## ABSTRACT

基于云中间架构

SINGH, SHAURABH KUMAR. A Cloud-Based Middleware Architecture for Connecting Manufacturing Machines to Enable the Digital Thread in Product Lifecycle Management (Under the direction of Dr. Binil Starly).

机床数字线连接 产品生命周期中

The digital factory of the future will be driven by the integration of physical smart 数字工厂 智能机床 machine tools and cyber-enabled software, working seamlessly to improve manufacturing 软件 无缝工作 intelligence, flexibility, agility and production efficiency. The objective of this study is to develop and demonstrate a middleware software architecture to interface physical machines 本文研究的目的： 研究一种软件中间架构，在生产车间使用客户端与机床进行交互。 on a shop floor with client manufacturing applications. The first portion of this study was investigating methods through which we could stream data from both legacy and modern CNC machines and have its data stream to a database capable of handling high frequency data records. In-process data is made available from the machines through the MT-CONNECT communication protocol. Hardware specific adaptors were built to interface with the machine controllers to enable the transfer of machine generated data. The architecture also allows any number of third-party apps to be interfaced with the machine data for applications in machine monitoring and digital part verification. A series of manufacturing apps were built for the Digital Manufacturing Commons (DMC) through a cloud architecture provided by Amazon Web Services (AWS). In the second objective, we utilize this cloud based middleware to allow automated interaction between data provided by a product designer with manufacturing capability and capacities shared by manufacturers. The middleware built within AWS allows design and manufacturing information to be brokered without revealing sensitive information across the two parties. This research enables computing algorithms to mediate between physical machines and design services to enable the digital thread in product management.

A Cloud-Based Middleware Architecture for Connecting Manufacturing Machines to Enable the Digital Thread in Product Lifecycle Management

by
Shaurabh Kumar Singh

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Master of Science

Industrial Engineering

Raleigh, North Carolina

2017

APPROVED BY:

Dr. Binil Starly
Committee Chair

Dr. Paul Cohen

Dr. Rohan Shirwaiker

Dr. Yuan-Shin Lee

# DEDICATION

This work is dedicated to my family; my father, my mother and my sister, and my advisor Dr. Binil Starly.

## BIOGRAPHY

Shaurabh Kumar Singh was born on 23rd October 1993 in India. He completed his Bachelors in Mechanical Engineering from the Vellore Institute of Technology, Vellore, India. Shaurabh's father, who is a retired Indian Naval Officer, is also a Mechanical Engineer, was his source of inspiration. During Shaurabh's high school days, his father would often take him to his workplace and show him fancy machines. This always fascinated him which led him to pursuing his bachelor's degree in mechanical engineering. With many internships, especially the one at Mazagon Dock Limited (Ship Builders to the Nation), his interest with machines and manufacturing grew stronger. His Bachelors' final project was titled "MATLAB based Intelligent OR Systems using AHP, TOPSIS and fuzzy AHP Multi-Criteria Decision-Making (MCDM) techniques". He further decided to pursue his academics at NC State University, NC, USA. At NC State, he presently explores his long-term interests in machines and intelligent systems together in the field of smart manufacturing.

## ACKNOWLEDGMENTS

This thesis would not have been possible without the contribution and support of many people. However, I would be remiss if I did not specifically mention those who had the most significant impact on my research during the past couple of years. First of all, I would like to thank Dr. Binil Starly, my committee chair, who has been supportive of my career goals and who worked actively to provide me with the protected academic time to pursue those goals. I would also like to thank Dr. Richard Wysk and Dr. Harshad Srinivasan for their support and guidance during my first semester here at NC State.

My thesis would not have happened if it were not for the financial and technical support of Digital Manufacturing and Design Innovation Institute (DMDII) - UI Labs. Jim Barkley, Alex Maties, Marcin Welninski and all the others from Digital Manufacturing Commons (DMC) team and their support made this thesis happen in time.

I would like to thank Pulakesh Barua, Principal Engineer at Datazen Engineering, to provide me with projects and tools to help me grow my understanding of the industrial perspective. Also, I am grateful to Tim Shinbara, VP of Manufacturing at AMT, and Association for Manufacturing Technology (AMT) for providing a platform for students like me to be able to participate in a world-wide competition and present our ideas and work.

Lastly, I would like to thank my friends and family for their support. My father, mother and my sister have always supported me through thick and thin. There are too many friends to name but I would like to particularly thank Srishti, Kunal, Naman and Shirsa for their ever-lasting support. I would also like to thank all the DIME Lab members who were all there with me on the same boat. And certainly, the NC State University for making it all come true.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

The integration of digital manufacturing technology across the product lifecycle is extending its reach down to the physical machines on the production floor. This can be attributed to the advancements made in hardware and software solutions in manufacturing plants. Its understanding critically hinges on the ability to securely and easily capture, transfer, and analyze real-time streaming data from production machines to central Information Technology (IT) systems. The central systems target not only the data directly related to manufacturing, but also encompasses the entire life cycle data with respect to a product. To achieve this goal of integration across the product lifecycle, the community has come up with a vision of the 'digital thread' 数据线程，数字主线 that aims to provide an integrated view of all data pertaining to the design, manufacturing, use and disposal of the product. Its implementation will require a variety of interoperable 互操作 software and hardware adaptors to create this digital network that will span through factory floors of multiple enterprises. Since a product system touches several different technologies, it is imperative that the digital thread can only be implemented when the technology solution contains a middleware architecture which will enable an ecosystem 生态系统 of third-party applications (apps). Such a middleware architecture for the digital thread can help to rapidly deploy apps, exchange them, remove them or reactivate as necessary.

This first chapter will introduce the digital thread concept and the resulting digital 第一章介绍数字主线的概念 manufacturing paradigm. The section will describe the 'what's, the 'why's and the 'how's when it comes to digital manufacturing. In section 1.2, a related concept of cloud manufacturing is discussed which takes implementation of digital manufacturing to a next level. It is followed by section 1.3, where an open source cloud front end platform developed originally by General Electric(GE) and extended by Digital Manufacturing and Design Innovation Institute (DMDII) is discussed. Section 1.4 and Section 1.5 will elaborate on the specific problem motivation and the research objectives of this thesis.

数字化设计和制造概述

## 1.1. Overview of Digital Design and Manufacturing

Digital Design and Manufacturing is a technology-based approach to manufacturing that utilizes digital threads to link different processes of the product lifecycle. A digital thread

使用数字主线连接不同产品的声明周期

[6] is a communication framework that digitally connects the assets of manufacturing processes and provides an integrated view of these elements throughout the product manufacturing lifecycle. The interconnectivity and relay of information across the product

互联

lifecycle helps design/manufacturing engineers to make better business and manufacturing decisions with regards to product improvement and improving the existing production. It

帮助工程师为后续工作作出更好的决策 从而提高产品质量

would enable them to be able to respond to the customer and adapt to market demands in a

更敏捷的对客户和商家的需求作出反应

more agile manner. To achieve a digital thread, a hierarchy of business and technical processes are required which help integrate the data-driven decision making in the industrial world. Organizations are working towards solving the challenges of creating and installing digital threads on disparate factory floors. One of the major challenge is simply the disparity in the

统一性

software and hardware technology in the manufacturing sector. Partners in the industry range from large and accomplished digitally enabled manufacturers to the small shops which do not own a single internet compatible manufacturing machines. Some factory floors and job shops simply lack a communication infrastructure to help the direct interconnectivity of machines to the higher level IT systems. The lack of standard protocols also hinder the data interaction 缺少统一的标准协议 needed for a digital thread implementation. Many of the enabling technologies that help implement the digital thread are being developed by several technical communities, ranging from newer machine communication protocols (ex. MT-CONNECT, OPC/UA etc.), advanced process controllers, sensors, augmented reality devices and intelligent machines.

The main goal of digital design and manufacturing is to provide the industry leaders with better insight about the product at any stage from inception to final disposal to avoid significant and costly errors and at the same time gain efficiencies. Data analytics help the 能够掌握每一个零件的生产过程 避免失误 designers to better understand the factory floor challenges and how to find optimal design solutions. Manufacturers are beginning to respond [4] to opportunities presented by the digital revolution in design, fabrication, production and service. Given the vast and complex nature of digital manufacturing, experts advise manufacturers to first use the data to address specific problems. Many large manufacturers are finding value in using data analytics to optimize factory operations, boosting equipment utilization and product quality while reducing energy consumption. Such analytics can only be performed when machines connected on the shop-floor are streaming data in quantities and rates far beyond current data collection methods.

Digital manufacturing is evolved from initiatives such as computer-integrated manufacturing, design for manufacturability, lean manufacturing and others that emphasize the need for a more elaborate yet collaborative product and process design. It enables the execution of production with real-time access to shop-floor data. It further provides manufacturing companies with a platform to improve their productivity in planning and production processes. It allows part manufacturing processes to be optimized within a managed environment. One can produce flexible instructions capable of visualizing part information, along with the machining and tooling instructions. Factory models can be created faster and with and optimal layout, material flow and throughput before production ramp-up. It can be used to support six-sigma and lean initiatives by providing a graphical environment to analyze dimensional variation. Digital manufacturing systems facilitate sharing quality data across the industry by creating complete and verifiable CAD-based machine inspection programs for numerical control (NC) machine tools.

Circa 2017, digital manufacturing technologies appear to mostly refer to production of
数字化生产更倾向于单个件
discrete products – such as those produced for the aerospace, automotive, medical and
航空航天 汽车 医疗
industrial goods. These products begin from a digital model of the product built during the
工业产品
product development phase. Digital manufacturing is never used in the context of continuous manufacturing such as oil refinery, food, pharmaceutical, agriculture, textiles. The use of the
炼油厂
digital technology in the continuous manufacturing industry is present but they are mostly referred to as 'Smart Manufacturing' technologies.

The digital thread for a product must extend across multiple independent enterprises.
独立的公司
Thus, the digital space in which the digital thread of a product must exist would be in a 'cloud'
云环境
environment. Here the 'cloud' refers to compute servers and storage space maintained by external infrastructure vendors such as Amazon Web Services, Microsoft Azure, Google App Engine and many others. Cloud manufacturing then focuses on offering scalable, secure,
可扩展的 安全的
robust, high-quality and low-cost services during the manufacturing processes. In the next
强健的 高质量的 低成本的
section, different characteristics of cloud manufacturing will be discussed.

## 1.2.Overview of Cloud Manufacturing

云制造概述

Cloud computing [15] was initially introduced in the field of Information Technology sector. Rather than each individual organization maintain an in-house server infrastructure for computation and storage, the cloud offered companies the ability to offload their computing across shared servers through vendors that provide the Infrastructure as a Service (IaaS). It proved to be a disruptive innovation in the IT sector. Companies that could not afford to maintain their own computing infrastructure can now 'rent' out time on the cloud servers. As a storage mechanism, it allowed companies to share data rapidly across the enterprise since cloud server typically are maintained throughout the globe ensuring speedy delivery of content. Much of these advantages are also leverages in the realm of design and manufacturing, cloud manufacturing is gaining momentum in the academia as well as the industry.

租借云服务器

With the introduction of cloud manufacturing (CMfg), the manufacturing resources can be used as 'services' [38]. Many of the IaaS providers (Amazon, Microsoft, Google, VMware), also provide a Platform as a Service (PaaS) that allows users to design and build services to interact with the data contained in the cloud. These services can be dealt with in an integrated and an intelligent way to felicitate a network of complete sharing of the manufacturing resources. CMfg strives to provide robust, high quality, safe, economical and on-demand manufacturing services for the entire product life cycle during manufacturing.

Manufacturers are using plenty of ways to implement the cloud technology to revolutionize manufacturing. Primarily, these organizations is applying their company-intelligence using data analytics and business intelligence. Such business models are used by those who significantly rely on build-to-order and engineer-to-order strategies. The cloud-based platforms usage in digital manufacturing is growing today as idea-to-market schedule constraints are demanding greater collaboration earlier in the design cycles. Using cloud-based distributed systems to organize and manage intelligence data and rules is accelerating in the manufacturing market.

An interesting trend is the increasing reliance in the market on multiple-level resource planning strategies to gain greater efficiencies. Efficiencies in material planning, supply-chain management and logistics economy are few of the many that can be addressed. Manufacturers can also use this to achieve higher independence from a single ERP vendor occupying their entire operations. Many manufacturers who use a monolithic ERP system find it difficult to, without intensive programming and customization, scale down to the smaller operational needs globally.

Cloud manufacturing is starting to dominate the high-tech manufacturers and can prove to be a successful medium to bring the small and the medium scale manufacturers up-to the global level. Using these systems will help the manufacturers to streamline key areas of their business by freeing up more time to invest in new products and product development strategies.

Among the various cloud computing platforms available, Digital Manufacturing Commons (DMC), developed by Digital Manufacturing and Design Innovation Institute (DMDII), is used as the platform to conduct the research in this thesis. The DMC is a PaaS specifically meant to enable digital manufacturing technologies for the benefit of all organizations in the design and manufacturing space. The next section describes DMC at length, addressing its various characteristics and applications.

### 1.3. Digital Manufacturing Commons Platform by DMDII

数字化制造通用平台

The Digital Manufacturing and Design Innovation Institute (DMDII) has been developing a leading open-source internet software platform for connecting communities and sharing solutions across the manufacturing product life cycle. This open source cloud called the Digital Manufacturing Commons [46], or DMC, intends to support an online community of users, who will support the development of the platform by sharing data and models, and who will build software tools that can live in the DMC marketplace. The DMC consists of an open-source core platform, a distributed common for data and models, and a service cloud for model execution Figure 1.1. Combined, these key pieces provide a powerful software system that enables a world of innovation, collaboration, and emergent behaviors.

- Community – The DMC is a web-based collaborative environment with integrated identity management, project management and professional networking which makes it easy to connect and collaborate.
- Apps – Digital manufacturing applications can be published, shared, and linked to create new solutions faster. Users can discover them through the marketplace and execute them in public, private or hybrid cloud environments.
- Data – Secure data management features allow users to easily organize, store, and find data, all with fine-grained permissions for sharing and access control.
- Distributed Commons – The commons is distributed, so data, apps, and users can be in a variety of hybrid environments with proprietary and open source data and applications, mixed-use licensing arrangements, and public-private cloud environments.

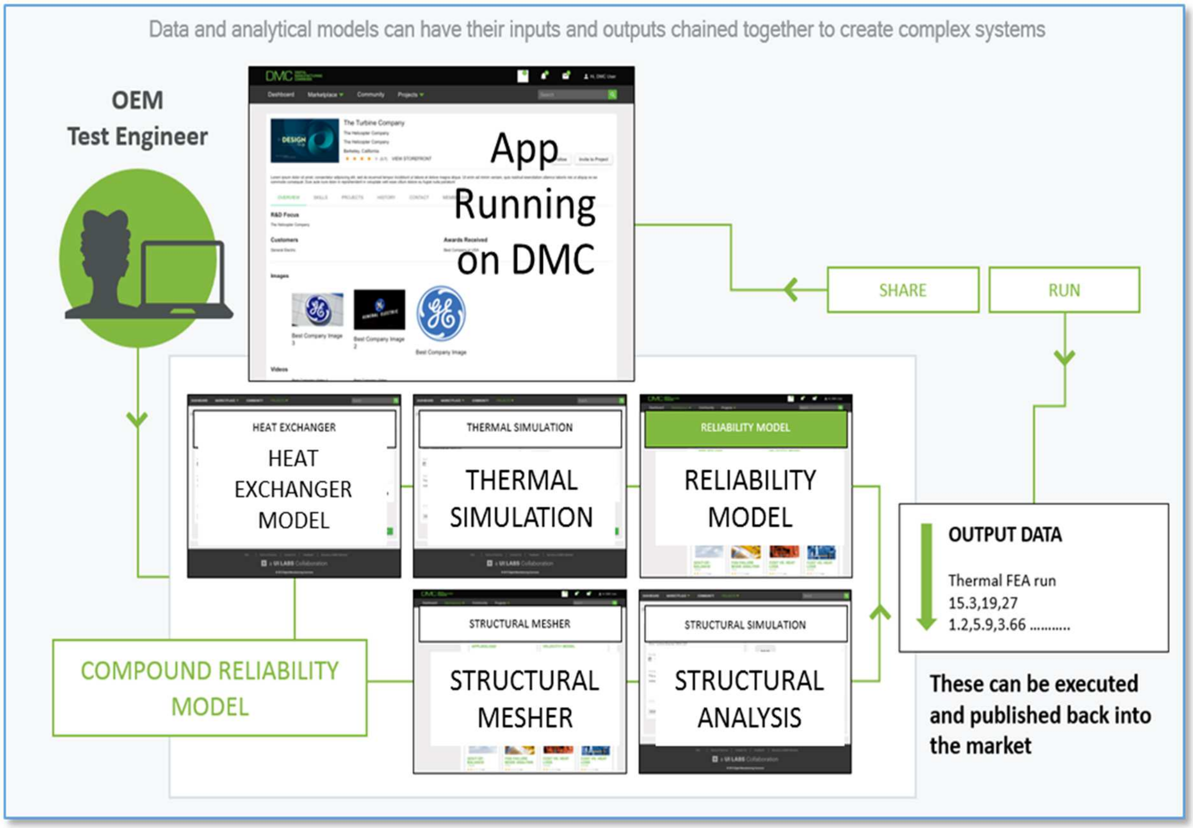



Figure 1.1: Online analytical collaboration through the Digital Manufacturing Commons. DMC allows various physical models to be linked together to perform engineering design based simulation.

The kernel that powers DMC is DOME. DOME (Distributed Object-based Modeling Environment) provides a common Internet-based computational infrastructure that makes the exchange and coordination of related parameters between different model types a transparent process. It enables a new distributed, declarative, emergent system definition process that does not require an explicit master integration model. DOME allows design teams to rapidly create integrated design models in a spontaneous, ad-hoc manner, allowing them to focus on developing and analyzing design scenarios rather than being constrained by a rigidly defined integrated design environment.

The DMC software is distributed and federated. The distributed feature allows for data,

分布式 联合式

models, and users to reside in any location that can be identified by the core platform and

分布式允许数据 模型 用户 遍布在任何一个地方

service cloud. The federated feature allows for users to securely control which users, machines, and services have access to their data and models. This allows services to broker access to other services on behalf of a user, organization, or other entity. It has a modern, services-oriented architecture which provides a high-degree of extensibility. This consists of a plugin architecture for module development, and a RESTful web services API. As an open source platform, any interested party can contribute to the codebase and participate in development activities for a specific module, or the core platform itself. At the front-end, users can design apps directly interfacing with the data stored for manufacturing intelligence applications. In Chapters 3 and 4, applications created using the DMC DOME platform are discussed giving an insight into the steps involved in design and deployment of a DOME based app.

