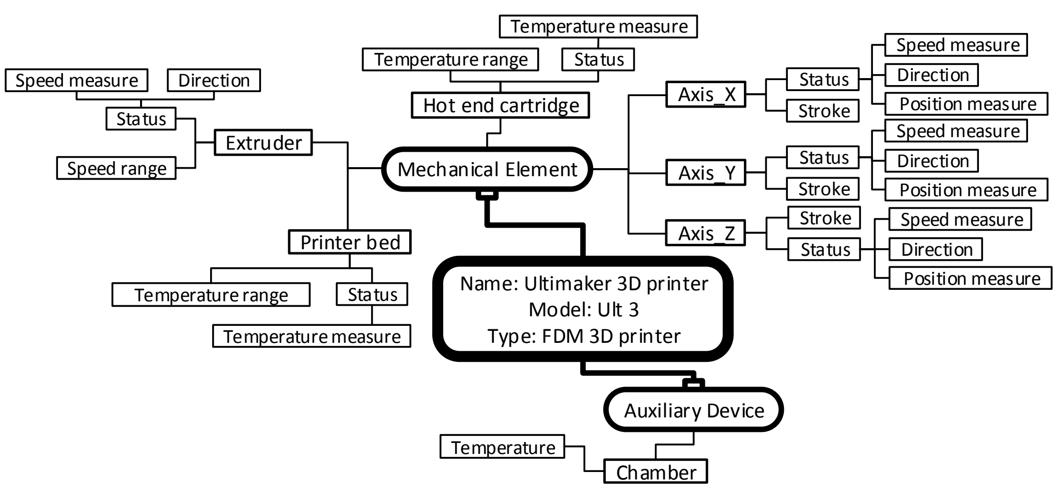


Figure 4.6. An example of Ultimaker 3D printer IF3D model.

measure, and position measure. The stroke indicates the minimum and maximum range of the axis, and the status describes the current position, speed and direction of the axis. Each component in the auxiliary device is constituted by subcomponents that describe the details of physical parts according to their function. As shown in the FDM 3D printer's information model, there are parts working together to illustrate the chamber's temperature and humidity and constitute the auxiliary device description. Through the auxiliary device description, the configuration parameters of humidity and temperature of the FDM 3D printer are updated and controlled.

When different types of 3D printers are designed with different structures to satisfy the diversity of tasks, its information models require the specific modelings. Figure 4.5 shows the information model for DLP 3D printer. In this 3D printer, the angular stepper, axis and shutter are classified as the mechanical element, which is responsible for simple function and able to work independent. Likewise, the chamber and light projector constituted the auxiliary device, in which the chamber is modeled by the cooperation of the temperature and humidity, and the light projector consists the exposure time, UV power, and wavelength.

Figure 4.6 shows the Ultimaker-3 3D printer machine information model built by the IF3D. In this model, each component of the 3D printer has been grouped to two categories by their functional features. The mechanical elements define the discrete components that can perform independently without support of other elements such as the extruder, hot end cartridge, printer bed, and X/Y/Z axes. The auxiliary devices describe the complex effects such as the chamber in the 3D printer.

4.2.4. Configuration Center & Digital Twin Platform. The configuration center is indispensable in any smart system. The role of the configuration center is very important to coordinate hundreds of commands or interactions in the system. It is the brain of the cloud based digital twins system and helps connect all parts as well as manage the logical processing signal in the KRC. The details of work flow of the configuration center are shown at Fig 4.7. When the users' requests come from the private cloud, the configuration center