问题方案和解决动机

## 1.4. Problem Scenario and Solution Motivation

We have discussed the digital thread concept and how it enables the next generation of the digital manufacturing technology. We have also briefly highlighted how cloud computing enables organizations to reap the benefits of the digital thread. Several challenges exist that prevent the implementation of the digital thread. This section highlights few of the problems surrounding the digital thread implementation. This will further serve to highlight the need for the specific research objectives in this thesis.

连接传统机床

Problem 1: Connecting Legacy Machines: While many modern machine tools possess sensing and control systems, the data communications and digital interfaces are frequently complex and/or proprietary. The lack of plug-and-play type digital integration is an obstacle to achieving seamless digital operation of these machines within the manufacturing enterprise. Technologies such as MT-CONNECT (MTC) allow the standardization of data streaming out of the manufacturing. While new machines may be MTC enabled, many manufacturing machines in use today simply lack the necessary adaptors to communicate with the digital network. Therefore, the first problem to tackle is can we come up with relatively cheap methods to enable machines to communicate over the network.

标准通信协议 解决不同机床数据采集问题

Problem 2: Data Storage and Retrieval: For the digital thread to be meaningful, machines must

数据存储和采集

stream its in-process data during the fabrication of the part for later retrieval and analysis. Many of the MTC enabled solutions are intended to only store discrete points of data. Time-series data, particularly when sensors are mounted on the machine can bog down traditional SQL database schemas. While proprietary plant historians exist for the continuous manufacturing space, the discrete manufacturing typically contain conventionally siloed solutions that fail to integrate across the lifecycle of the product. Any database solution must be scalable to handle 'Big Data' type characteristics when machines are streaming out lots of data that must be captured. database system which stores this machine data and presents both static as well as real-time of machine shop floor as well as company specific data. This database can then act as an efficient and secure platform for front-end users to perform different analysis.

基于云制造的数据分析

Problem 3: Enabling Information Sharing through Cloud Manufacturing: Consider a case where product designers find it difficult to search and find a potential manufacturer to make their product. Current practice of identifying suppliers through the Internet, social media, and trade shows can be time consuming and in most cases not optimal. It is significantly challenging to land the right manufacturer who has the right set of machines with the required specifications to make the product. It is not feasible for designers to share their design and manufacturing process files across the job shops in the country. Privacy and security of the data can be threatened. In addition, the same piece of information must be shared by the product designer to multiple job shop companies before the right one is selected. Therefore, to circumvent this problem, can a cloud manufacturing setup be utilized for designers to share their data with a middleware, which then extracts relevant pieces of data to compare it against services, capabilities and capacities offered by the job vendor. This middleware, which would basically be an automated algorithm 'hides' information across the two sides. How would such a middleware architecture look like, particularly through the use of a cloud computing infrastructure. The middleware architecture in a well-defined cyber-physical setup can act as a direct link between the machine back end and the customer. This could potentially eliminate the unnecessary time and resource consuming process of 'finding the one'.

### 1.5. Research Objectives

This thesis address two research objectives with respect to integrating manufacturing machines to a digital network. The first objective demonstrates how we can connect machines, specifically CNC machines to the local university network and then to the cloud. This thesis

第一个目标 如何连接 机床 CNC 在局域网 和云端

demonstrates specific apps that were created through a variety of third-party vendor applications. In the second objective, we address a case scenario through which we

第二个目标 如何通过云端 跨地域 跨车间 进行连接

demonstrate how a cloud enabled middleware architecture can help connect designers with manufacturing shops across the country. The research objectives are:

**Research Objective 1:** To develop a middleware architecture which connects manufacturing machines, such as CNC machines, to a university wide network and then made accessible globally to any authorized user. 发展一个中间架构连接机床 CNC，在世界范围内可供任意用户使用

The approach taken demonstrates several innovative use of technology. First, low cost computing platforms such as the Raspberry Pi, were utilized to network connect various types of CNC machines to the network infrastructure. Second, we show data streaming from the manufacturing machines can be collected into advanced databases built to handle time-series data. Third, we show how our middleware architecture can enable an eco-system of third-party app to broaden how the digital thread can be leveraged by a variety of engineering and non-engineering disciplines.

**Research Objective 2:** To demonstrate a cloud manufacturing framework to connect product designers with manufacturing machines while maintaining information asymmetry.

阐述一个云制造框架，连接机床，解决信息不对称问题

This objective shows a cloud computing framework leveraging various services offered through the Amazon Cloud. This app framework demonstrates a user case in which a product designer can check to see whether a manufacturing process plan is compatible with the capabilities of a manufacturing job shop. The framework demonstrates how details of the process plan are hidden from the specific job shop until a match is made. In the same line, critical information from a manufacturing job shop – such as core technical capability and capacity information is hidden from the outside world.

The two objectives eventually lead to building the foundation for a manufacturing cyberinfrastructure that can perhaps connect all manufacturing machines in a global network.

### 1.6. Thesis Outline

文章大纲

The thesis is organized as follows. Chapter 2 reviews the appropriate literature with respect to the advancements made in the fields of digital manufacturing, cloud manufacturing and the Industrial Internet. Chapter 3 discusses the first research objective. It describes the solution architecture for streaming data from legacy manufacturing machines to the cloud. It also displays various apps which are built to demonstrate machine data analytics and monitoring applications. Chapter 4 discusses the second research objective which focusses on verifying instructions from the cloud-front end to the physical machine at the back end through a secure and efficient middleware system. Finally, Chapter 5 discusses the overall results and summary of the approach. It will also highlight the technical contributions of this thesis. The chapter ends with a discussion on how this current work can be extended towards realizing a network of manufacturing machines across a national cyberinfrastructure.

## CHAPTER 2

文献概括

## LITERATURE REVIEW

The industrie 4.0 wave hit the manufacturing process technology around the beginning 工业4.0浪潮 of this decade. It increased focusses on intelligent machines, automation and data interaction between manufacturing technologies. Industry and academia have realized the importance of a digital thread connecting conceptual design of a product down to its eventual disposal. There has been plenty of research work done in the last 30 years separately on communication 数据通讯协议 protocols, databases 数据库 and cyber-physical 物理信息系统 systems. There is extensive research on the cloud-based setups in business settings heavily focused on cloud based computing for finance, technology simulations, and consumer based services. However, in the past five years, a trend has shifted towards a more integrated approach connecting these different pieces of technology.

This chapter will review the various enabling technologies that make up the approach 本章主要介绍不同的技术 连接 机床到云端 to connect machines to the cloud. Section 2.1 discusses the digital network for data extraction 数字提取中数字网络 机床 from machines. It also reviews the previous work done for data extraction 数据提取, streaming and storage 数据存储 into middleware database systems. Section 2.2 covers the current literature on cloud computing technologies and how they have influenced manufacturing systems technology. It is followed by a review of the current communication protocols in the final section 2.3. It focusses on the MTConnect industrial standard for data acquisition and briefly talks about other complementary communication architectures which enable machines to be inter-connected.

## 2.1. Digital Network on Machine Shop Floor

数字网络在车间中的应用

### 2.1.1. Digital Manufacturing: Past and Current Trends

The driving forces of the digital thread are issues of information integration and informatics across the product lifecycle management. This section presents different 数字化工厂的不同看法 viewpoints on digital manufacturing. This section describes the conventional approach and how it being transformed to a digital version.

The ever-increasing global competitiveness challenges the manufacturing market to take an integrated approach towards design and manufacturing in order to reduce the product development time. Paritala *et al.* [4], discuss how smart living is taking root. Mass production is no more the trend, but rather mass customization of products and services. This requires 大批量不再是趋势 大量定制化是如今的潮流 manufacturing bodies to evolve from a labor-intensive approach to IT enabled manufacturing setups. The conventional approach involves an in-line process where the product is designed and the designs are sent to the shop-floor for prototyping and manufacturing. Many of the manufacturers are still loaded with outdated machines which hinder their capability to compete to meet market demands. The significant motivations behind adaptation of digital technologies are time-to-market factor and optimizing costs. In recent years, digital tools like rapid manufacturing, data analytics, cloud computing and system networking are being vastly used by the companies throughout the product lifecycle.

Choi *et al.* [1], demonstrate how digital manufacturing enables simulation of the 数字仿真 product lifecycle by introducing a virtual factory system which combines procurement, production, logistics and services using digital tools for predictive analysis and solving current problems in a virtual environment.

Figure 2.1 depicts the virtual factory architecture. Choi et al [2] suggested that a top-down approach from a managerial standpoint together with a bottom-up approach to the gradual development of technologies is key to a successful implementation. The concept of the virtual factory in a top-down approach can be applied to performance factors like on-site monitoring, automation, demand-supply relation prediction and much more. Bottom-up approach would help develop the factory design which includes designing and testing for manufacturing processes, machine equipment and outputs.

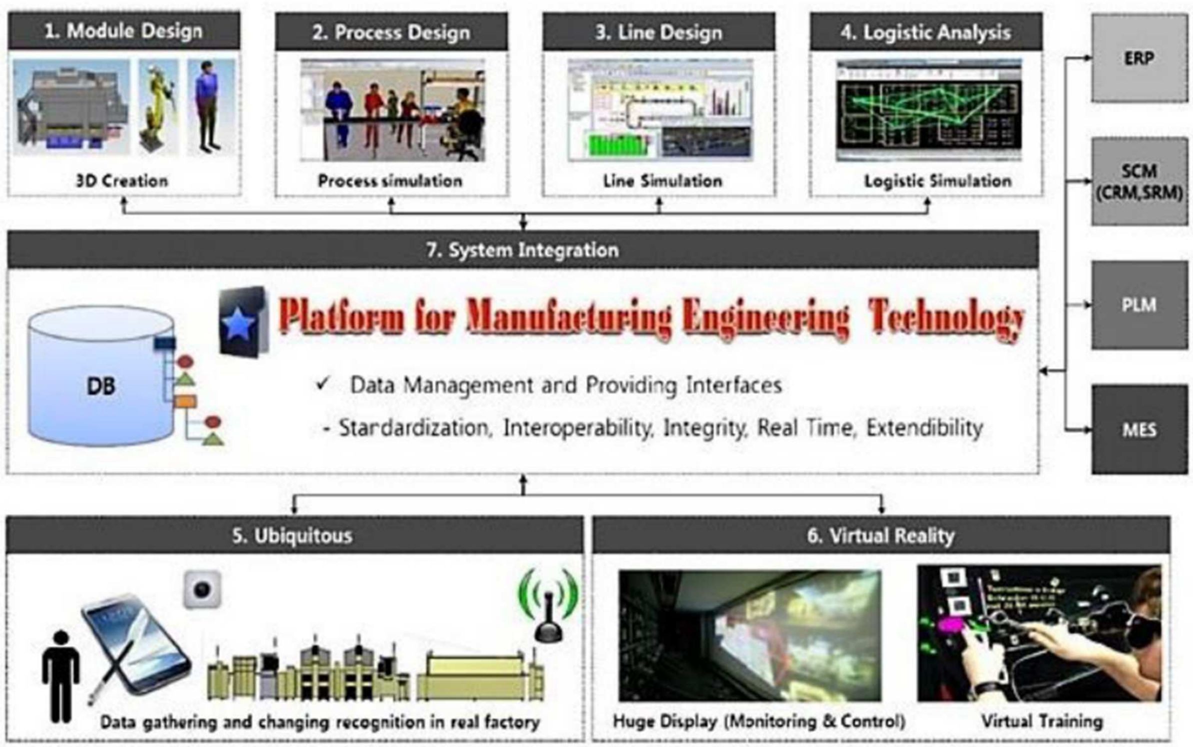



Figure 2.1: Architecture of the virtual factory (Choi et al. [2],)

Digital thread, as defined by Mark Cotteleer *et al.* [3], is a single, seamless strand of
数字主线
data that stretches from the initial design to the finished part. While the primary focus of the

article was to focus on the digital thread for additive manufacturing, the concept presented in 增材制造 Figure 2.2 is applicable to traditional manufacturing as well. An important take away from their description is the comprehensive discussion on digital thread throughout a product lifecycle.

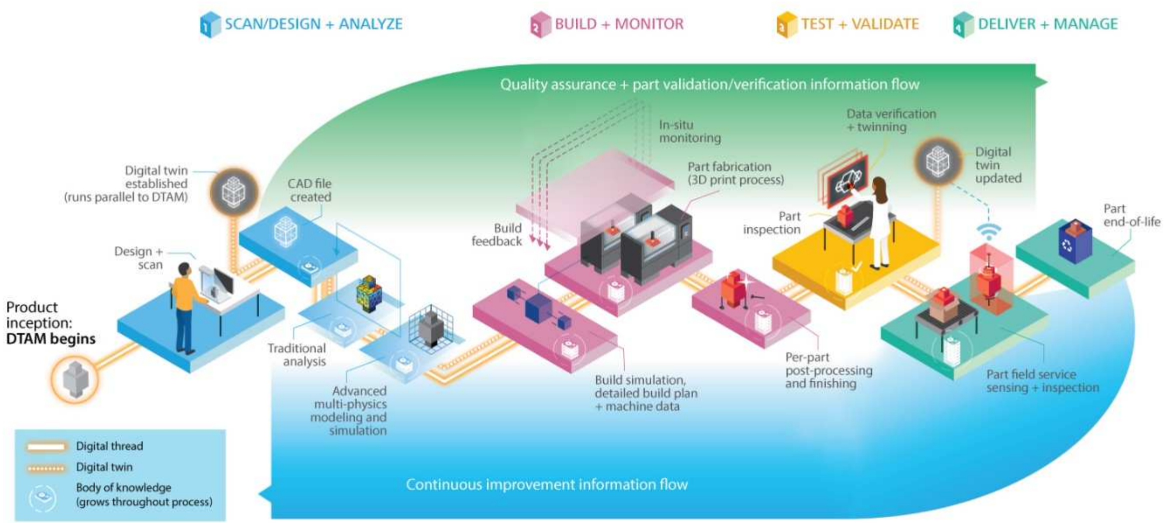



Figure 2.2: Digital Thread example in Additive Manufacturing (Mark Cotteleer et al. [3])

Figure 2.2 describes the information/data flow between different stages in a product's lifecycle. This data can be extracted, stored, communicated and analyzed at each stage for both feedback control and validation. These different types of data at each unit step can grow to large magnitudes very easily. Therefore, it is important to come up with well-defined architecture 架构 which can hold, regulate and serve data of this magnitude to users who are interested in viewing certain portions of the data. For example, a product designer who is only interested in viewing the in-process data produced during the fabrication of a critical feature on the part, must have the ability to extract such data out of the digital thread. This requires the digital thread to have some sort of data architecture that binds the different processes together.

Moneer Helu *et al.* [6], discuss a reference architecture which would integrate heterogeneous manufacturing systems across a product lifecycle for the digital thread. This

集成多样性的制造信息系统 通过 数字主线中的产品生命周期

architecture provides secluded access to internal and external clients. This helps to protect the intellectual property and confidential information, and enables the integration of manufacturing with other product lifecycle data. They have implemented their architecture with a contract manufacturer. They generated knowledge which helped them identify performance improvement opportunities which would otherwise be unobservable to a manufacturer.

Rainer Stark *et al.* [5], presents a new architecture design approach for modularized design of a Cyber Physical Production system (CPPS). They highlight the importance of integration of CPPS in existing production system architectures. The degree of capacity utilization and performance requirements of the new CPPS would indicate if functional units are duplicated or improved stations need to be developed.

This section gives a comprehensive idea about what digital manufacturing is and how

介绍什么是数字化制造以及机床是如何通过数字主线进行连接

things are connected throughout the product lifecycle explaining the concept of digital thread. These digital manufacturing architectures are enabled by the data that flows through them. The next section 2.1.2 addresses the most prominent issue of interfacing with the machine back end on the shop-floor and how this data is streamed out and utilized in downstream applications.

机床数据提取

### 2.1.2. Machine Data Extraction

This section focusses on the work done previously on the extraction of machine data for various kinds of engineering analysis. The focus of this thesis is connecting all types of machines ranging from old legacy machines incapable of data interaction to the newer robust machines. These machines will eventually communicate over the network infrastructure which will then connect to the main server that shares the stored data with external systems.

There is a significant lack of literature on data extraction from old legacy machines. There are a few articles by CISCO [39] and INTEL which talk about complete digital factory integration and how legacy devices/machines can be all interlinked with the help of Industrial Internet of Things (IIoT). Many legacy devices, especially those used in commercial and industrial applications, tend to have long life-spans. Because of which, it is not always economically feasible to upgrade a large number of embedded devices to enable them to communicate with the network. An alternative is to use gateways, which provide value by attaching to existing devices and their sensors to secure, aggregate, and filter their data. This creates opportunities to optimize the efficiency of the device, prevent failures, and create new services. The gateway needs to be intelligent and have sufficient processing power to enable end-to-end analytics that will drive business transformation.

The Cisco Industrial Ethernet 4000 Series [39] Switches offer an industrial machine 思科 connectivity solution for a secure, scalable way to connect machines to OEE platforms. The 4000 switch supports the MTConnect open standard, which is discussed in section 2.3, that's used to track machine operation, utilization, and overall efficiency. CISCO further talks about the use of "SMART BOX", combining MTConnect industrial protocol for data extraction, MEMEX's full-featured MERLIN manufacturing communications platform and the 4000 switches. The SmartBox offers network isolation, which prevents unauthorized access from both directions, that is, to or from the machines and equipment on a network.

Yet another challenge is to analyze different kinds of data being shared over a network, 分析不同种类数据 without losing any, ranging from low frequency data at 10 Hz to high frequency sensor data at 2048 Hz and higher. Jonathan Downey *et al.* [7], discuss real time monitoring of the CNC process using additional sensors in a production environment. They determined that the best machining configuration was a turning configuration, given that most vertical and horizontal machining operations utilize tools with multiple cutting surfaces. However, pointing out that turning operations on a lathe typically employ single-point cutting or two-point drilling. This indicates that cutting tool wear is less complicated in a turning configuration and is simpler to interrogate during a real-time production environment. After establishing their network and continuously being able to stream sensor data, they studied plots cutting forces versus depth of cut for a new and worn insert. They demonstrated the deployment of the sensors on a real-time production machine showing that the configuration works outside a laboratory environment. Similarly, Joao A. Duro *et al.* [8], also discuss a similar approach for CNC machine monitoring using a multi-sensor data fusion framework.

This section lists out the work being done with regards of enabling machines to share data. MT Connect Adapters, which are the most elementary halves of any "SMART BOX" like plug and play adapters, can be built to enable all kinds of machines for communication. A detailed discussion on MTConnect can be found in section 2.3. The next section 2.1.3 describes the cyber-physical architecture in the second half of this sections. It also describes the central role that databases play in terms of storage and retrieval of mostly time-series based data.

### 2.1.3. Databases in Middleware Architecture

中间件架构的数据库

The data from the machines on the shop-floor must be streamed in and archived in well-defined database systems. Practical experiences as well as literature in this field testify that the role of the database systems is becoming more and more central for control and automation in manufacturing technologies. In this thesis, the database management system used is a relational model built using PostgreSQL database system. This section presents literature on the use of relational databases and the various attributes associated with them, particularly in the context of manufacturing.

Andrea Bonci *et al.* [9], discusses a distributed database centered approach for 分布式数据库 modeling, simulation and control of cyber physical setups in their research paper. They propose a database-centric architecture that integrates networking, artificial intelligence and real-time control issues into a unified computing model. The major concept put forth here was the unifying role of the PostgreSQL database management system (DBMS) to host heterogeneous technologies for acquisition, actuation and data processing. Similar to what was discussed in the previous section, adaptors were built to special purpose computing units and machines and then connected in the PostgreSQL database framework. The DBMS must be a distributed system, particularly when hundreds of machines are reporting data. A distributed system is a network that consists of autonomous computers that are connected using a distribution middleware, PostgreSQL in this case. The distribution of the information here is carried on by a simple synchronization mechanism obtained by a very well calibrated data replication system. A distributed infrastructure is obtained when suitable mechanisms are used to propagate the updates on database tables across the distributed database management systems. This paper, in summary, proposes a viable enabling technology for the understanding of the issues in the scenario of industrial automation where the smart features of cyber-physical systems are really enabled by a capable database system.

A characteristics of utilizing databases in manufacturing environment is the need to store time-series data. In manufacturing automation, such high frequency time-series data was always stored in proprietary plant historians. However, plant historians do not themselves lend to cross-functionality unless specific adaptors were provided by the vendors themselves. Plant historians can be enabled by open-sourced databases. Anatoly Sorokin *et al.* [12], demonstrate the use of PostgreSQL database for storing time series data. With the volume of experimentally measured time series data rapidly increasing, better storage solutions must be offered than simple arrays of numbers or opaque blobs for archiving time series data. Anatoly *et al.* have equipped an access method based on SAX (Symbolic Aggregate approXimation) with the PostgreSQL database. In short, Symbolic Aggregate approXimation (SAX), invented by E. Keough, J. Lin and A. Fu, has described an algorithm that takes in input time series and transforms it into strings. They have compared this new data type with the existing methods of storing them into opaque blobs and simple arrays. This new data type has been successfully tested in a database supporting a large-scale plant and with a very large set of simulated time series data.

In this thesis, PostgreSQL relational database serves as the database management system in the middleware architecture of the cyber-physical setup at NC State ISE lab. Peng

Yue *et al.* and Wade L. Schulz *et al.* [11], talk about the different kinds of database management systems available and evaluate the relational database architectures like PostgreSQL and MySQL, and NoSQL database architectures like MongoDB. The group evaluate the databases based on their ability to manage unstructured genomic annotations. In terms of unstructured data, the NoSQL database such as MongoDB, outperforms that of relational models like PostgreSQL. Although, with structured designs, both PostgreSQL and MongoDB performed with similar efficiencies. With manufacturing data, ideally being structured following technical standards like MTConnect, higher readability and user-friendly interface of PostgreSQL makes it a good choice for a distributed database management system dealing with structured data.

The PostgreSQL database, used in this thesis, provides a platform for data interaction with the front-end users. This platform invites an ecosystem of third party applications to be built around it. D. Mourtzis *et al.* [13], have discussed in their research paper development of Mobile apps for product customization and design of manufacturing networks. The apps developed by them support two fundamental business functions. The first app enables the integration of the customer in the design phase of highly customized products. The second app enables the design of the manufacturing network to handle the production and transportation of a personalized product for a customer. Plenty other apps varying from simple machine monitoring to a fancy virtual factory can be created with the help of such a middleware platform.

This section reviewed some of the previous works with defining database managements systems in a cyber-physical setup. It further explored various types of time series storage formats in a relational DBMS and evaluated the database system used in this thesis with other systems to study their performances. A database system serves as an important platform for secondary interaction, yet there are significant limitations when it comes to using local systems for database management like scalability and data-back up to name a few. The next section 2.2. discusses literature on Cloud Manufacturing. Cloud Manufacturing addresses the limitations of having a local system and opens many gateways to improve the middleware architecture.

云制造

## 2.2. Cloud Manufacturing

Cloud manufacturing is a new manufacturing paradigm that is developed from existing advanced manufacturing models and enterprise information technologies using cloud computing, Internet of Things and advanced computing technologies. It transforms manufacturing resources into manufacturing services, which can be operated in an intelligent and unified way to enable the full sharing and regulation of these manufacturing resources. Cloud Manufacturing (CMfg) can provide safe and reliable, efficient and high quality, economical and on-demand manufacturing services for the entire product lifecycle. The concept of Cloud manufacturing was initially proposed by the research group led by Prof. Bo Hu Li and Prof. Lin Zhang [14] in China in 2010.

In this section, literature on cloud manufacturing has been discussed. In the section 2.2.1, various cloud services are discussed. It focusses on the abilities and discusses the options available in cloud computing resources. In the section 2.2.2, literature on how these cloud computing services have been used to enhance the productivity in manufacturing setups have been discussed.

云计算服务

### 2.2.1. Cloud Computing Services

Cloud computing is essentially a network of remote servers hosted on the Internet to provide services on demand, rather than a local server or a personal computer. Amazon Web Services (AWS) [37] defines it as the on-demand delivery of compute power, database storage, applications, and other IT resources through a cloud services platform via the internet with pay-as-you-go pricing. Cloud computing provides a simple way to access servers, storage, databases and a broad set of application services over the Internet. A Cloud services platform such as Amazon Web Services owns and maintains the network-connected hardware required for these application services, while the user provisions and uses what is needed via a web application.

Cloud computing has three distinct types [38] referred to as: Infrastructure as a Service
云计算三种明显的区别 架构
(IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Infrastructure as a
平台 软件
Service, abbreviated as IaaS, contains the basic building blocks for cloud IT and typically provide access to networking features, computers and data storage space. Infrastructure as a Service provides with the highest level of flexibility and management control over the IT resources and is most similar to existing IT resources that many IT departments and developers are familiar with today. Software as a Service (SaaS) provides a completed product that is run and managed by the service provider. In most cases, people referring to SaaS are referring to end-user applications. Platforms as a service (PaaS) remove the need for organizations to manage the underlying infrastructure and allows to focus on the deployment and management

of the applications. In this thesis, Amazon Web Services is used. The main cloud computing service being used is Amazon EC2 which is an Infrastructure as a Service (IaaS). It gives the most management control, flexibility and scalability over the service.

Radu F. Babiceanu and Remzi Seker [18], in their research paper, talk about big data 大数据 and virtualization of Cyber-Physical System 虚拟化物理信息系统. The recent advances in sensor and communication technologies can help define a foundation for linking the physical facility and machine world to the cyber world of Internet applications. Their work provides a review of the current status of virtualization and cloud-based services for manufacturing systems. It also throws light on the use of Big Data analytics for planning and control of business and manufacturing operations. It proposes an architecture for the development of predictive cyber-physical systems that include capabilities for attaching to the Internet of Things using cloud services like IaaS, PaaS, HaaS and SaaS and many others, and Big Data algorithmic analytics.

GE's Global Business Integration Technologies Laboratory (GBITL) goal [40] was to advance traditional manufacturing and create a dynamic network of people, machines and organizations that would promote collaboration, rapid prototyping, and product development for complex systems. By using AWS GovCloud (US), GE developed a revolutionary manufacturing platform, a "Crowd-driven Ecosystem for Evolutionary Design (CEED)", which connects people, materials, models, simulation, and equipment in an ITAR-compliant, secure, and distributed global environment. Predix, yet another platform for the Industrial Internet by GE, is by connecting industrial equipment, analyzing data, and delivering real-time insights, Predix-based apps, trying to achieve new levels of performance for both GE and non-GE assets.

Presently, the cloud is dominated by three giants: Amazon Web Services (AWS), 云技术三大巨头 Google Cloud Platform (GCP) and Microsoft's Azure. While AWS has a significant head start on the others, Google and Microsoft are quickly catching up. Madhuri *et al.* [17] and Gandhi *et al.* [16], performed comparative studies comparing Amazon Web Services and Microsoft Azure. The common conclusion from both these papers was that the AWS is preferred when it comes to efficient scalability and automation offerings, robust applications, reliability w.r.t. the feature set and the various levels of security. Microsoft's Azure is preferred especially when one is invested in Microsoft, if scalability is not a big concern or if cost is rather a very big concern. Prices of the two giants are similar with a slight edge in favor of Azure. But looking at the bigger picture, this differences are insignificant. Most of the services provided by these giants is quite comparable, but as of 2017, AWS holds the edge over the others as the "public king" of cloud.

Understanding the cloud world helps further in understanding how it can be implemented in various setups of digital manufacturing. This section focused on defining various aspects of a cloud computing services and presented some literature to highlight these services. In the next section 2.2.2, literature on how the cloud based services can be deployed in a manufacturing setup is discussed.

### 2.2.2. Cloud-Based technology in Manufacturing

云技术在制造中

Cloud Manufacturing (CMfg) [14-15] refers to a service-oriented and networked product development model where service consumers can configure services and use them to reconfigure manufacturing systems by using Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Hardware-as-a-Service (HaaS) and/or Software-as-a-Service (SaaS). There is plenty of literature of using these services to implement a cloud based manufacturing system. This section reviews some of the literature to clear the concept of CMfg and study the various ways in which it is has been implemented.

Xiaoqing F. Liu *et al.* [19], discusses a Cyber-physical manufacturing cloud (CPMC) and its scalable service-oriented architecture, a virtualization technique of the machine tools to be accessible over the Internet from cloud, various communication protocols, and testbed. It integrates the paradigms of CMfg and cyber physical systems (CPS) to enable direct operation of machining tools from a manufacturing cloud on the Internet. It has a web-based application for publishing web services for manufacturing operations. It also has an application center to publish software applications for manufacturing operations. It uses the RESTful manufacturing internet protocol MTConnect for acquiring machining tool data to monitor their operations over the Internet. The implemented CPMC testbed depicts the feasibility of monitoring machining operations and performing manufacturing operations directly from a manufacturing cloud on the Internet.

Dazhong Wu *et al.* [20], describes democratizing digital design and manufacturing using high performance cloud computing focusing on performance evaluation and benchmarking. They have introduced a new workflow to utilize high performance cloud computing (HPC) resources for accelerating compute intensive tasks in manufacturing. A set of experiments was conducted to evaluate the performance of many HPC clouds using a large-scale finite element model with more than eight million degrees of freedom. The elapsed time, speedup, scalability, and stability were used to measure the performance of the cloud computing services. The experimental results showed that the Microsoft Azure Cloud with 32 cores, and the Nimbix Cloud with 16 nodes speed up the finite element analysis over a workstation with 8 cores by more than seven-fold and eight-fold. The in-house computer speeded up the FE analysis over cloud computing by approximately two-fold because of better I/O performance and much larger memory. This paper verifies experimentally the superiority of cloud services and why the current trend is shifting towards a more stable, reliable and scalable cloud technology.

Xiuhong Chen *et al.* [24], present a case study using Community Earth System Model (CESM) on a commercial cloud computing environment, Amazon AWS. The numerical models used for simulating climate of earth are generally run on high-performance computing 天气预测 (HPC) systems. This work investigates an alternative to this approach by carrying out climate model simulations on a commercial cloud computing environment. They test the performance and reliability of running CESM on Amazon Web Service (AWS) EC2, the cloud computing environment in AWS which serves as a create virtual computing cluster. The parallelization efficiency test of the CESM model on the AWS EC2 virtual cluster is performed and they found that, up to 64 cores, the AWS EC2 can render a parallelization efficiency comparable to if not slightly better than that of the traditional Linux cluster with InfiniBand connection. Over 64 cores however, the efficiency of AWS EC2 improves significantly. By running the CESM on the AWS EC2 it was found that a highspeed network is key for the scalability of the parallelized climate models.

Yongkui Liu *et al.* [21], focus on workload-based multi-task scheduling in cloud manufacturing in their research paper. The centralized management and operation of manufacturing resources and services enable CMfg to deal with multiple manufacturing tasks simultaneously. An important issue with cloud manufacturing is therefore to optimally schedule multiple manufacturing tasks to achieve better performance from a cloud manufacturing system. Task workload provides an important basis for task scheduling in cloud manufacturing system discussed here. They proposed a new task workload and service modelling methods which incorporates new elements like service quantity, service efficiency and enterprise capacity. All these enable dynamic calculation of the time for a service to fulfil a task or a subtask as well as service utilization. Their research indicates that scheduling larger workload tasks, especially ones with a higher priority is a better strategy in achieving better system performance, a higher service utilization, and a higher rate of successful execution of tasks without compromising with the task quality.

Hehua Yan *et al.* [22], discusses integration of robotics with the cloud technology which demonstrates a new approach to task execution as well as resource sharing compared to conventional industrial robots. There are a few challenges associated with cloud robotics. Highly flexible load scheduling mechanisms are still very immature. Also, due to time variability and service quality, the traditional optimization mechanisms for the network service quality do not meet the requirements of smart manufacturing. This research paper analyzes the cloud robotics w.r.t. different backgrounds like cloud computing and big data. The applications of cloud robotics show that enormous computational and storage issues can be fundamentally improved. The countermeasures discussed for improving the performance are: 1) Adaptive optimization mechanism for network service quality; 2) Load scheduling mechanism based on task complexity and network service quality; 3) Cloud-assisted cooperative learning mechanism. This study comes up with solutions to address key issues pertaining to service quality, scheduling mechanisms and a self-learning mechanism in cloud robotics.

David Golightly *et al.* [23], give a human factors perspective on manufacturing in the cloud. Human factors can have a significant contribution to the paradigm of cloud manufacturing, well as long as humans are involved in the loop. The issues and opportunities can be found at the service provider, application provider and consumer user groups. Human factors contribute across various groups through Human Computer Interaction (HCI) and User Experience (UX), user-centered automation, and product design. Consumers can have more say in the product design, either explicitly through customization requirements or more implicitly through the data footprint of their use of products and services. The potential of a modelling software and cheaper prototyping can lead to a more iterative, user-centric approach towards product design. Whereas on the operating side, if well-managed, the introduction of new forms of technology can give manufacturing service providers the opportunity to bolster their workforce, while realizing more efficient and cost-effective automation. Human factors can be open and responsive to such opportunities.

This section covered literature on cloud-based manufacturing systems. Initially, it discusses how a cloud manufacturing system is used and how the commercial cloud computing platforms perform in such cases. An important part of enabling cloud manufacturing services are the various protocols that can be used for communication by machines in a cloud manufacturing setup. Next section 2.3 will discuss about the industrial internet protocol used in this thesis as well as literature on the alternatives that have been used.

工业自动化通信协议

## 2.3. Communication Protocols for Industrial Automation

A communication protocol [45] primarily can be defined as a set of rules which allows multiple entities of a communication-system to transmit information through any variation of a physical quantity. These rules or standards define the syntax, semantics and synchronization

语法 语义 同步性

of communication. The communication protocols may be implemented either by hardware, software, or most likely a combination of both.

In this section, various machine-to-machine communication protocols are discussed.

本节中 将讨论机床交流间的不同协议

The focus in the first section 2.3.1 is on the industrial automation machine-to-machine protocol used in this thesis called MTConnect. This is followed by a discussion of an alternative communication protocol called Object linking and embedding for Process Control Unified Architecture (OPC-UA), in the section 2.3.2. It also briefly describes a few low-level messaging protocols like MTQQ and RabbitMQ, which are newer protocols being built to enable the Internet of Things.

### 2.3.1. MTConnect Industrial Standard

MTConnect [36] [47] is an open technical industrial standard developed to extract machine-specific process information from numerically controlled machine tools. It is a

从数控机床中提取数据

protocol developed for the exchange of data between machines and other equipment on the shop, and a cyber-physical system. This kind of data is primarily used for monitoring and data analysis. MTConnect is referred to as a read-only standard which means that it only defines

只读标准

reading or extraction of data from control devices and not the writing of data to any machine or any equipment. In other words, it does not have the ability to directly control the machine.

没有能力控制机床 只能读取

There are hardware/software adapters which extract the machine process data and process them

into the MTConnect defined data format, called the MTConnect adapters. This data is
适配器
presented in an XML formatted output called the MTConnect Agents, using Hypertext
代理器
Transfer Protocol (HTTP) as the underlying protocol. MTConnect provides with a RESTful interface which means that the interface is stateless. Such interfaces allow any external third-party system to access values of attributes being streamed by the machine.

The MTConnect Agent in turn consists of four kinds of XML outputs called the devices, streams, assets and errors [41].

- Devices:
  设备
  - Descriptive about the configuration machine and the type of data that can be delivered.
  - Returned by /probe command issued to Agent.
- Streams:
  数据流
  - Data samples and events from the device/s.
  - Returned by /sample or /current command issued to Agent.
- Assets:
  资产
  - Retrieve information on mobile assets.
- Error:
  错误
  - Returned when an error occurs which prevents further processing.

Athulan Vijayaraghavan *et al.* [36], introduced MTConnect as a data exchange
数据交换标准技术
technical standard that allows disparate entities in a manufacturing system to share data seamlessly in a common data format. The information on the efficient use of machines and systems in the production facility will be extremely valuable to designers of processes and

systems as green manufacturing systems 绿色制造 and facilities are continued to develop 可持续发展. The interoperability enabled by MTConnect can provide the mechanism for system monitoring 系统监控 and optimization 优化 with respect to energy as well as resources. H. Atluru *et al.* [29], present a research paper on whether MTConnect can deliver on its promise of data to information. They performed various case studies implementing MTConnect on a shop-floor. They realize that in addition to providing for real-time data transfer between the machine tool controller and the various smart machine-systems, it also enables data access from a host of other applications and modules within the smart machine program. They found that it will be very easy to implement interoperable subsystems across the shop floor with MTConnect. These subsystems can in turn include a variety of devices like programmable logic controller (PLC), databases, scales, maintenance systems, ERP systems and more.

Pei Lei *et al.* [30], introduced an MTConnect based monitoring system for finish machining the assembly interface of a vertical tail on a large passenger aircraft 大飞机垂尾制造. An MTConnect setup of the finish machining system was proposed. The validity and the interoperability of the proposed system are verified by monitoring key data in the aligning 校准, clamping 夹紧 and machining 加工 process. They showed that it is not the advantage of MTConnect to just get and monitor the process data, instead it is the unified standard format and an HTTP+XML based process which provide a plug-and-play setup. Since data streaming in from multiple machines would have a common definitions and values, it would ease the realization of integration and interoperability of various machine types on the shop-floor. The process data collected from the machines could also be used to optimize the machining process in addition to the supervision of the process leading to intelligent manufacturing.