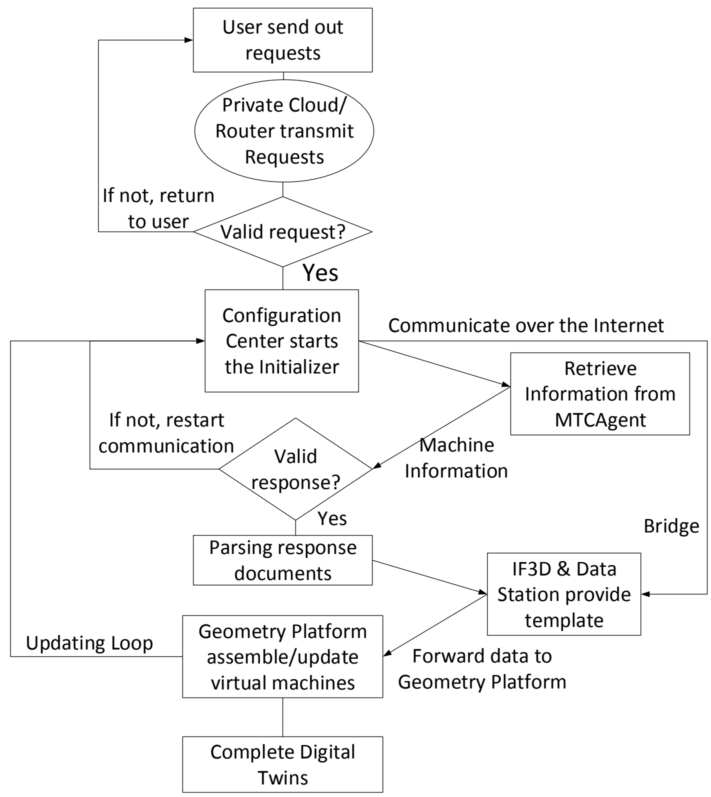


Figure 4.7. The work flow of the proposed method.

immediately actives a work sequence in the KRC. Starting from the initializer, the orders are received and the initializer then sends the communication request to the MTCAgent to inquire into the physical machines. As mentioned above, processing and exchanging data between the initializer and the parser are conducted based on the symmetrical processing mechanism where the processes operate at the same time. The key concept in this phase is that the configuration center helps synchronise the processes, time, and workloads between the initializer and the parser. In addition, the communication between the parser and the data station or the bridge between the data station with the IF3D is also coordinated by the configuration center. Thus, the configuration center is designed to provide full knowledge of connecting all portions in the KRC and to guarantee the fastest processing in the whole system. In the terminal working sequence of the KRC system, the digital twins are presented at the digital twins platform.

5. SIMULATION AND RESULTS ANALYSIS

We study the system of factory level that is located at the Intelligent Systems Center of Missouri University of Science and Technology (Missouri mini factory) and the University of Arkansas (Arkansas mini factory). We use the Hardware-In-The-Loop (HIL) scheme to evaluate the cyber-physical manufacturing system.

5.1. HARDWARE-IN-THE-LOOP (HIL) SCHEME

We built a testbed ([26]) that includes 3D printers and robots, control and communication devices, and software, in order to evaluate the feasibility of the proposed method for building digital twins. As shown in Fig. 5.1, the system in the testbed consists of two subsystems, each having 3D printer machines integrated by an industrial robot. One of the two subsystems provides the capability of hybrid additive and subtractive manufacturing, and the other subsystem provides the capability of additive manufacturing and dimensional measurement. Each subsystem has a controller for communicating with the controller of the other subsystem. The testbed will be connected to the cloud service through the IMSP protocol. Remote users will be able to place an order from the Internet and check on the progress. Such a system can be easily scaled up to consist of many subsystems, each having any number of machines.

5.2. CASE STUDY AND APPLICATION

In this section, we first discuss the processes of building the digital twins for two different types of 3D printer in terms of the scalability of the proposed method. The scenarios are designed to make the experiments closest to the real manufacturing process. Then the rest section talks about the application of digital twins. As shown in Fig 5.2, we

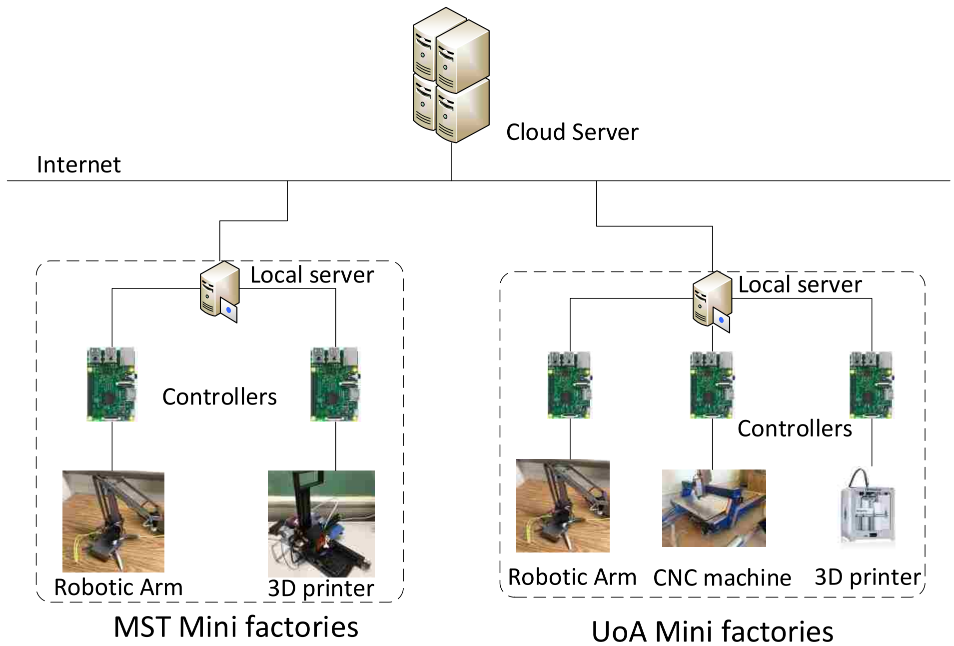


Figure 5.1. CPCM Test Bed Architecture

use Bukito 3D printer at the MST factory, while at the Arkansas factory, we use Ultimaker 3D printer. In first scenario, a user sends a request to the private cloud server, and then it is coordinated to the KRC to ask for establishing the digital twins of the Bukito 3D printer. The request is first processed by the configuration center in the KRC. After the request validation is completed, the configuration center activates the initializer to contact the MST factory's MTCAgent for the first inquiry for the Bukito 3D printer information. The initializer forwards the MTCAgent response with XML document standard to the parser under the supervision of the configuration center. Meanwhile, the configuration center notifies the data station to reserve places for the Bukito 3D printer information model that is transferred by the parser. The data including the physical machine's information are combined with the IF3D to completely describe the structure and status of the machine. At the terminal work sequence, the Digital Twin is presented at the digital twins platform. In the second scenario, the KRC received two requests come from different users coordinating by the Private Servers. One asks for constructing a digital twin for the Bukito 3D printer at the

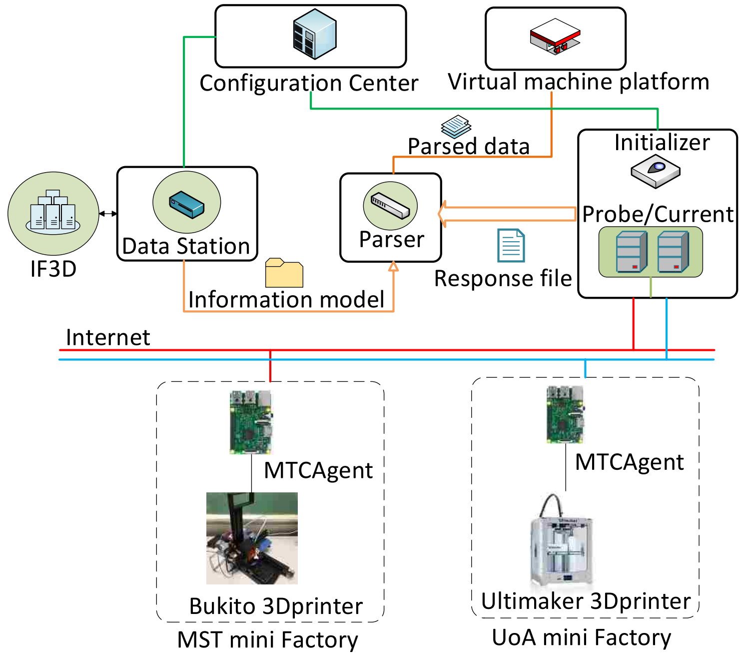


Figure 5.2. Developing 3D printer Digital Twins for MST and UoA mini factory.