### 2.3.2. OPC-UA & other Industrial Automation Communication Protocols

OPC- Unified Architecture (UA) [42] is a machine to machine communication protocol for industrial automation which was developed by the OPC Foundation. The OPC-UA is a platform independent service oriented architecture that integrates all the functionalities of the individual OPC, an industrial telecommunications protocol, Classic specifications into one extensible framework. This multi-layered approach accomplishes the following set of goals.

- Functional equivalence.
- Platform independence: from an embedded micro-controller to cloud-based infrastructure.
- Secure: encryption, authentication, and auditing.
- Extensible: ability to add new features without affecting existing applications.
- Comprehensive information modeling: for defining complex information.

MTConnect-OPC UA [47], an interesting integrated approach, is a set of companion specifications to ensure interoperability between MTConnect industrial specifications and the OPC Unified Architecture (UA) specifications with the manufacturing technology equipment that implement those standards.

Message Queueing Telemetry Transport (MQTT) [43] is a machine-to-machine data transfer protocol that is growing in its use as a messaging protocol for Industrial Internet of Things (IIoT). What differentiates MQTT from other messaging protocols as an IIoT protocol is its lightweight overhead two-byte header, publish-subscribe model, and bi-directional capabilities that requires minimal network bandwidth. MQTT is suited to industrial control systems applications that access IIoT data. The newest version is based on an open OASIS standard. OASIS is the Organization for the Advancement of Structured Information Standards, an international consortium that promotes the adoption of product-independent standards for information formats.

RabbitMQ [44] is yet another open source message broker software which implements the Advanced Message Queuing Protocol (AMQP). The RabbitMQ message broker, in a case study done by Google, deployed atop Google Compute Engine where it demonstrated the ability to receive and deliver more than one million messages per second. To put this volume in context, one million messages per second translates to 86 billion messages per day. U.S. text messages reached 6 billion per day in 2012. Apple processes about 40 billion iMessages per day, and WhatsApp recently hit a new daily record in December when it sent 20 billion daily messages. RabbitMQ is already a player in the ever-growing field of real-time analysis, big data and IoT. While it has been extensively used in text-messaging protocols, it is yet to be seen how MQTT and RabbitMQ can be utilized in manufacturing factories.

This section talked about the various machine-to-machine communications protocols. Besides MTConnect, there are plenty other protocols which have been discussed. Protocols like MTConnect-OPC-UA provide with an interesting idea of integration of such protocols to fruit the benefits from each of the constituent protocol. The current literature presents with an opportunity as well as a challenge to transcend the bounds of limitations. The next Chapter 3 and 4 present address the Research Objectives and explain the steps taken to successfully achieve them.

研究目标

## CHAPTER 3

## STREAMING DATA FROM MANUFACTURING MACHINES

机床数据流

### 3.1. Introduction and Problem Background

The challenge in dealing with data from legacy manufacturing machines is to figure
难点：从传统机床中提取数据
out the interface to extract in-process data from its controllers during part fabrication. Most legacy machines are incapable of digitally interacting with the outside world. However, the newer machines have an MTConnect adapter which enables them to extract and format the data and stream it on to the network as the MTConnect agent. Similarly, a set-up can be created to facilitate the streaming of manufacturing data from legacy machines. Once the data is streamed to a database or onto the cloud, an ecosystem of third party apps can be built on top of it enabling interpretation and analysis of the data.

### 3.2. Technical Approach

This section discusses the technical approach involved in addressing the problem statement. Section 3.2.1. explains the details of the machine setup at NC State. This involves the adapters for extraction of data from new as well as old legacy machines. Section 3.2.2.
针对 新机床 和 老机床 进行数据采集
discusses the middleware architecture setup using a local Red Hat OS with a PostgreSQL
中间架构
database. Section 3.2.3. describes the sample part that was machined and for which the machine data was analyzed and used to demonstrate various apps.

#### 3.2.1. Machine Setup and Information Flow Diagram

机床安装 以及 信息流示意图

The machine set-up in the NC State laboratory is shown in Figure 3.1 and Figure 3.2. Process related machine data from a HAAS VF2 is stored in System Variables or MACROS
哈斯VF2 机床 系统变量 宏指令

as defined by HAAS. These MACROS are then read by an MTConnect hardware Adapter.
宏指令 被 MTConnect硬件适配器读取
Alongside machine data, sensor data is also collected through a Hall sensor measuring spindle
机床数据 传感器数据 霍尔传感器
power consumption and a three-axis accelerometer installed within the workpiece holding
主轴能源消耗 三轴加速器
fixtures within the machine. As seen in the Figure 3.2, a local system or a raspberry-pi can be used as the MTConnect Adapter in combination with corresponding machine hardware available at the HAAS machine. There are possibly two modes of enabling legacy machines to be Ethernet compatible. First, a low cost <$150 computer through the machines' RS232 port. Second, when there is no PLC or any compatible ports to pull data from, a low cost system-on-chip board such as a Raspberry Pi or a Beagle Bone Black, to directly interface with the
树莓派
machine control boards. In both cases, the machine data formatted through MTConnect standards was streamed through the university ethernet infrastructure to the local DB maintained within the university. The software side of the adaptor is a python script which runs a series of multiple queries and extracts relevant machine data as requested by the database. The local computer system at the HAAS runs its own agent within the board. The data collected by the adapter is filtered and stored in specific tags as defined by MTConnect. Using the TCP/IP connection via the NCSU server, the data is collected at the central system where it is produced as an MTConnect agent output (XML file format). This agent is then used to extract and push data into relevant tables within the database.

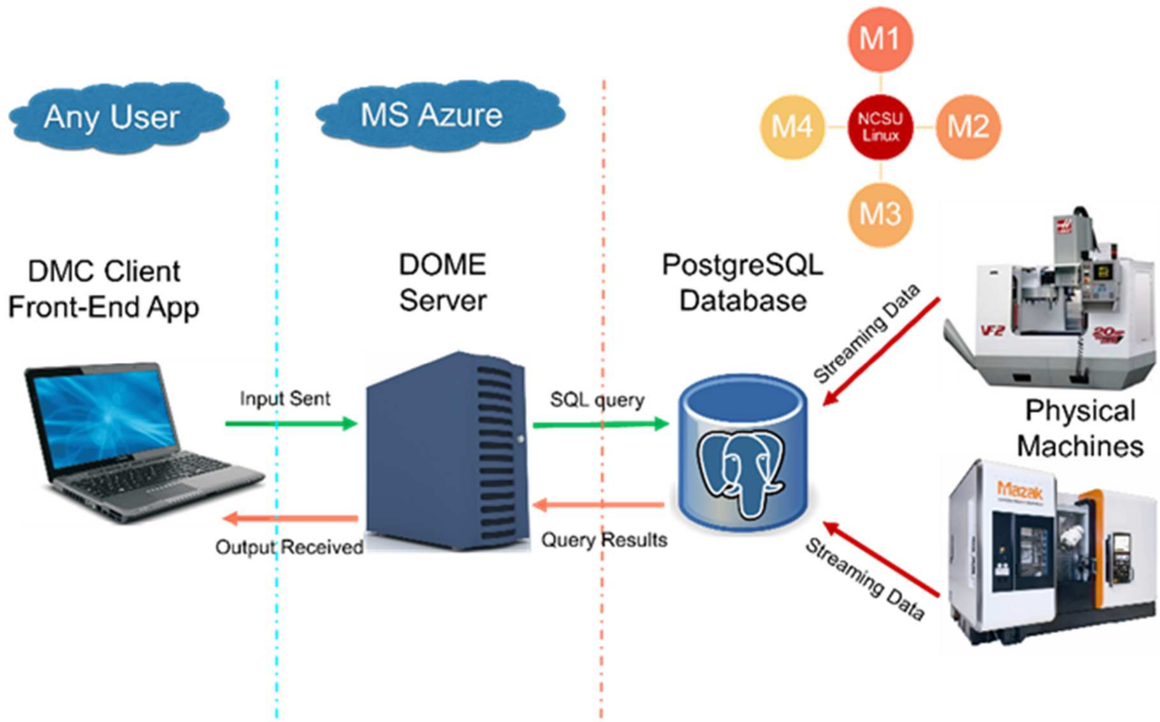



Figure 3.1: Information Flow from Physical Machine to Client Applications

Many such set-ups can be replicated for different machines across a shop-floor. Another implementation is directly connecting a preconfigured MTCONNECT enabled MAZAK Integrex i-100ST as shown in Figure 3.2. The MAZAK and HAAS along with any other machines will stream data through its respective agent to the local DB. The external hardware setups, as demonstrated at the HAAS, allow legacy machines to be interfaced for digital communication. Besides which, ‘smarter machines’ i.e. MT-CONNECT enabled, like MAZAK Integrex can also stream the data generated during machine operations to the local database.

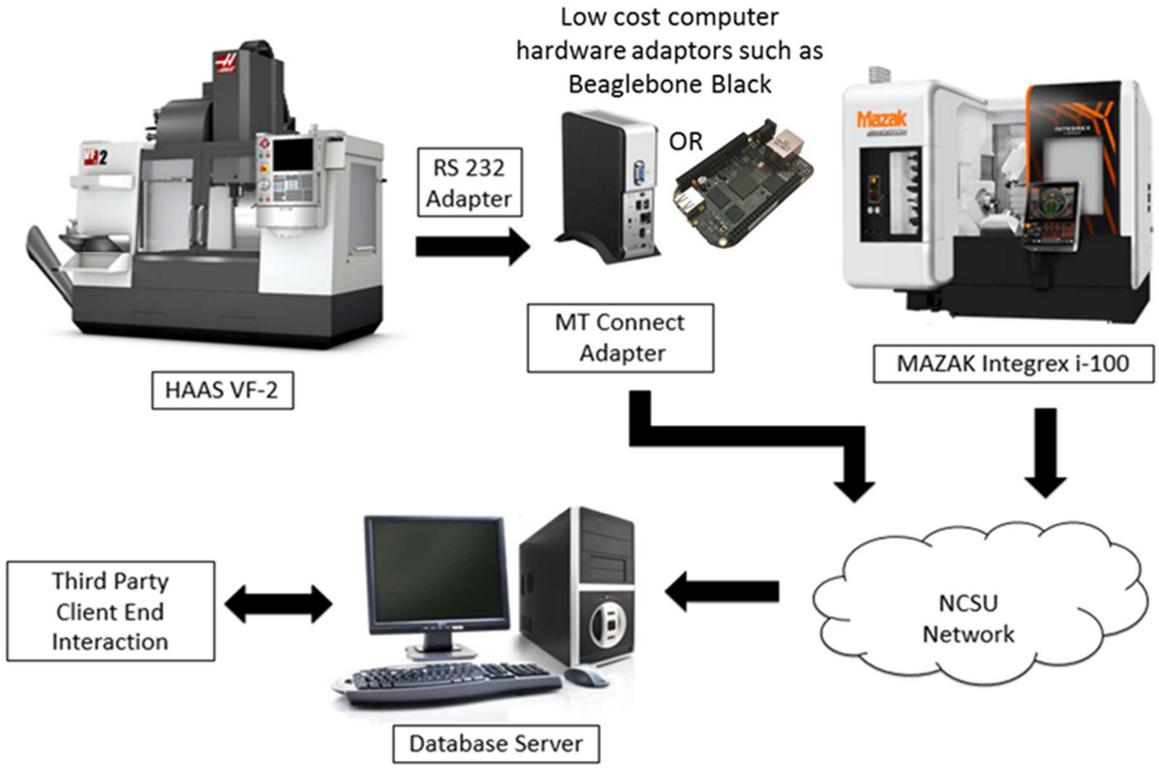



Figure 3.2: Detailed information flow from physical machine through hardware and software adaptors to third party client, such as Digital Manufacturing Commons (DMC) client.

Figure 3.3 shows the MTConnect adapter interface at the Benchman 4000 machine in the ISE lab. The Benchman 4000 is a low level 20-year-old legacy machine incapable of interacting with the NC State network. As seen in the Figure 3.3, various sensors; current, IR, press and light sensors, had to be installed to extract or sense the data out of the machine. The major difference here with the HAAS VF2 or the MAZAK discussed above is that it does not have any active ports for communication. This called for installation of sensors. Here, the Arduino serves as the MTConnect adapter and streams the Agent onto the NC State network from where it is streamed into the local PostgreSQL database.

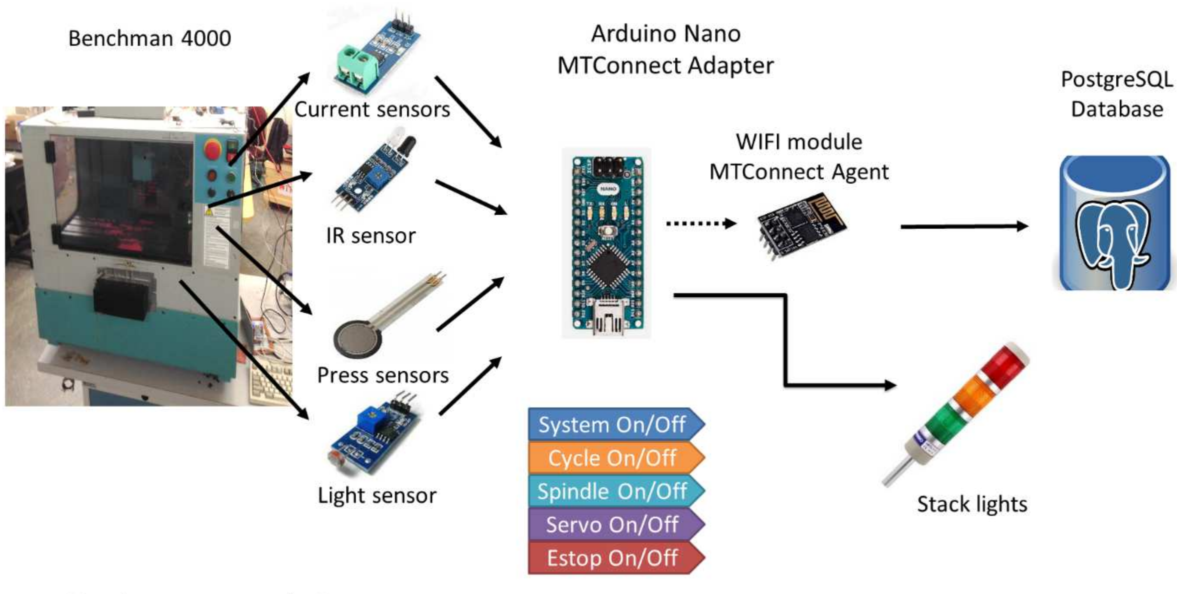



Figure 3.3: MTConnect Adapter Setup for Benchman 4000

机床中间件架构扩展第三方应用

### 3.2.2. Machine Middleware Architecture Enabling 3rd Party Apps

A high-level logical structure of the middleware is presented in Figure 3.4. As described earlier, each physical machine with its respective controller (such as SIEMENS, FANUC, MAZAK) will have a corresponding MTCONNECT Agent, which continuously pushes data from the machine to any service that requests them. The underlying streaming data from controllers and any sensors associated with the machine is then stored in a structured DB. To make this data useful, scripts organize this data in three hierarchical layers. At the bottom
三个层次
of the stack is the machine information layer, which contains raw data from all machines on
底层：机床信息
the shop-floor. The 2nd layer contains process related information, particularly the processes
第二层：加工过程信息
carried out by the machine. If available, this layer obtains input from the other shop-floor ERP or MES systems. A summary information layer is computed from the two layers below it and
信息汇总层：计算层

made available through a Generic Access Library. Any 3rd party app then retrieves data
第三方APP从GAL中获取数据
through the Generic Access Library (GAL). The layers below GAL provides hardware abstraction, meaning that any third-party software does not have to directly access any of the hardware devices at the shop-floor for proper function. This middleware architecture is quite like the hardware abstraction provided by the Android Operating System which allows any software developer to build apps without directly accessing any of the specific hardware contained in smartphone.


Figure 3.4: A high-level middleware architecture interfacing machines on shop-floor to 3rd party ecosystem of apps.

The information from the MTConnect agents is stored in structured tables in the database. Each data point is associated with many elements that characterize the capabilities of the machine type. A python script reads each of the attributes of the element in the xml tree

介绍 PostgreSQL

that the Agent generates and sends it to the database. Some of the key elements of a component are its timestamp, type, id, sequence and its value. PostgreSQL was chosen primarily because of its capability of writing high frequency time series based data. It uses a multiple row data storage strategy called MVCC (Multi Version Concurrency Control) to make PostgreSQL extremely responsive in high volume environments. MVCC is the method PostgreSQL uses to handle data consistency when multiple processes are accessing the same table. It is particularly suited for Big Data type applications. Yahoo, Instagram and Myspace are few of its prominent users.

For the purposes of this study, data is streamed to the PostgreSQL DB at rates of 100ms. 数据库读取数据100ms For example, when the MAZAK machine is performing a cutting 切削 operation, the MTCONNECT data stream page is refreshed every 100ms or at 10Hz. At each 100ms refresh rates, scripts read the XML page and stores relevant data from the file. When data from the sensors were generated, we took an alternate approach, where data is first collected into a local file first and then at the end of the operation, the file is written to the DB tables. This approach is far more superior and efficient than directly writing sensor data to the disk. This allows multiple machines to stream data whenever a process operation is complete. Any filtering maybe done at the machine level to save on disk space and computation time.

### 3.2.3. Sample Part File Machined

简单件加工

The entire data collection for one of the apps discussed in the section 3.2.2. is based on machining of a simple part. In Figure 3.5, the SolidWorks CAD file of the part model is displayed. It also shows the tool path 加工轨迹 of the operations that are performed on the part. It starts with face milling the top surface, followed by outside contour milling to define the dimensions of the part. Pocketing is performed which is followed by inside and outside chamfering of the edges. Figure 4C shows the final machined workpiece. The tool path basically defines the movement of the bed of the HAAS CNC machine. The data collected then can necessarily be mapped to do further analysis. It can be used to make many applications and perform various analysis on the front-end. The next section demonstrates applications built by interaction with the data streamed from various machines.