Missouri factory, and another requests for constructing a digital twin for the Ultimaker 3D printer at the Arkansas factory. The requests are verified one by one by the configuration center in the KRC. Afterward, the initializer is immediately activated to communicate with factories' MTCAgents by using Probe/Current commands to retrieve information of machines. In the next steps, the initializer follows the same processes in the first scenario to forward the MTCAgent's responses to the parser. At this point, the configuration center notifies the data station to reserve two slots for the Bukito 3D printer's information model and Ultimaker 3D printer's information model. Since the Bukito 3D printer and the Ultimaker 3D printer use two different types of information models, the IF3D provides the types of model to combine with the corresponding information in the data stations.

The most important applications of building digital twins are to help connect the physical machines to the users over the Internet to remotely examine and monitor what is happening inside the machines. As shown in the Fig. 5.3, the digital twins presents the physical machine in the manufacturing process. It also provides a diagnosis panel,

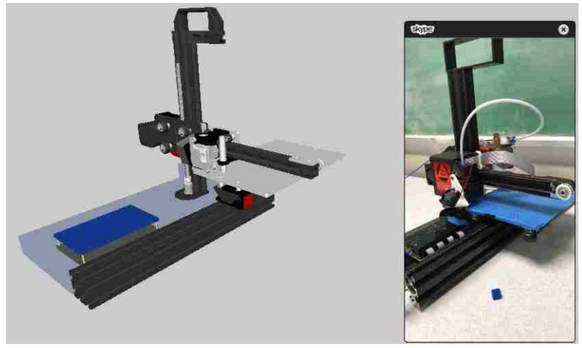
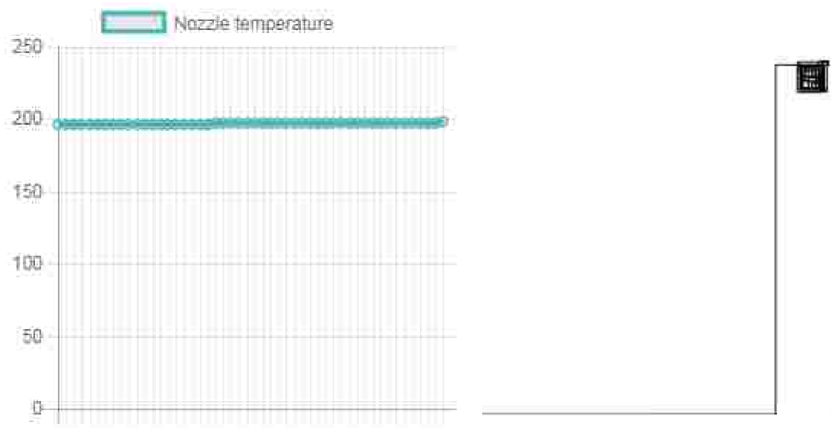


Figure 5.3. The digital twins and physical machine in the manufacturing process



Nozzle temperature Extruder path

Figure 5.4. The diagnose panel

including information of the nozzle temperature, and the extruder path. Take the Figure 5.4 for example, the diagnosis panel shows the status of the physical machine during the manufacturing process through the temperature and the extruder position information. The nozzle temperature of the physical machine is 200 degree Celsius and the extruder path indicates the position of the extruder on the X/Y plane. This tool significantly improves the diagnosis capacity of the system that helps the technician remotely monitor, diagnose and analyze multiple manufacturing machines over the Internet. By using digital twins, the manufacturers successfully reduce the human-in-the-loop, the production cost and time.

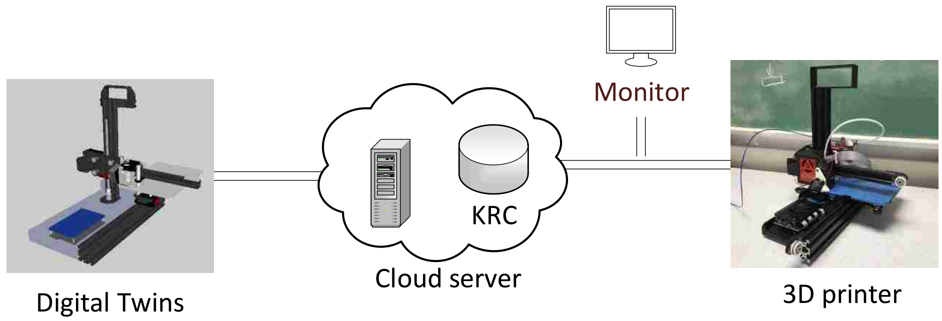


Figure 5.5. In the simulation digital twin are created for the physical 3D printer.

5.3. SIMULATION AND EVALUATION

In this section, simulations are used to evaluate the performance of the CBDT method. As shown in Figure 5.5, by using WireShark software, we can monitor the data streams in the system when the digital twins are built. The capability of the CBDT method is evaluated, analyzed, and compared with previous work, the web-service based virtual machine method.

The Bukito printer at the Missouri factory is adapted to support hybrid tasks with the MTConnect protocol. Every printing task is conducted and measured in 10 s. During the process of building digital twins, the digital twins platform sends requests to the configuration center with the frequency of 10 HZ to inquire and update the machine's status. The performance of the KRC is measured based on predefined parameters as follows:

- Delta-time: the total consuming time of the KRC to get full information over the Internet from the physical machine for building each virtual stage.
- Throughput: the total packet sizes that are delivered per time unit in the network.

For these two parameters, the delta-time is estimated whenever the physical machine changes the status and the throughput is estimated per time unit. We set up a control mechanism using the WireShark between the MTCAgent and the KRC to measure these

	B	C	D	E	F	G	H
1	Time	Source	Destination	Protocol	Length	Delt time	CumByte
2	0.14364	131.151.8.106	131.151.8.64	TCP	54	0.00006	6483
3	0.14384	131.151.8.106	131.151.8.64	Socks	217	0.00020	6700
4	0.14449	131.151.8.64	131.151.8.106	TCP	60	0.00066	6760
5	0.15914	131.151.8.64	131.151.8.106	Socks	71	0.01465	6831
6	0.16773	131.151.8.64	131.151.8.106	Socks	1514	0.00859	8345
7	0.16773	131.151.8.64	131.151.8.106	Socks	1164	0.00000	9509
8	0.16782	131.151.8.106	131.151.8.64	TCP	54	0.00009	9563
9	0.16812	131.151.8.106	131.151.8.64	TCP	54	0.00030	9617
10	0.1687	131.151.8.64	131.151.8.106	TCP	60	0.00058	9677
11	0.24282	131.151.8.106	131.151.8.64	TCP	66	0.07412	9743
12	0.24322	131.151.8.64	131.151.8.106	TCP	66	0.00040	9809
13	0.24329	131.151.8.106	131.151.8.64	TCP	54	0.00007	9863
14	0.2435	131.151.8.106	131.151.8.64	Socks	217	0.00021	10080
15	0.24417	131.151.8.64	131.151.8.106	TCP	60	0.00067	10140
16	0.28027	131.151.8.64	131.151.8.106	Socks	71	0.02195	10211
17	0.28893	131.151.8.64	131.151.8.106	Socks	1514	0.00866	11725
18	0.28894	131.151.8.64	131.151.8.106	Socks	1164	0.00000	12889
19	0.28901	131.151.8.106	131.151.8.64	TCP	54	0.00008	12943
20	0.28932	131.151.8.106	131.151.8.64	TCP	54	0.00031	12997
21	0.28968	131.151.8.64	131.151.8.106	TCP	60	0.00036	13057
22	0.34286	131.151.8.106	131.151.8.64	TCP	66	0.05318	13123
23	0.34357	131.151.8.64	131.151.8.106	TCP	66	0.00071	13189