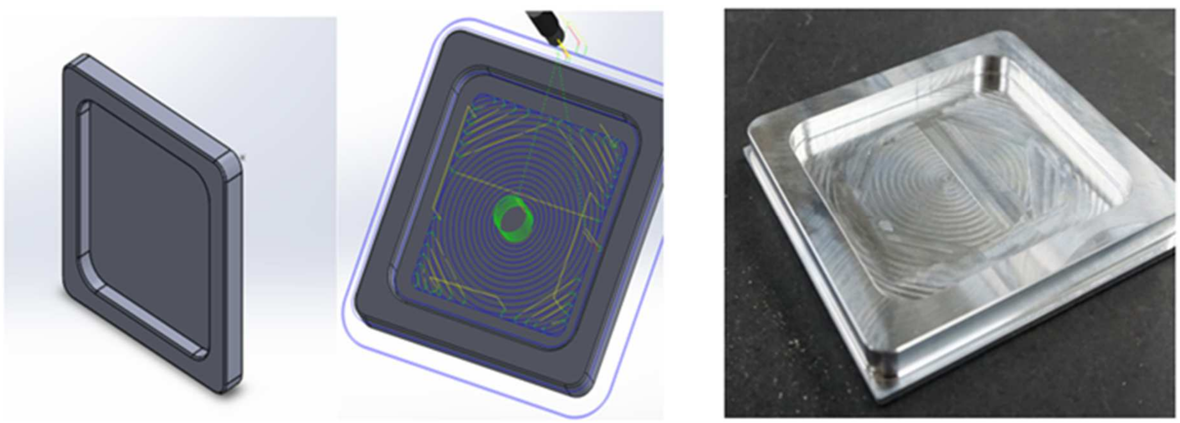


Figure 3.5: A) SOLIDWORKS CAD file; B) CAM tool path; C) Machined Workpiece

### 3.3. Results

机床数据流

#### 3.3.1. Streaming Machine Data to Digital Manufacturing Commons (DMC)

As discussed earlier, DMC is a free and open-source software project to develop a collaboration and engineering platform, which enables plug-and-play functionality across the entire digital thread from product development to manufacturing and services. The apps discussed in this section may not be suitable for enterprise level applications since data is directly streamed to a 3rd party open platform. However, this would be perfect for academic and government labs to share data among the scientific community. The information flow shown here can be scaled to hundreds of manufacturing machines across the world, all streaming data to a cloud platform. Users, such as students and researchers can then download process related data files for analysis or process development. These models can be clubbed as ‘Data-as-a-Service’ Models which allow a repository of shareable manufacturing related datasets.

机床状态

#### 3.3.1.1. Application 1: Machine Status

Figure 3.6 demonstrates the “Machine Status” app. The user defines the inputs as per the format in the default html interface. At NC State University, three machines are connected to the local database which in turn weekly sends the updated data to an AZURE PostgreSQL DB on the DMC end. Once the inputs are defined and the user hits the Run button, the DMC does all the calculations at the backend in an Ubuntu Linux environment. Then the html page on the front end is refreshed and the user can see all the outputs as seen in the Figure 3.6. This app focusses on retrieving data related to machine performance and efficiency along with the status updates. Parameters like OEE and part count can be obtained which help the user realize the efficiency of the machines on the shop floor.

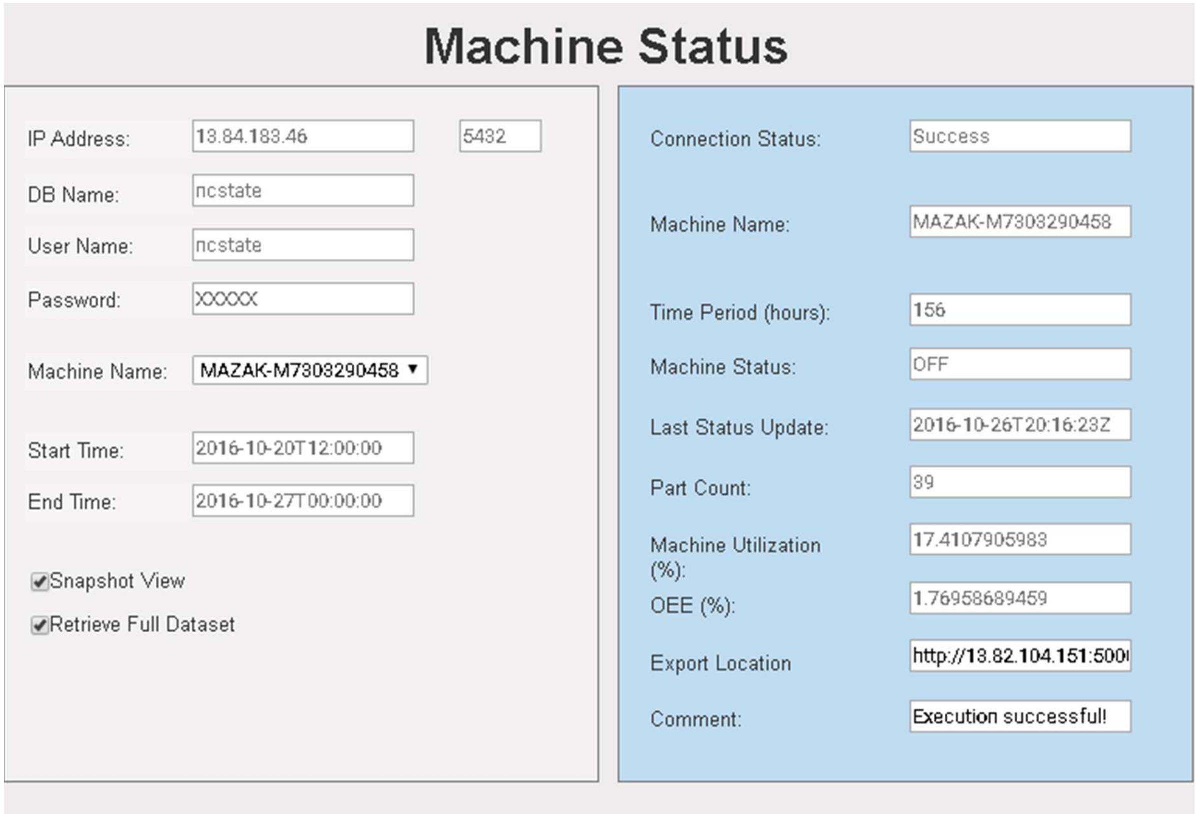



Figure 3.6: DMC App: Machine Status

#### 3.3.1.2. Application 2: Machining Data Plots 加工数据展示

Figure 3.7 shows the Machining Data app which retrieves sensor data from the database and calculates averages of the sensor values. The data is collected from two different sensors, Hall sensor 霍尔元件 and Vibration sensor 震动传感器, presently. This application has a similar input interface as that of the previous app. It also gives an option to the user to extract all the relevant data in a csv file. The user has the option to either simply collect the data or take a snap view at the parameters calculated using the data and the plots between Hall Sensor and Vibration Sensor data versus time. This helps in studying the how the behavior of the current and vibration fluctuations differs during a cutting cycle when compared to an idle state.

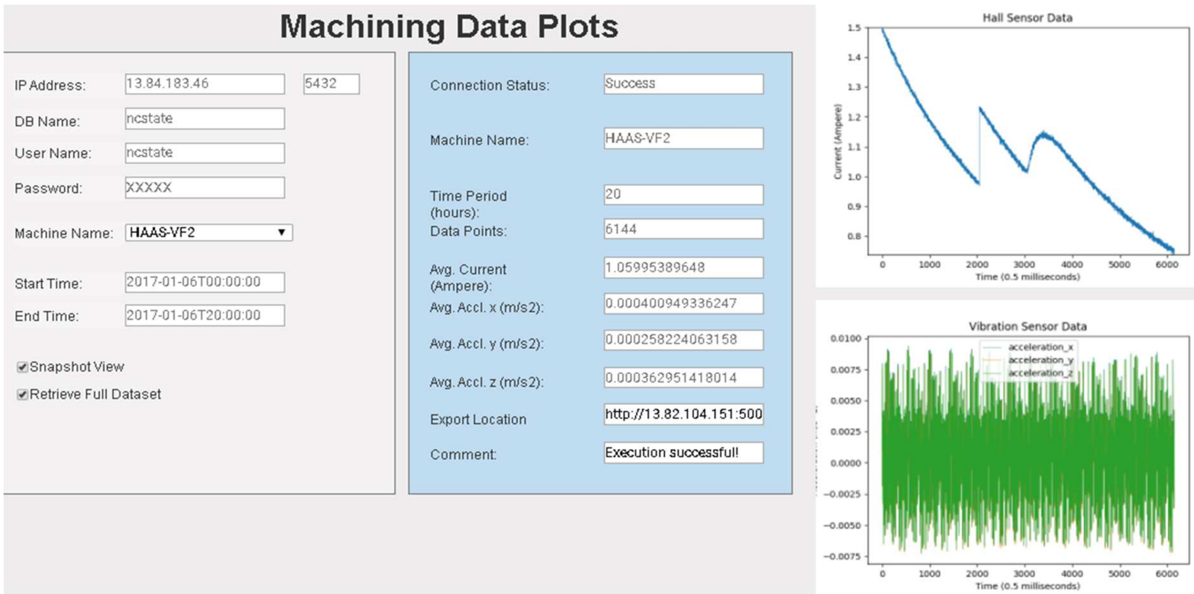



Figure 3.7: DMC App: Machining Data Plot

### 3.3.2. Application 3: Analysis of Machine Data Cloud

机床数据云分析

The main goal of this app is to demonstrate the ability to perform real-time digital 软件的目的：阐述铣床生产零件实时监控 verification of a part produced on a milling machine. This verification is performed by comparing the geometry data of a part with actual position data reported by the machine axis 对比：几何数据 和 实时数据 systems. Real-time streaming data from the machine is stored in a database instance from which client apps can interface with the stored data. Scripts in the database request data from the machine through the agent-adaptor framework of MT-CONNECT. The connection to the HAAS VF2 is made using an RS 232 cable. The cable is in-turn connected to a very compact a local system (or a system on chip Raspberry Pi) which acts as the adaptor. The formatted data is then presented by the Agent in an XML file format which can be obtained over the

NCSU network through an HTTP link. This data is then extracted and stored in a PostgreSQL database schema structure. All this sequence of events happen simultaneously in real-time.

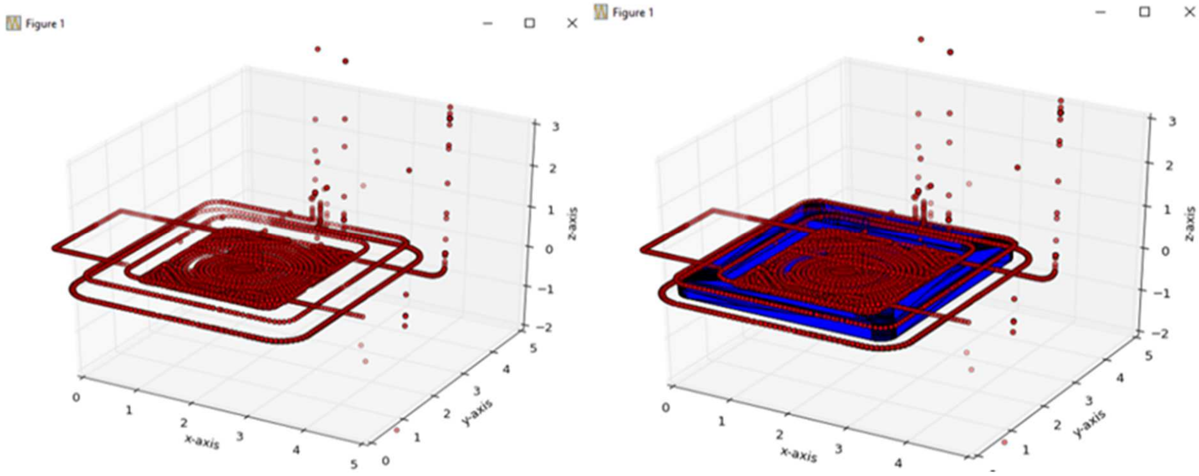



Figure 3.8: From Left to Right: (A) Raw Machine Data Cloud; and (B) Merged STL and Data Cloud

The data collected from the machine can be predefined. The linear x, y and z coordinates along with the spindle speed, tool number have been collected at the machine interface. The app extracts the [x,y,z] coordinates from the database. The data cloud obtained from the set of 3-D coordinates are then plotted against the original CAD file. For the ease of analysis, the initial CAD file was converted to an STL format. The raw machine data cloud of axis movements, as seen in Figure 3.8A, is simply the scatter plot of the xyz coordinates obtained from the database. This data cloud corresponds to the actual path followed by the center-point of the tool. Figure 3.8B shows the merged plot of the STL file and the data cloud. It gives an idea about the real tool path is with respect to the digitally defined part geometry.

### 3.3.3. Application 4: Interfacing the Machine Database with LabView

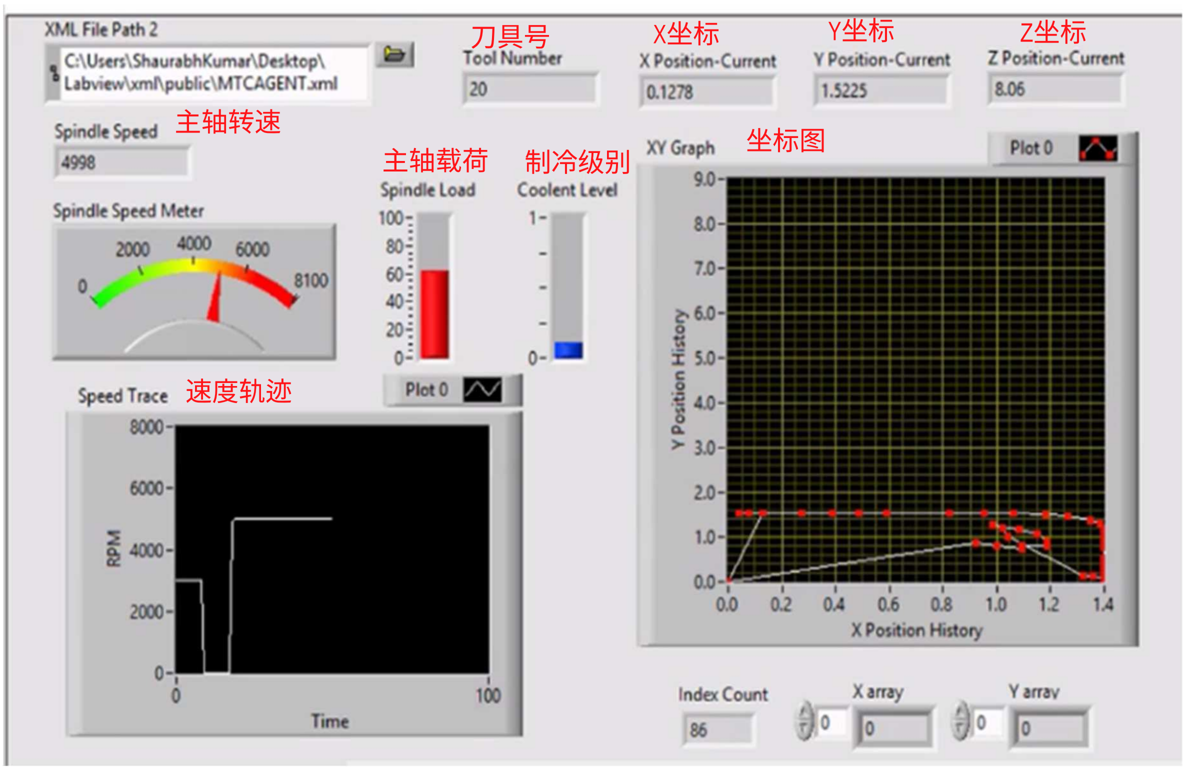



Figure 3.9: LabView Interface demonstrating remote monitoring feature

This application primarily focusses on two aspects. First, it displays live incoming data
此应用主要在两个方面。 第一：实时更新显示数据
into a user interface which can be viewed by any student within the NCSU network. This entire GUI was built using widgets which allowed programming the interface relatively easy and did not involve extensive UI programming skills. Secondly, from specific http links the source
第二：可以下载XML文件
files of the code can be downloaded and the XML file can be viewed, exclusively on the NCSU server. Figure 3.9 is an example of the interface seen through one of the systems at NCSU. This aspect represents the classical machine monitoring application. Except here, building the

GUI interfaces was done through the LabView toolkit, an example of a professional grade 3rd party platform.

LabView编程

### 3.4. Evaluation of Approach 方法评估

There are several challenges to adopting such an approach on a discrete product focused shop-floor. Few of them have been discussed below with potential solutions on possible ways in which the community could address them.

This research highlights ways in which streaming data from sensors and the machine

亮点： 将机床和传感器中的数据存储到云端

control system can be stored in the cloud. However, more effective data management strategies must be adopted. Newer database schema architectures must be developed along the lines of

新数据库必须依赖云计算

edge, fog and cloud computing technologies. Not all data needs to be stored in a database.

并不是所有数据都要存储在数据库中

Some of the data can be locally filtered based on intelligent algorithms and relevant data can be streamed and archived for future use. Examples of fog and pervasive computing can be

雾计算 普适计算

borrowed from mobile computing technologies for effective manufacturing data management.

While technologies exist to gather tremendous amount of data, there is still a lack of effective physical models that can act upon this streaming data. For example, how can

缺少有效的物理模型支持大量的数据流

streaming data from the machines be effective at cutting tool management or enabling

有效利用 和 预测

predictive based maintenance. The community has begun to think of possible solutions through – Cyber Physical Architecture and the Digital Twin solutions. Better physical models need to

物理系统架构 数字孪生

be built at the enterprise level where companies can see immediate benefit from streaming data from the machines.

Many of the manufacturing machines on the shop-floor have been around for more than 10yrs, some even at 20 to 30yrs of active use. They have been retrofitted with a number of controller upgrades over the years. Still the challenge of making them 'internet' compatible is daunting. While there are solutions of low-cost system-on-chip boards that can help alleviate the problem, implementing it may be hard and expensive. If legacy machines can't be brought online to the enterprise level IT-system, then digitalization can't be achieved. These solutions will only be applicable to the newer manufacturing machines. This problem must be addressed before digitalization can be truly achieved in shop-floor level manufacturing.

章节总结

## 3.5. Chapter Summary

In summary, a solution to interface machines on a shop floor towards a digital factory
总结：在车间，通过中间架构解决机床的交互问题
solution through a middleware architecture is discussed here. High granular streaming data
高粒度数据
from the machines can now be efficiently stored, archived and retrieved. This pilot solution can be replicated for all machines on the shop-floor. Machine monitoring apps are deployed to
机床监控app
the DMC, opening the possibility of integrating a variety of machines on the shop-floor. A prototype application to verify a part's geometry by comparing original data with the
对比 原始数据和机床中实时坐标采集的数据
coordinates obtained from the actual machine coordinates is also demonstrated. This will help to optimize part probing during machining and also provide an initial verification pathway for part tested. This research essentially bridges the machine and the digital platform at the back-end and explores manufacturing intelligence at the front. The ultimate goal is to create a digital
最终目标：创造一个数字化网络
network which would replicate the above described system for all the machines in the ISE lab at NCSU. This would introduce the academic world to digitalization.

The proposed middleware solution enables hardware abstraction layer at the bottom which reduces software development costs. The apps demonstrate a low-cost solution to obtaining data from manufacturing machines on the shop-floor including legacy manufacturing machines that may not have sophistical machine controls or PLCs to extract data from. The solution discussed here demonstrates how incoming streaming data can be stored into a next generation structured SQL database, such as PostgreSQL. This is in contrast to using traditional SQL type databases such as Access, ORACLE or MySQL. PostgreSQL, an open source database is well suited for fast writing of streaming time-series based data.

## CHAPTER 4

## Verifying Instructions from Cloud Front-End User with Physical Machine Capabilities

验证云端用户的指令

### 4.1. Introduction

The current trend in the manufacturing industry calls for a new generation of production system with better interoperability and business models [26]. "Cloud" provides new service models and more opportunities for the manufacturing industry. In this Chapter, a Cloud-based manufacturing system is developed to support ubiquitous manufacturing, by connecting all the

本章中，一个应用云的制造业系统，通过全球连接所有的机床和制造者

machines and manufacturers across the globe.

Consider a case where product designers have to manufacture parts for which a manufacturing process plan is built – say the steps needed to machine a prismatic part. Once the CAD/CAM analysis is done, the next step is to find the right manufacturer who can build the part. This process of finding manufacturers and figuring out whether the machines on their shop-floor can meet the product specifications and thus build the part or not, consumes a lot of time. Processes like identifying suppliers through the internet, social media and trade shows can be time and resource consuming. It is highly challenging to land the right manufacturer who has the right set of machines with the necessary specifications to meet the product specifications. It is not feasible for designers to share their design and manufacturing process files across the machine shops in the country. Privacy and security of the data is a big worry. The cloud-based cyber-manufacturing system discussed in this Chapter demonstrates how it can act as the bridge between the product designers, vendors and manufacturers and in turn

桥梁： 产品设计者—商人—厂家

automate the interaction. Since "Cloud" is omnipresent, anyone from anywhere can access this

无所不在

platform to find their right manufacturers. It can serve as a global platform for collaboration saving time and resources which would be significantly spent if done manually.

In the previous Chapter, the cyber-physical setup involved a private computing system 私人系统—红帽Linux OS in a RedHat Linux OS. With such a system flexibility, scalability and global collaboration is difficult to achieve. Whereas, a cloud based system not only addresses these factors but also supports disaster recovery, capital-expenditure free services and improved security. Cloud-manufacturing systems with such characteristics have the potential to revolutionize collaborative manufacturing to enable supply chain wide collaboration.

技术方法

## 4.2. Technical Approach

In this section, the technical approach towards a cloud-based manufacturing system is discussed. Section 4.2.1 lays out the algorithm behind the entire setup in a flow-chart. It is followed by the understanding of the cyber-physical setup in the section 4.2.2. Section 4.2.3 focusses on comprehending AWS and its various services and their roles. Section 4.2.4 further discusses the various tables in the RDS PostgreSQL database which also serves as a Machine Data Library.

亚马逊云服务 PostgreSQL数据库服务

### 4.2.1. Algorithm

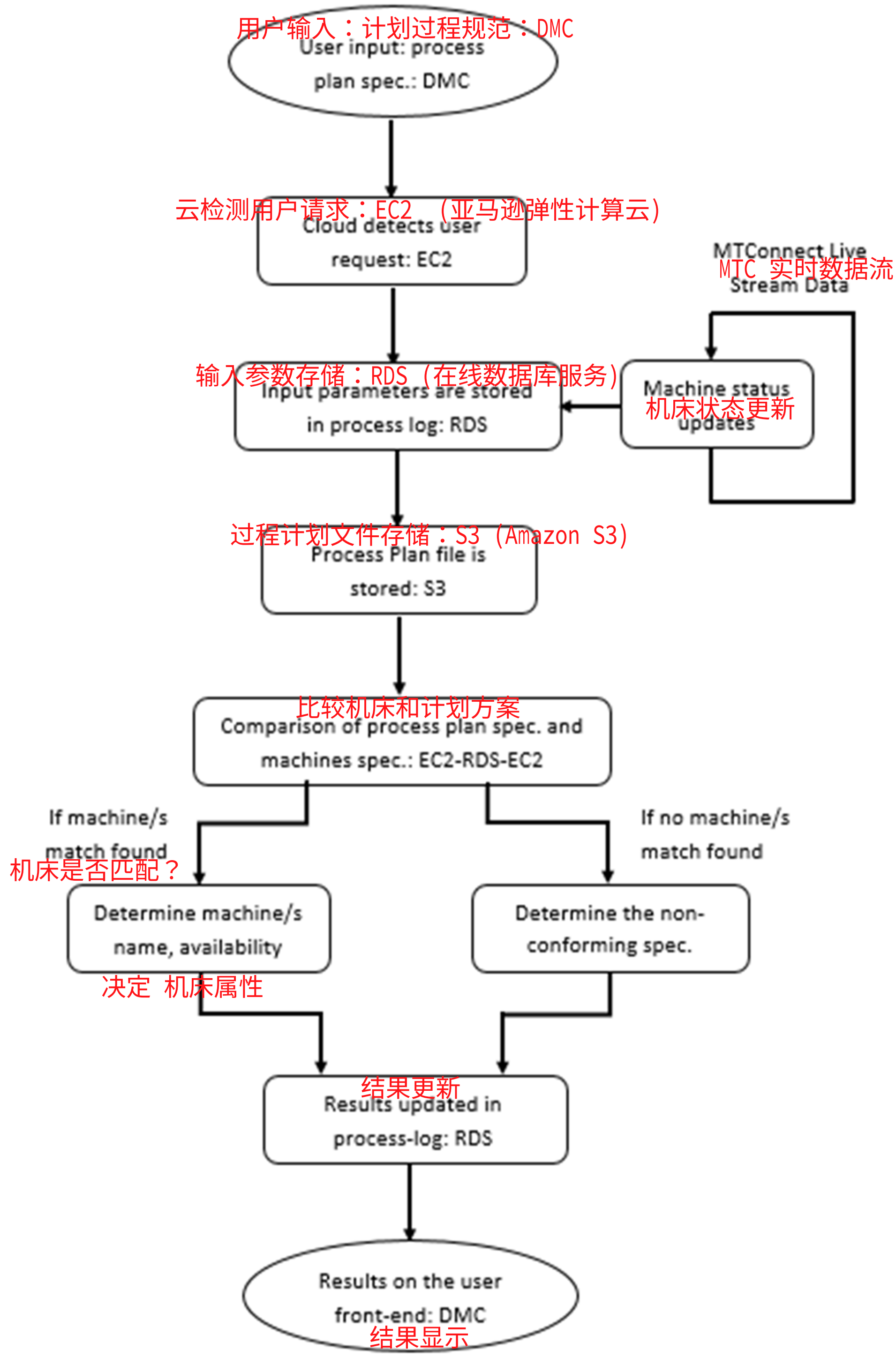



Figure 4.1: Algorithm-Flowchart for data interaction

The algorithmic flowchart for the data interaction can be seen in the Figure 4.1. The user strictly interacts at the front end and needs to provide the material name and the setup sheet file for the part that is to be machined. The file address for the setup sheet is an http link which is read and stored in the cloud. The middleware then reads through the setup sheet and then runs through the live machine information stored in the Amazon cloud. It then follows it up by mapping all the input parameters to the machine parameters based upon the relations predefined. Once it is done, it reports back to the user through the front end. The result that the user would see includes whether any machine is capable of manufacturing with the specifications in the setup sheet.

物理信息系统安装

### 4.2.2. Cyber-Physical Setup

The cyber physical setup can be seen in the Figure 4.2. In this section, the three distinct sections of this setup are discussed.

- Physical Machine Backend 物理机床后端—后端
- Middleware Architecture hosted in the Cloud 中间架构—云端
- User Interface Frontend hosted in the Digital Manufacturing Commons 用户交互界面

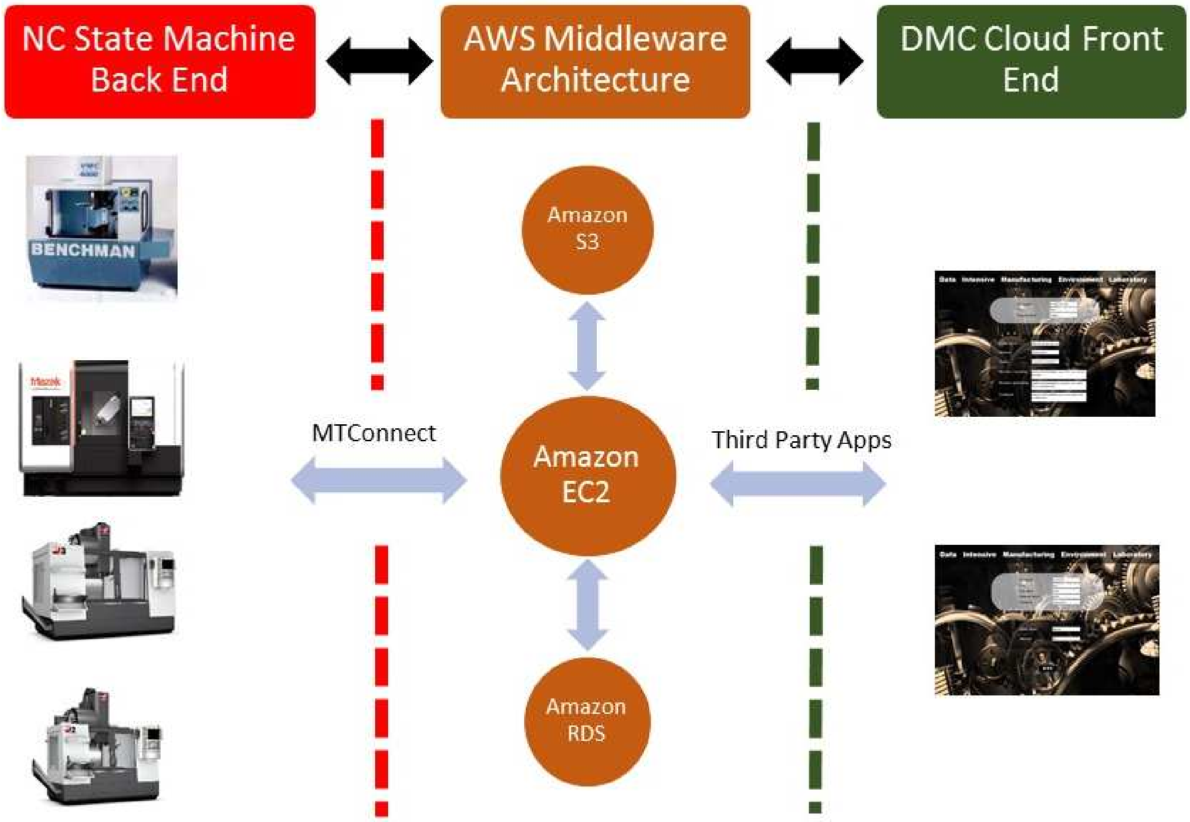



Figure 4.2: Cyber-Physical Setup

The machine backend consists of the machines on the shop floor that are digitally connected to the network. Presently at NC State, three machines are digitally connected: 1) Mazak i-100 ST, 2) HAAS-VF2, and 3) Benchman-4000. As discussed in the previous chapter,
马扎克 哈斯
each of these machines have an MTConnect Adapter-Agent setup either manually installed or
MTC 适配器
came pre-installed. The MTConnect agent is then streamed onto the NC State network.

MTConnect formatted data provides interoperability between different kinds of machines and machine equipment. Such a data-driven manufacturing setup allows management to a) observe trends in production and labor time, b) make corrections to the
在生产和工作中观察趋势

maintenance and quality issues, and c) minimize business as well as safety risks throughout
产品质量和可持续监控 提高安全性
the operation. In manufacturing, material and human resources are the mainstays of the
支柱
business, manufacturers can now utilize data-driven processes to understand how to make better use of employee skills and time. Relying on data means manufacturers can more easily assess risks as well as predict the impact of design on product quality and sales. To demonstrate the use of such a data-driven environment in manufacturing, this thesis utilizes the cloud computing services of the Amazon Web Services in the Middleware Architecture.

As mentioned earlier, the middleware architecture is built using the Amazon Web
中间架构建立在亚马逊云服务平台
Services (AWS). As is seen in Figure 4.2, the middleware structure uses three services of AWS: 1) Amazon Simple Storage Service (Amazon S3), 2) Amazon Relational Database Service (Amazon RDS), and 3) Amazon Elastic Compute Cloud (Amazon EC2). Together, the middleware is responsible for saving input files from the front end, streaming in the
保存终端输入文件
MTConnect Agent Data from the machine back end to the hosted DB, and processes the data
MTC 数据保存到数据库
w.r.t the input parameters and reports back the output to the user. The middleware can be better
将 数据反馈给用户
understood in the next section where all the services used are discussed at length. The front-end consists of a similar DMC interface as discussed in the previous chapter for Applications 1 and 2. DMC users are able to interact with the middleware by providing the input necessary. The middleware processes the request and provides the result to the user.

#### 4.2.3. Middleware Architecture: Amazon Web Services (AWS)
亚马逊服务

This section describes the three Amazon Web Services used for this research. Amazon Web Services (AWS) offers cloud computing platforms. AWS is marketed as a service to
提供云计算平台
provide large computing capacity. AWS has more than 70 services of which 3 are discussed

here. As can be seen in Figure 4.3, an overview of the roles of the services is represented. The first service in use is the Amazon Simple Storage Service (Amazon S3) [37], which is a storage service to store and retrieve any amount of data from anywhere on the web. Customers use S3
存储数据 检索数据
as the primary storage for cloud-native applications and with the purpose of backup & recovery. It is very convenient for the user to transfer large chunks of data into or out of Amazon S3. For this research, Amazon S3 is defined to store all the input files from the front-
S3 用来存储输入文件 例如:表单 零件
end, like a setup sheet for a part to be machined.

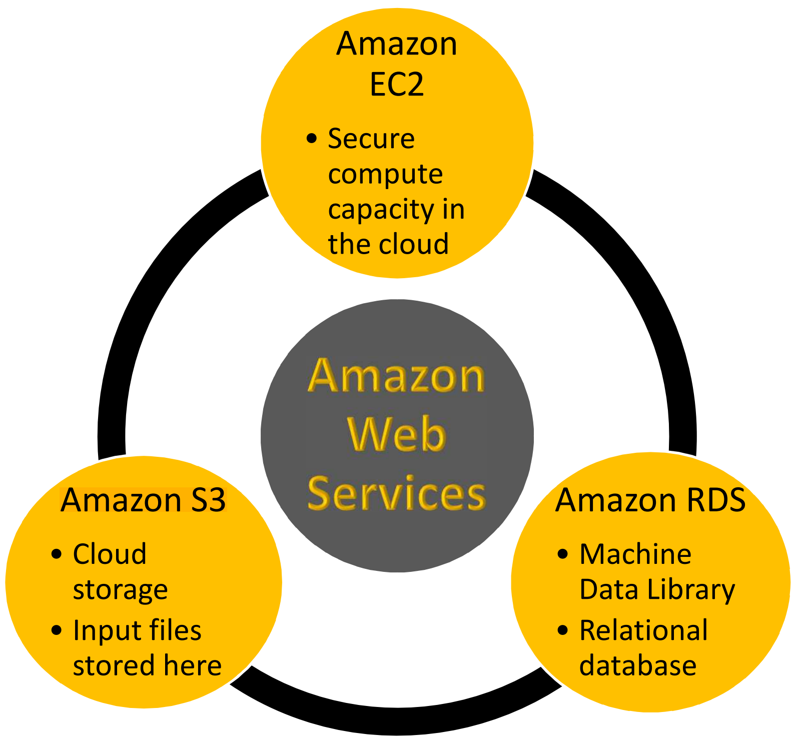



Figure 4.3: Amazon Web Services Used

The second service used is the Amazon Elastic Compute Cloud (Amazon EC2) [37]. It
第二个服务： 亚马逊弹性云计算
is a service that provides secure and efficient compute capacity in the cloud. Its major purpose

is to make cloud computing easier for developers. It provides the developers tools to build failure resilient applications. For this research, Amazon EC2 is used to run all the scripts to run in a continuous loop. The scripts running on the EC2 has a two-way interaction with the other two services. When the user sends in the request, scripts within EC2 receives the input file (setup sheet) and stores it in the Amazon S3, where it is stored and can be accessed when needed. It then starts mapping all the input information to the machine capabilities which are defined in the data library in Amazon RDS. After all the processing, it responds back to the cloud front end with the results.

The third service in use is the Amazon Relational Database Service (Amazon RDS)
第三个服务：亚马逊关系数据库服务
[37] makes it easy to operate and scale a relational database in the cloud. Amazon RDS provides resizable capacity while managing time consuming database administration tasks in
大小可变空间
a cost efficient set up. It provides the user with the option of selecting any of the six database
可以选择6种数据库
engines: Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, and Microsoft SQL Server. For this research, PostgreSQL database was chosen primarily because of its efficient times
主要选择
series capabilities and performance in high volume environments. The Postgres database has
大容量环境
several tables which store different kinds of machine and process capabilities information. This database acts as a Machine Data Library which will be discussed in the next section 4.1.3.

#### 4.2.4. Machine Data Library (MDL)

机床数据库

To capture the data from the machines, it is necessary that we store and separate out
获取数据最重要的是：从机床数据流中存储和提取数据
the data streaming in from the machine. The MDL stores both static and dynamic characteristics of the machines, contained within the organization or across multiple enterprises spread around the world. Machine Data Library refers to all the tables in the

Amazon RDS PostgreSQL database instance. In this section, all the tables defined for this research are discussed. The objective of this library is to store all the updated machines' specifications and process capabilities. It also has a table which keeps track of all the previous

存储所有的历史数据

runs. The following Tables will better help understand the content of each one of them along with their purpose. The table structure has been shown with the records from 3 machines. However it can be extended to any number of machines provided they are streaming data in the MT-CONNECT standard.

Table 4.1: Machine Data Library: Machine Specifications

| machine_name | machine_param | units | value |
|---|---|---|---|
| MAZAK-M7303290458 | type | None | mill/turn |
| MAZAK-M7303290458 | wt | kg | 1020 |
| MAZAK-M7303290458 | travel_x | mm | 450 |
| MAZAK-M7303290458 | travel_y | mm | 220 |
| MAZAK-M7303290458 | rtsspeed | rpm | 12000 |
| MAZAK-M7303290458 | turn_dia | mm | 500 |

In Table 4.1, the machine specifications table 机床特征表 is seen where each row stores the name of the machine followed by the machine parameter, and its value and units. Various parameters like machine type 机床种类, maximum weight of the part 最大重量, rapid speed 加速度 and maximum tool diameter 刀具最大尺寸 among others are saved. The values can be updated by the admin who has access to the username and password of the database.