Figure 5.6. Captured connections.

above values. Figure 5.6 shows the part of the data streams between the MTCAgent and the KRC. In every communication, the KRC works as an HTTP client with the IP address 131.151.8.106 and sends the request to the HTTP server (MTCAgent) through the initializer with the IP address 131.151.8.64. The information from line 1 to line 12 in the Figure 5.6 shows the value of the machine's parameters that is provided by the MTCAgent. The numbers in the red boarder are the consuming time during the transmission of data from the source IP address to the destination IP address, and we can estimate the delta-time of each stage by accumulating them. The number in the blue boarder shows the exchanged package size of each data transferring, and the throughput is estimated based on the information in this portion. To evaluate the effectiveness of the CBDT, a method that is extended from the web-service based technology [16, 2, 25], was proposed here in two cases, the web-sensor (2) and the web-sensor (4), to compare with the CBDT in terms of the delta-time and the throughput. In the web-sensor (2), two sensors are used to monitor the printer head z-axis

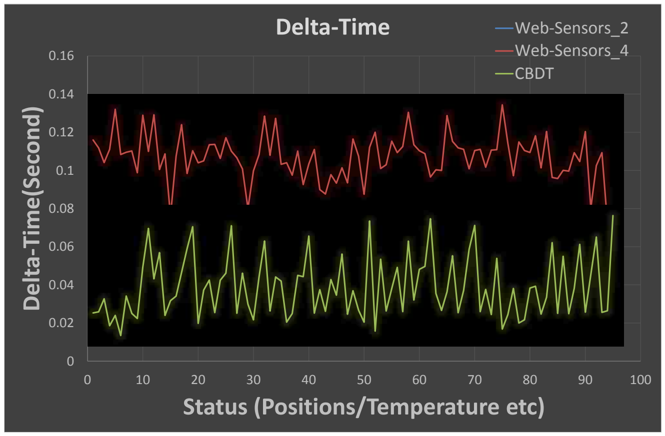


Figure 5.7. Delta-time of CBDT and web-sensor methods

position and printer bed y-axis position. The web-sensor (4) uses four sensors to monitor the physical machine. In these two cases, the data is transmitted in XML format through HTTP streaming.

Figure 5.7 shows the comparison of the CBDT, the web-sensor (2), and the web-sensor (4) in terms of the delta-time when the time running simulation is 10 s. Note that the CBDT requires the delta-time less than the web-sensor (2) and the web-sensor (4) because the CBDT uses predefined IF3D resource to build digital twin resulting in reducing a lot of data streams from the MTCAgent to the KRC. In contrast, in both the web-sensor (2) and the web-sensor (4), the status of the machine's components are updated through the singular transmissions from the embedded sensors. The higher the number of deployed sensors, the higher the delta-time required by the web-sensor (2) and the web-sensor (4) because more data streams are required to transmit from the embedded sensors.

Figure 5.8 shows the comparison of the CBDT, the Web-sensor (2), and the web-sensor (4) in terms of the throughput when the time running simulation is 10 s as well. It is clear that the CBDT provides better performance than the web-sensor (2) and the web-

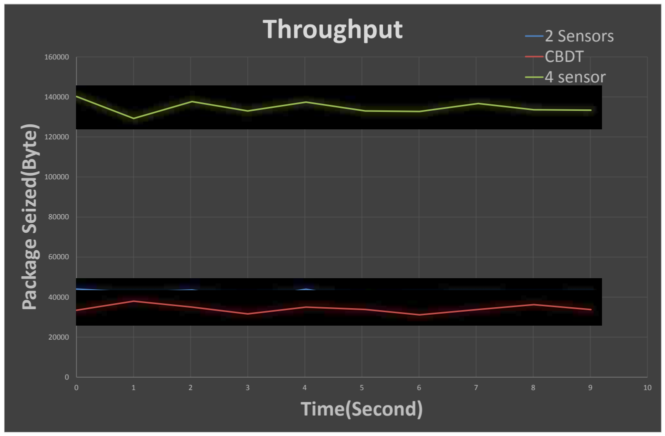


Figure 5.8. Throughput of CBDT and web-sensors

sensor (4), as the CBDT requires quite less total packet sizes. In contrast, the web-sensor (2) and the web-sensor (4) require lots of data from the MTCAgent, leading the high burden of bandwidth for the system.

6. CONCLUSION

In this work, we have investigated a method of using the information model to build digital twin of 3D printers, the cloud based digital twin. While the existing methods using information model to building digital twin have limitations in defining the machine's structure, it cannot be applied in building virtual 3D printer in general. In addition, it is a big challenge to adapt the virtual 3D printer to the cyber-physical cloud manufacturing system due to the cloud server's limitation of computing capability. In this paper, we first proposed the knowledge resource center (KRC) which helps redefine the machine's structure as well as coordinate all flexible data streams during the process of building digital twins. In the knowledge resource center, we define a new information model to illustrate the 3D printer, which works as a shared center and describes all necessary information of the machine. By taking the advantage of using shared resources given by the KRC and the data supplied by the MTConnect protocol, the cloud based digital twin successfully deducts the system's overhead and provided a smooth work of building virtual 3D printer application on cyber-physical cloud manufacturing systems.

REFERENCES

[1] X. Wang, X. Chen, C. Yuen, W. Wu, M. Zhang, and C. Zhan, "Delay-cost tradeoff for virtual machine migration in cloud data centers," Journal of Network and Computer Applications, vol. 78, pp. 62 – 72, 2017.

[2] L. Wang, P. Orban, A. Cunningham, and S. Lang, "Remote real-time {CNC} machining for web-based manufacturing," Robotics and Computer-Integrated Manufacturing, vol. 20, no. 6, pp. 563 – 571, 2004. 13th International Conference on Flexible Automation and Intelligent Manufacturing.

[3] L. Wang, B. Wong, W. Shen, and S. Lang, "A web-based collaborative workspace using java 3d," in Proceedings of the Sixth International Conference on Computer Supported Cooperative Work in Design (IEEE Cat. No.01EX472), pp. 77–82, 2001.

[4] W. Lihui, G. Robert, and R. Ihab, "Web-based implementation of virtual machine tools," in International Manufacturing Science and Engineering Conference, vol. 1, 2014.

[5] A. A. Kadir, X. Xu, and E. Hammerle, "Virtual machine tools and virtual machining-a technological review," Robotics and Computer-Integrated Manufacturing, vol. 27, no. 3, pp. 494 – 508, 2011.

[6] X. Xu, "From cloud computing to cloud manufacturing," Robotics and Computer-Integrated Manufacturing, vol. 28, no. 1, pp. 75 – 86, 2012.

[7] K. Lee and S. Noh, "Virtual manufacturing system a test-bed of engineering activities," CIRP Annals - Manufacturing Technology, vol. 46, no. 1, pp. 347 – 350, 1997.

[8] K. Iwata, M. Onosato, K. Teramoto, and S. Osaki, "A modelling and simulation architecture for virtual manufacturing systems," CIRP Annals - Manufacturing Technology, vol. 44, no. 1, pp. 399 – 402, 1995.

[9] T. Kjellberg, A. von Euler-Chelpin, M. Hedlind, M. Lundgren, G. Sivard, and D. Chen, "The machine tool model a core part of the digital factory," {CIRP} Annals - Manufacturing Technology, vol. 58, no. 1, pp. 425 – 428, 2009.

[10] C.-H. Yeung, Y. Altintas, and K. Erkorkmaz, "Virtual {CNC} system. part i. system architecture," International Journal of Machine Tools and Manufacture, vol. 46, no. 10, pp. 1107 – 1123, 2006.