Table 4.2: Machine Data Library: 刀具特征表 Tool Specifications

| machine_name | tool_name | tool_number | tool_grade | tool_diameter |
|---|---|---|---|---|
| MAZAK-M7303290458 | END MILL | 2 | None | 12.7 |
| MAZAK-M7303290458 | END MILL | 3 | None | 4.826 |
| MAZAK-M7303290458 | END MILL | 4 | None | 9.4 |
| MAZAK-M7303290458 | END MILL | 6 | A | 12.7 |

In Table 4.2, the tool specifications table is seen where each entry stores the machine name, tool name, tool grade, tool number and the tool diameter. Since certain tools from different machines can be used for one another, the user will be intimated if the need be. These tools are then checked for their tool diameter and the respective machine name by the EC2 script to check their validity for the part to be made.

机床时间表

Table 4.3: Machine Data Library: machine schedule

| **machine_name** | **day** | **time_begin** | **time_end** | **job_description** |
|---|---|---|---|---|
| MAZAK-M7303290458 | 0 | T09:00 | T11:00 | testing_1 |

Table 4.3 shows the machine schedule table and where the schedules are inputted in advance. These could be updated weekly or daily depending upon the work environment and requirements. Table 4.4, on the other hand, shows a live machine availability status in the machine availability table. This table tells you the updates for each machine including what they are expected to do based upon the advanced scheduling and what their actual status is.

Table 4.4: Machine Data Library: machine availability

| **machine_name** | **timestamp** | **sched_status** | **current_status** |
|---|---|---|---|
| MAZAK-M7303290458 | 2017-05-04T20:47:37Z | scheduled to be idle | Machine is OFF. |
| HAAS-VF3 | 2017-05-05T16:00:00 | scheduled to be idle | Machine is OFF. |
| HAAS-VF2 | 2017-04-26T13:21:02.293 | scheduled to be idle | Machine is ON and READY. |
| BENCHMAN-4000 | 2017-05-05T12:01:023 | scheduled to be idle | Machine is OFF. |

Table 4.5: Machine Data Library: process capability

| process | 尺寸公差<br>diametric_tol | 位置公差<br>positional_tol |
|---|---|---|
| drill | 0.2 | 0.13 |
| bore | 0.05 | 0.05 |
| mill | 0.0762 | 0.0762 |
| ream | 0.06 | 0.13 |

Table 4.5 shows the process capability table where the process capabilities are listed. Presently the processes for which capabilities are defined are the hole making operations like drilling, boring and reaming 铰孔; and simple rough and finish milling. The tolerances 公差 must be defined for each feature machined in the setup sheet either manually or while generating the setup sheet. Table 4.6 shows the materials table containing the material list and their respective densities for mass related calculations in this research.

Table 4.6: Machine Data Library: 材料表 Materials

| Material | density |
|---|---|
| **stainless steel** | 7800 |
| **Aluminum** | 2700 |
| **Titanium** | 4500 |
| **Iron** | 7874 |
| **Copper** | 8950 |

加工状态

Table 4.7: Machine Data Library: process status log

| timestamp | status | setup sheet | material | machine | availability | feedback |
|---|---|---|---|---|---|---|
| 2017-02-18T17:40:20.150000 | DOME | https://s3.amazonaws.com/ncsudome/setupsheet3.html | stainless steel | None | None | None |
| 2017-02-18T17:40:25.014000 | AWS | None | None | MAZAK-M7303290458, HAAS-VF3, can machine the part. | MAZAK-M7303290458 is available now. HAAS-VF3 is available now. | MAZAK-M7303290458 will provide better finish in better time. |

Table 4.7 shows the process status log table. The purpose of this table is to keep a log of all the runs made from the front end. As seen in the Table 4.7, it updates the status to 'DOME' in the table when the user runs the application from the front end. DOME is the front-end DMC application. Once the inputs are received the status changes to 'AWS' where it is processed and sent back to the front end. It also stores the key output feedbacks if any, reference to the location of the setup sheet on Amazon S3, material name and timestamp.

All the tables mentioned above together serve as a reservoir of machine specification data. This is used as the basis for comparison when an input setup sheet is received. If the machine specifications meet the requirements of the part to be built, the machine is a viable option for the user. In the next section, various possible outcomes are discussed at length.

## 4.3. Result

### 4.3.1. DMC App: Identification and Verification of Machine Capabilities for a Given Part

The DMC application is quite similar to the applications demonstrated in section 3.3.1. Like the previous DMC apps, this app's front end is also scripted in HTML and CSS. As is seen in the Figure 4.4, the inputs require the user to give in the IP Address, username, database name, port and password for authentication purpose. The input parameters connect the user to a specific Machine Data Library. The second set of inputs require the user to specify the http location of the setup sheet file of the part that is to be machined and the material of the corresponding part. For the purposes of demonstration, we have used the setup sheet made available through Autodesk Fusion CAM.

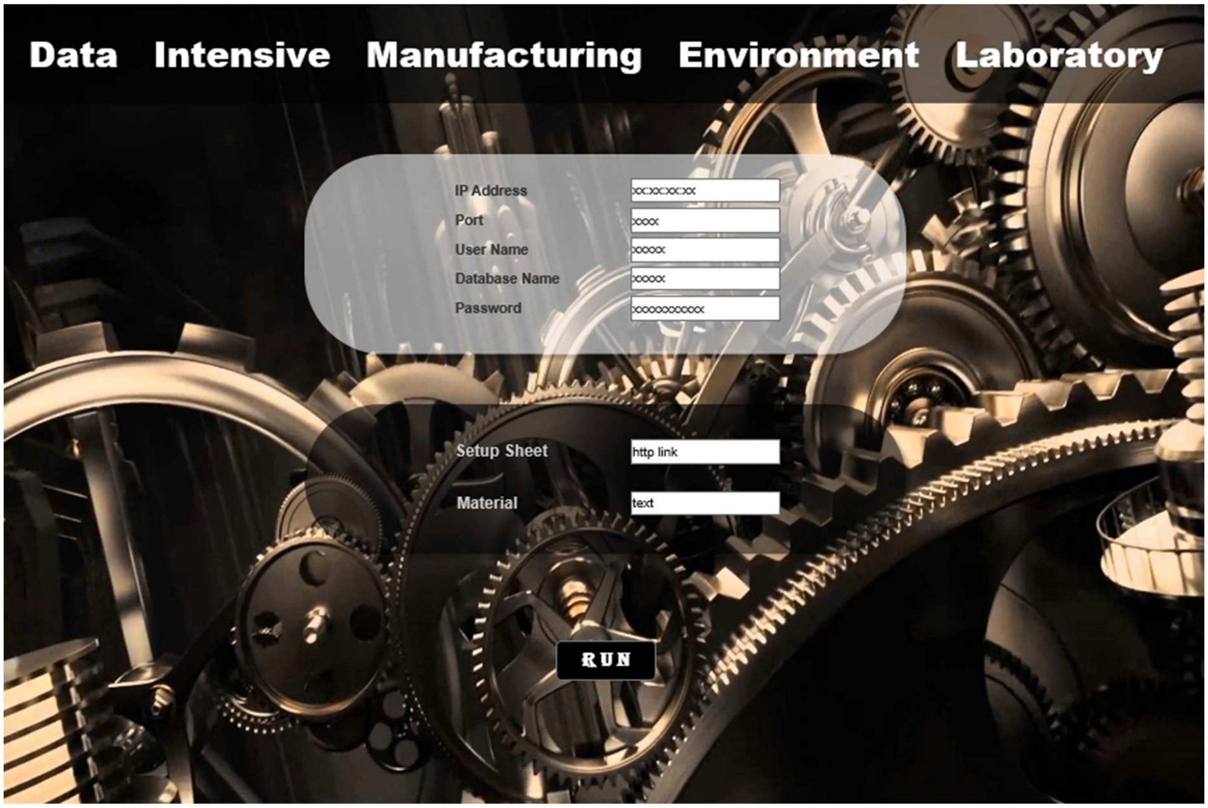



Figure 4.4: DMC App: Machine Capability Check: Default Input Interface

Figure 4.5 on the other hand shows an example of how an Autodesk Fusion setup sheet looks like. The setup sheet is basically divided into two sections. First section, 'Job', lists out the stock dimensions and the part dimensions with an image of the stock/part on the right. This gives an idea about the amount of material that would be used if this part is machined. The second section has two important sets of information. First, an overview of the CAM operations including information like cycle time, no. of operations, maximum spindle speed and maximum rapid feed among others. Second, same information but for each feature operation along with the tool used and its properties can be extracted out. It is important to note that the tolerance in many cases if undefined might be 0, therefore, one must either define it during CAD/CAM process or manually after the setup sheet is generated.

JOB DESCRIPTION: Setup1
DOCUMENT PATH: Part v9

**Setup**

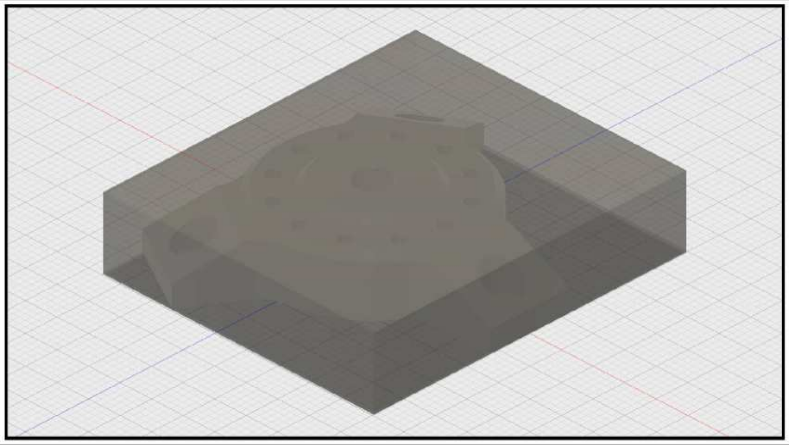
WCS: #0

STOCK:
DX: 4.291in
DY: 4.955in
DZ: 1.25in

PART:
DX: 4.291in
DY: 4.955in
DZ: 1.125in

STOCK LOWER IN WCS #0:
X: 0in
Y: 0in
Z: -1.25in

STOCK UPPER IN WCS #0:
X: 4.291in
Y: 4.955in
Z: 0in



**Total**

NUMBER OF OPERATIONS: 7
NUMBER OF TOOLS: 7
TOOLS: **T16 T17 T18 T19 T20 T35 T36**
MAXIMUM Z: 0.591in
MINIMUM Z: -1.435in
MAXIMUM FEEDRATE: 623.35199in/min
MAXIMUM SPINDLE SPEED: 16000rpm
CUTTING DISTANCE: 2339.508in
RAPID DISTANCE: 622.911in
ESTIMATED CYCLE TIME: 23m:37s

Operation 1/7
DESCRIPTION: Adaptive12
STRATEGY: Adaptive
WCS: #0
TOLERANCE: 0.004in
STOCK TO LEAVE: 0in
MAXIMUM STEPDOWN: 0.394in
OPTIMAL LOAD: 0.3in
LOAD DEVIATION: 0.03in

MAXIMUM Z: 0.591in
MINIMUM Z: -1.329in
MAXIMUM SPINDLE SPEED: 16000rpm
MAXIMUM FEEDRATE: 623.35199in/min
CUTTING DISTANCE: 549.491in
RAPID DISTANCE: 426.708in
ESTIMATED CYCLE TIME: 1m:5s (4.6%)
COOLANT: Flood

**T36** D36 L36
TYPE: bullnose end mill
DIAMETER: 0.75in
CORNER RADIUS: 0.06in
LENGTH: 2.9in
FLUTES: 5

Operation 2/7
DESCRIPTION: Drill2
STRATEGY: Drilling
WCS: #0
TOLERANCE: 0in

MAXIMUM Z: 0.591in
MINIMUM Z: -1.389in
MAXIMUM SPINDLE SPEED: 5668rpm
MAXIMUM FEEDRATE: 33.4706in/min
CUTTING DISTANCE: 17.919in
RAPID DISTANCE: 78.693in
ESTIMATED CYCLE TIME: 33s (2.3%)
COOLANT: Flood

**T16** D16 L16
TYPE: drill
DIAMETER: 0.199in
TIP ANGLE: 118°
LENGTH: 1.393in
FLUTES: 1
DESCRIPTION: #8

Figure 4.5: Sample Input Setup Sheet File

Once the inputs are filled in and the ‘Run’ button is clicked, AWS comes into play. The job of Amazon EC2 is to run scripts to perform all the necessary operations to yield results. Since the setup sheet is in html file format, the main script ‘final.py’ running on EC2, converts the html file to xml file format. This makes it easier to navigate through the file. Second, it checks the units defined in the setup sheet so that all the parameters can be mapped accordingly. Then it starts with creating and filling a dictionary for the input parameters which is used to compare the parameters with the basis parameters in the Machine Data Library. Following are the parameters that are compared and validated by the script:

- Range of stock: To check if the given part falls within the maximum allowable dimensions of parts defined by the machines.
- Maximum rapid feed
- Maximum spindle speed
- Estimated cycle time
- Machine type: Mill/Turn
- Part diameter and length if turning
- Weight of the part
- Material capability
- Maximum tool diameter
- Minimum tolerance capability
- Specific feature tolerance capability
- Machine Availability
- If multiple machines available, priority

There is another script running simultaneously alongside this script named 'machine_avail_update.py'. This script updates the scheduled and actual status of the machines in the library. This helps the main script in determining the machine availability and prioritizing accordingly. Once all the parameters are verified, the main script writes to the 'process status log' table, mentioned in the previous section, mentioning the result for the given set of inputs. These results could vary depending upon the input parameters. In Figure 4.6, a successful set of inputs can be seen. It states that machines 'MAZAK-M7303290458' and 'HAAS-VF3' are capable of building the part. Also, based upon availability and capability the feedback suggestions are given so that the user can make the decision on its own.

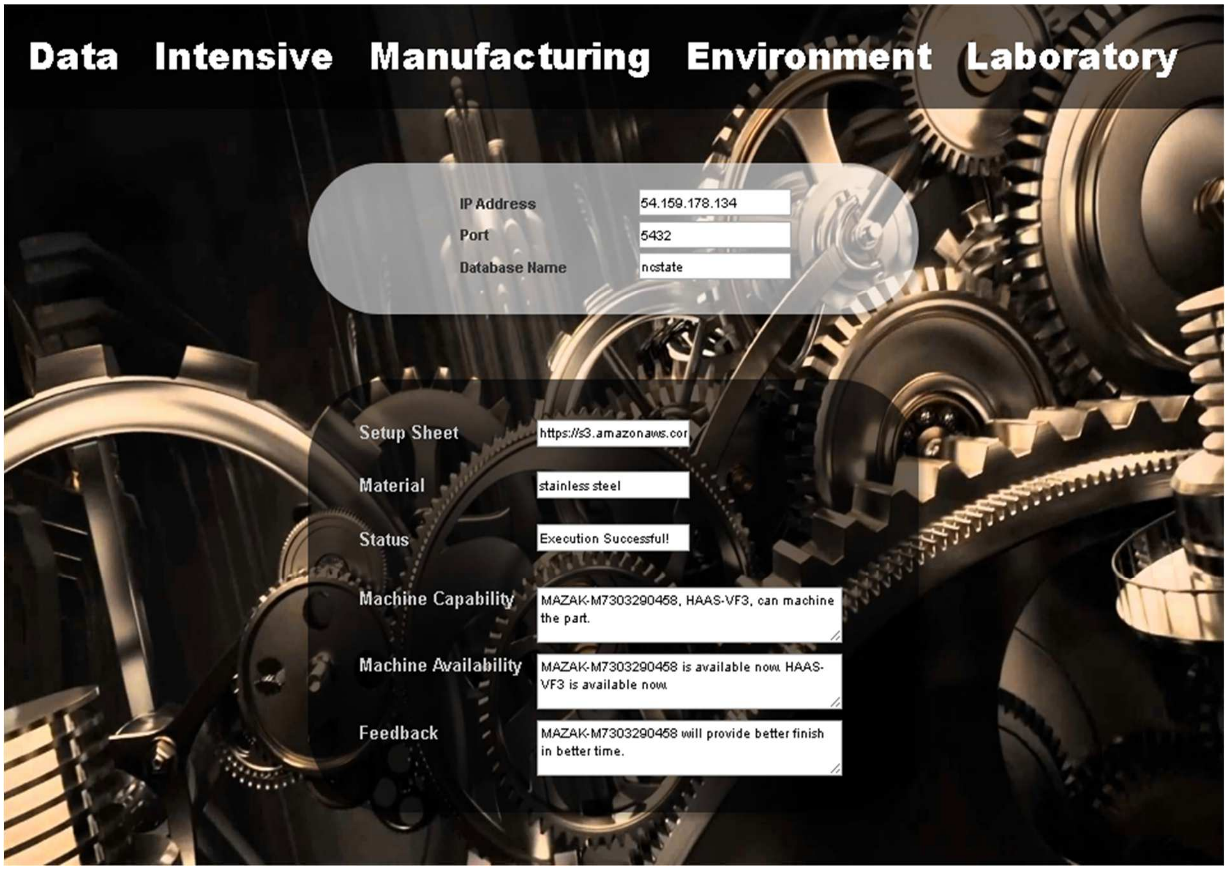



Figure 4.6: DMC App: Machine Capability Check: Sample Output 1

Figure 4.7 shows another possibility where the input tolerances of specific parts cannot be met by any of the machines. It informs the user that no machine can build that tolerance and then points out which particular feature is the bottleneck. This helps the user in either checking whether the set tolerances were correct or working on the part with a lower tolerance, or perhaps move to more sophisticated machines. Similarly, there are possibilities in where if the input parameters are not correct or not in the format expected, the app would show the error and ask the user to correct itself.