[11] K. Erkorkmaz, C.-H. Yeung, and Y. Altintas, "Virtual {CNC} system. part ii. high speed contouring application," International Journal of Machine Tools and Manufacture, vol. 46, no. 10, pp. 1124 – 1138, 2006.

[12] Y. Altintas, C. Brecher, M. Weck, and S. Witt, "Virtual machine tool," CIRP Annals - Manufacturing Technology, vol. 54, no. 2, pp. 115 – 138, 2005.

[13] A. JÃűnsson, J. Wall, and G. Broman, "A virtual machine concept for real-time simulation of machine tool dynamics," International Journal of Machine Tools and Manufacture, vol. 45, no. 7 - 8, pp. 795 – 801, 2005.

[14] Y. Seo, T.-H. Choi, and S.-H. Suh, "Web-based implementation of virtual machine tools," in Proceedings KORUS 2000. The 4th Korea-Russia International Symposium On Science and Technology, vol. 3, pp. 122–127 vol. 3, 2000.

[15] S.-H. Suh, Y. Seo, S.-M. Lee, T.-H. Choi, G.-S. Jeong, and D.-Y. Kim, "Modelling and implementation of internet-based virtual machine tools," The International Journal of Advanced Manufacturing Technology, vol. 21, no. 7, pp. 516–522, 2003.

[16] H. Hanwu and W. Yueming, “Web-based virtual operating of {CNC} milling machine tools,” Computers in Industry, vol. 60, no. 9, pp. 686 – 697, 2009.

[17] S. Qin, R. Harrison, A. West, and D. Wright, “Development of a novel 3d simulation modelling system for distributed manufacturing,” Computers in Industry, vol. 54, no. 1, pp. 69 – 81, 2004.

[18] J. Michaloski, B. Lee, F. Proctor, S. Venkatesh, and S. Ly, “Quantifying the performance of mt-connect in a distributed manufacturing environment,” International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. 2, pp. 533 – 539, 2009. 29th Computers and Information in Engineering Conference, Parts A and B.

[19] B. E. Lee, J. Michaloski, F. Proctor, S. Venkatesh, and N. Bengtsson, “Mtconnect-based kaizen for machine tool processes,” International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. 3, pp. 1183 – 1190, 2010. 30th Computers and Information in Engineering Conference, Parts A and B.

[20] “MTConnect Protocol..” Available at http://www.mtconnect.org/.

[21] A. Vijayaraghavan, W. Sobel, A. Fox, D. Dornfeld, and P. Warndorf, “Improving machine tool interoperability using standardized interface protocols: Mt connect,” Laboratory for Manufacturing and Sustainability, 2008.

[22] N. Diaz, M. Helu, A. Jarvis, S. Tönissen, D. Dornfeld, and R. Schlosser, “Strategies for minimum energy operation for precision machining,” Laboratory for Manufacturing and Sustainability, 2009.

[23] “XML.” Available at https://www.w3.org/XML/.

[24] “HTTP.” Available at https://www.w3.org/Protocols/.

[25] L. Wang, R. Gao, and I. Ragai, “An integrated cyber-physical system for cloud manufacturing,” ASME Paper No. MSEC2014-4171, 2014.

[26] X. F. Liu, M. R. Shahriar, S. N. A. Sunny, M. C. Leu, and L. Hu, “Cyber-physical manufacturing cloud: Architecture, virtualization, communication, and testbed,” Journal of Manufacturing Systems, vol. 43, pp. 352 – 364, 2017. High Performance Computing and Data Analytics for Cyber Manufacturing.

VITA

The author of this thesis, Liwen Hu was born in January 29, 1993 in Sichuan, China. He received his Bachelor degree of Engineering in Mechatronics Engineering from Xihua University, Sichuan, China in July 2015. He joined the Master of Science program in Mechanical Engineering at Missouri University of Science and Technology, Rolla, Missouri in August 2015. The author received his Master of Science degree in Mechanical Engineering from Missouri University of Science and Technology in December 2017.