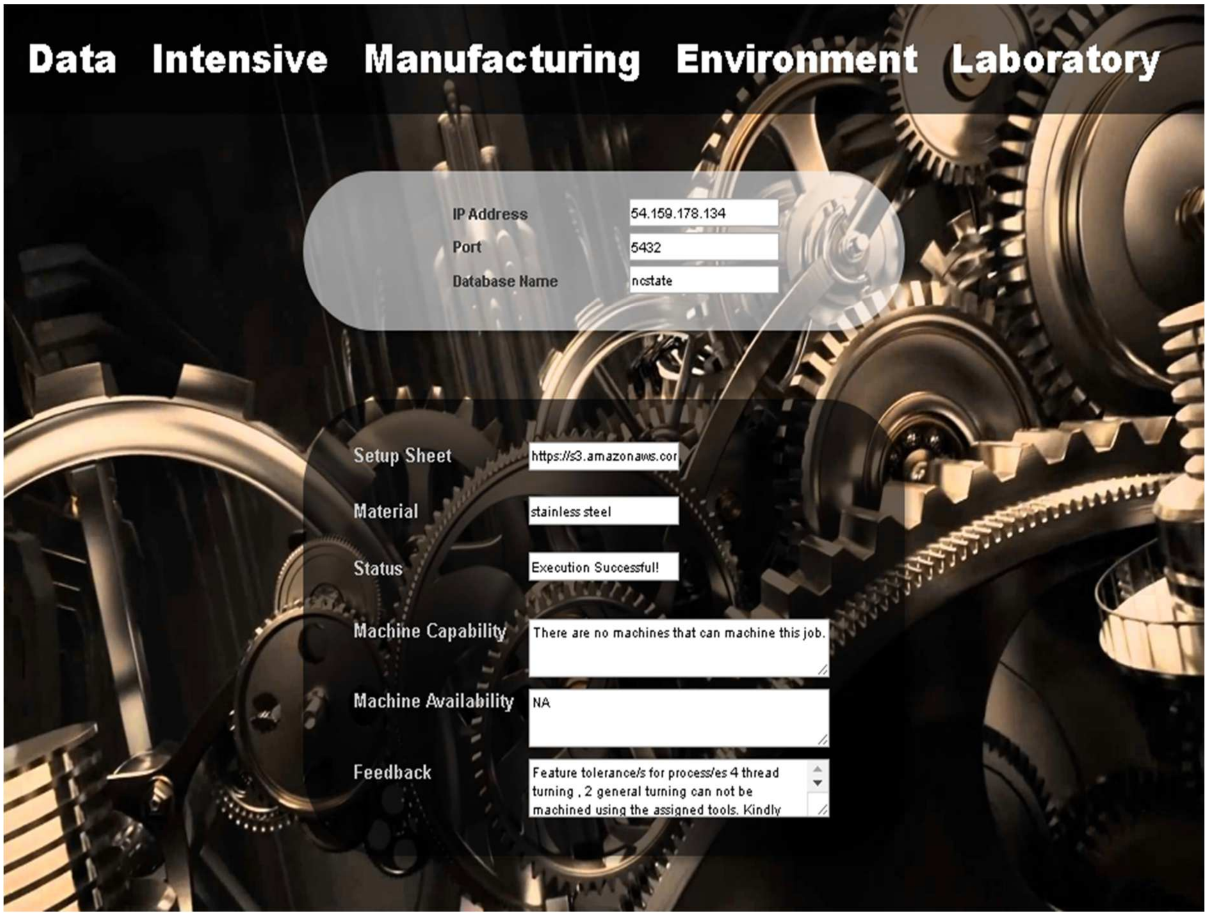


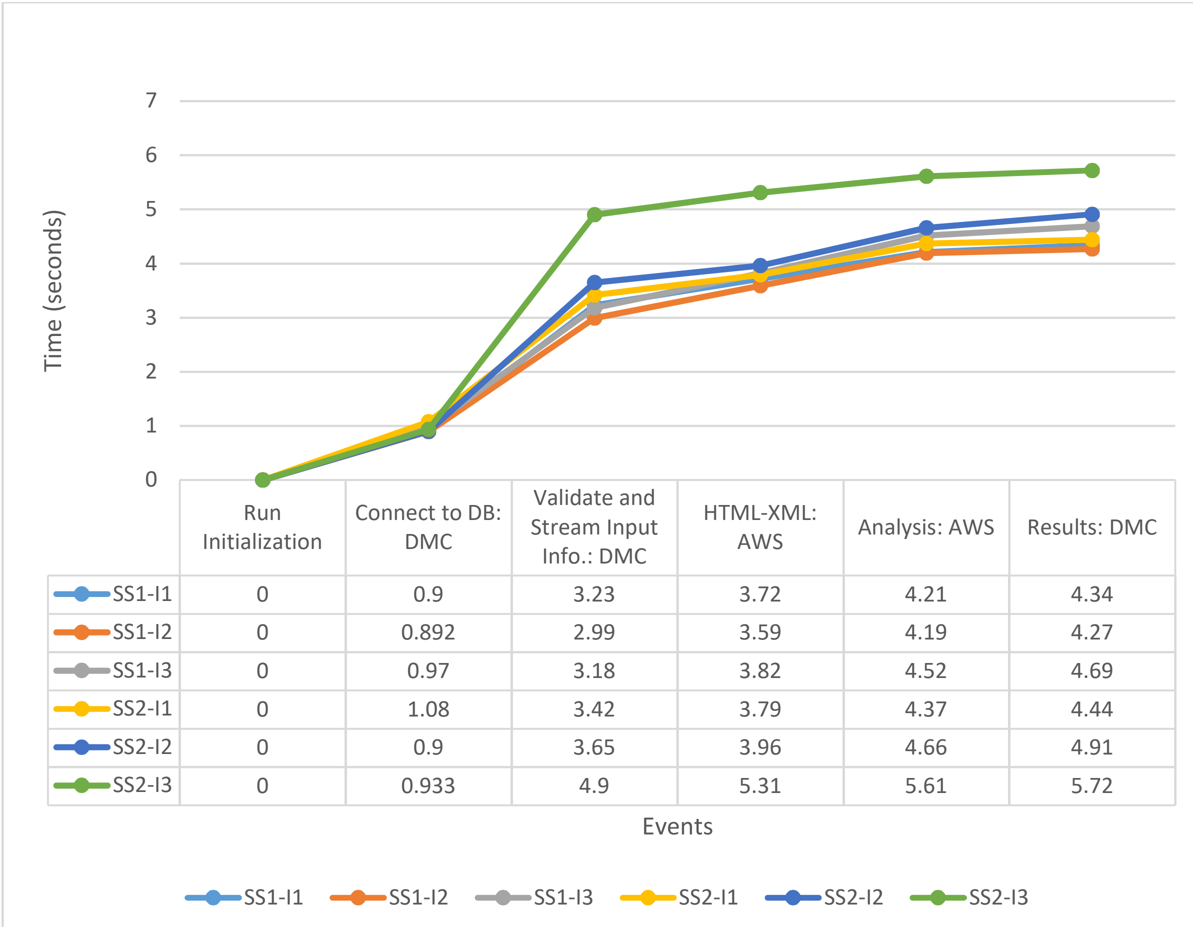

Figure 4.7: DMC App: Machine Capability Check: Sample Output 2

计算时间研究

### 4.3.2. Computational Time Study

This study was conducted to analyze data transfer, compute speeds and how much time does this verification process take. Therefore we ran a time study to analyze which steps in the process created a bottleneck in displaying results to the user. Three different setup sheets were run through the system. For each setup sheet, run thrice, we calculated the amount of time each step in the process takes. In the Figure 4.8, an event vs time 2-D plot is seen. This plot helps in figuring out the process within the information flow cycle that takes significantly more time. The plot shows three runs for two different setup sheets. The notation 'SS1-I1' refers to the first run of the first setup sheet. The two setup sheets used are for a turning and milling part in that order. Even though no different trend is observed due to different setup sheets, the common observations are quite interesting.

The two processes at the beginning consume considerably high computation time than the others. The first event is when the user hits the 'Run' button which is given the time value of zero seconds. For the second event, connecting to the Amazon RDS Postgres database, it takes roughly 1 sec of time. These kinds of connections usually do take a significant amount of time. The third event, which takes the most time, is where the input data is validated and streamed to the process status log table to update the status of the interaction. From here, AWS takes over. It is at this stage that DMC gives a 2-sec lag. Perhaps in the future versions of the DMC front-end, this issue will be addressed. The rest of the processes take reasonable amount of time. Although, the process of converting the html setup sheet to an xml file format does take about 0.4 sec on an average which is avoidable. As mentioned previously, the xml file format is chosen as it makes it easier to navigate through the setup sheet data.




| | Run Initialization | Connect to DB: DMC | Validate and Stream Input Info.: DMC | HTML-XML: AWS | Analysis: AWS | Results: DMC |
|---|---|---|---|---|---|---|
| SS1-I1 | 0 | 0.9 | 3.23 | 3.72 | 4.21 | 4.34 |
| SS1-I2 | 0 | 0.892 | 2.99 | 3.59 | 4.19 | 4.27 |
| SS1-I3 | 0 | 0.97 | 3.18 | 3.82 | 4.52 | 4.69 |
| SS2-I1 | 0 | 1.08 | 3.42 | 3.79 | 4.37 | 4.44 |
| SS2-I2 | 0 | 0.9 | 3.65 | 3.96 | 4.66 | 4.91 |
| SS2-I3 | 0 | 0.933 | 4.9 | 5.31 | 5.61 | 5.72 |

Figure 4.8: Computation Time Study Plot

## 4.4. Evaluation of Approach

When it comes to designing such a cyber-physical setup, there are many challenges that inspire the engineer. This section evaluates the current approach and discusses the issues addressed.

Today, while there are plenty of ways to validate whether a part can be built on a machine, it requires the customer to share a lot of data. This data could be sensitive data because of which such validation is done locally. There is lack of middleware architectures with necessary security layers which help the user to keep the data discreet. The app discussed in this chapter demonstrates the use of a software middleware - a broker that mediates data between the two parties. It presents a possible solution where the user does not have to share

不需要CAD文件和G代码

their CAD files or G-Code files with multiple manufacturers. Only the setup sheet is shared with the broker, a neutral third party which can extract relevant information. This setup sheet contains an overview of the product which still is safe within the AWS architecture. With this

setup 可以纵览零件文件

kind of data many levels of validation can be performed as mentioned in the previous section without exposing any data to either party.

Whereas the app helps the user keep the data discreet, it also helps to keep the machine specification data of the manufacturers safe and separate from the customers. This way both parties get to hide their data and at the same time reap benefits from the app. Therefore, such a middleware architecture is key to automate this process of matching making between user requirements with that manufacturer's technical capabilities. Without which, the customer must go through layers of interaction before being able to identify the right manufacturers who can complete the job.

Even though this app serves as a good medium to achieve a two-way secure interaction between the front end and the machine end, there is a need to create a more robust and comprehensive system in and around the middleware architecture. The time study showed a high consumption of time during html to xml conversion of setup sheet. This problem can be addressed by optimizing the existing code and maybe also using a more robust OS in EC2 with better system specifications. Also, during reporting back the values to the DMC frontend, a lot of time is consumed by DMC. Perhaps a newer version of DMC will help solve this issue and try out better messaging protocols that can help speed up the communication between servers. The interaction at the machine backend between the machine and the database must be made as automated as possible. Presently the tool information is manually entered. This can be automated by using sensors for tool detection. These problems must be addressed before a comprehensive secure two-way interaction can be established.

### 4.5. Chapter Summary

A solution to verify instructions and perform a two-way interaction across a secure middleware from cloud front-end to the physical machine is discussed in this chapter. Manufacturers can now save the machine specifications on the cloud platform which eliminates the issues of data storage as well as security. Together from the manufacturing setups across the world, a large pool of such machine specification data can be envisioned in a big Machine Data Library. DMC app demonstrated in this chapter serves as a link between the manufacturer and the customer. With the help of the app this interaction can be further made completely automated. The DMC app takes in inputs from the user and runs the information through the Machine Data Library to find a possible machine match for the part. The user then can contact the manufacturer and get the part made without the hassle of going around and asking different manufacturers all by itself.

The middleware architecture uses three Amazon Web Services: EC2, S3 and RDS which act as the brain, body and blood of this system. Amazon EC2 runs all the python scripts and is the center of interaction between all the other services as well as the back-end and the front-end. Amazon S3 is a file storage service which stores all the important files and the input setup sheet files. Amazon RDS serves as the Machine Data Library where all the machine specification data is stored. There are many other cloud computing platforms like that of Microsoft Azure, NASA's Open Nebula, Rackspace's Cloud Files and IBM Smart Cloud among many others. But Amazon Web Services offers very reliable, scalable, and inexpensive cloud computing services.

The time study suggested high time consumption during html-xml conversion and reporting back the outputs. There is a fixed delay at the DMC front-end which perhaps will be eliminated from the future versions. The html-xml conversion, even though makes it easier to navigate through the setup sheet, should be worked upon in order to achieve optimal response time for the app discussed in this chapter.

# CHAPTER 5

# CONCLUSIONS AND FUTURE WORK

## 5.1. Enabling the Digital Thread

In this thesis, a solution to interface machines on a shop floor towards a digital factory solution through a middleware architecture has been discussed. By being able to connect manufacturing machines on the shop-floor, a significant step is taken towards realizing the digital thread in product lifecycle management. The digital data can be collected to enable the digital thread throughout a product lifecycle starting from idea initiation to end of product life. The data extracted from the machines provide valuable in-process data that can serve to improve upon existing product designs. It can even help in improving future designs. It is also shown how streaming data from the machines can also be used to build third-party applications to enable the digital factory.

The first research objective, discussed in Chapter 3, throws light on how granular streaming data from the machines can be efficiently stored, archived and retrieved. The solution demonstrates how incoming streaming data can be stored into a next generation structured SQL database, such as PostgreSQL. This contrasts with using traditional SQL type databases such as Access, ORACLE or MySQL. PostgreSQL, an open source database is well suited for fast writing and reading of streaming time-series based data. This pilot solution can be replicated for all machines on the shop-floor. Machine monitoring apps discussed here are deployed to the DMC, opening the possibility of integrating a variety of machines on the shop-floor. DMC apps discussed here calculates various parameters like machine utilization and overall equipment effectiveness (OEE), and shows various sensor data plots. A prototype application to verify a part’s geometry by comparing original data with the coordinates obtained from the actual machine coordinates is also demonstrated. This will help to optimize part probing during machining and provide an initial verification pathway for part tested. An application to monitor the machine parameters using LabVIEW is also discussed. This is a primary approach which shows machine monitoring and axial data visualization. Even though most of these apps cannot be used on an industrial scale yet, they are good prototypes demonstrating the manner in which machine data can be utilized. The first half of this thesis essentially makes a successful attempt to bridge the machine and the digital platform at the back-end and explores manufacturing intelligence at the front.

The second half of the thesis primarily focusses on the middleware architecture involved in a digital-cloud manufacturing setup. It gives details on how the machine back-end and the user front-end interact through the middleware with the help of a DMC application. The middleware residing in the Amazon Web Services (AWS) consists of three important services: EC2 (the brain), RDS (PostgreSQL DB-the machine data library) and S3 (file storage). The application demonstrates how a customer can verify part specifications w.r.t the machine specifications. AWS EC2 contains the scripts which continuously runs in the cloud waiting for user inputs, performs necessary verifications through live data contains in the RDS machine data library. Currently, the machine data library consists of a few machines with a limited number of specifications which can be expanded upon. The input file used is a setup sheet generated through Autodesk Fusion. The specificity of which can be eliminated if there

Autodesk Fshion

is a way to figure out secure extraction of part-data directly from the CAM software. There is also a time study analysis which gives us an idea as to which part of the data interaction cycle consumes the most time. Establishing the authentication between the middleware and the DMC front end takes the most time which can definitely be improved upon. The overall idea here is to define a middleware architecture which provides a safe and efficient platform for interaction between the customers and the manufacturers.

## 5.2. Technical Contributions

This thesis has made several technical contributions to the digital manufacturing community. The work perform primarily lays down the infrastructure for a digital factory. The code for each of the various software adaptors built for the machines are uploaded to the GitHub account. The link for the corresponding GitHub repository is listed below: https://github.com/ssingh23/shaurabh_thesis.

The specific technical contribution are as follows:

1) A low-cost hardware solution to gather machine data from the machine's hardware controllers. Both an Intel NUC and a Raspberry PI has been used as low-cost computing solution to gather data from the machines.

2) Data has been formatted to create MT-CONNECT agent platforms for both the benchman and the HAAS VF series of machines. Since HAAS machines are widely used in US manufacturing, the adaptor and agent can be adopted and extended by the community. Specific scripts were written in the Python program language were written and stored in the computing hardware which constantly streams its data to the database.

3) Demonstrated the use of the next generation of SQL databases that support multi-concurrency that can handle time series based data - PostgreSQL. This database is currently storing data collected from machines in the ISE MFG laboratories. A copy of the database also exists in the Azure Cloud and the data made available to the community through the Digital Manufacturing Commons.

4) A series of 5 manufacturing apps were demonstrated and made available in the Digital Manufacturing Commons marketplace. Each app demonstrated various examples of how to interface with machine generated data. The apps are made available to the DMDII community at - https://portal.opendmc.org/marketplace.php#/home?product=services.

5) A middleware architecture is demonstrated using AWS infrastructure. The middleware allows direct data collection from the machines while interacting with users at the front-end in a cloud manufacturing marketplace. The middleware acts as the bridge between two-way interaction between the front end and the back end while being safe and a secure neutral third party. Customers must find the right manufacturer to build their parts. To achieve this, the customers must find the manufacturer with right set of tools and machines with the right specifications, verifying and vetting them before prototype or production orders can be sent. Many customers though, are not willing to share detailed part specifications before the deal is finalized and at the same time the manufacturers are also hesitant to share their machine utilization and capacity information with their clients. The middleware architecture discussed here addresses this issue. The middleware stores all the machine specifications and machine updates from the manufacturer and the setup-sheet

from the customer. It does it without having to share one party's information with the other. This AWS middleware is extremely safe and efficient and the security can be further strengthening by using Identity and Access Management (IAM) service of the AWS.

### 5.3. Future Work

Cloud manufacturing is still an expanding field with significant advancements in hardware and software computing. Two direct future work that can be expanded on based on the current work are as follows:

**Connecting Various Types of Machines to an Integrated Network:** The biggest challenge remains to be the installation of a successful digital factory setup on a shop floor predominantly occupied by machines incapable of interacting with the outside world. Most of which still are old legacy machines. The immediate future work is digitally connecting as many machines as possible in the ISE lab to the NC State network. This would require building customized MT-CONNECT adapters for each type of machine. This work can also be extended to additive manufacturing machines in the future. Future version of the MTCONNECT standard will allow data transfer from 3D printing machines and metrology machines, thereby broadening the scope of this work.

**Dynamic Production Scheduling and Instant Pricing Models:** If we can achieve the above integration of machines, then the work described in the second research objective can be extended to include more automated services. Machine schedules stored in a table in the machine data library are manually input at present. Perhaps, a forecasting algorithm could be developed or an existing one could be used which would assess the scheduling history and build models to predict capacity into the future. Unutilized capacity on job shop floors can be sold to allow product prototypes to be built at much lower prices than currently practiced. Such streaming data through middleware can even inspire new business models around prototype manufacturing.

## REFERENCES

[1] Choi S., Jun C., Zhao W. B., Noh S. D., "Digital Manufacturing in Smart Manufacturing Systems: Contribution, Barriers, and Future Directions", Advances in Production Management Systems: Innovative Production Management Towards Sustainable Growth, Vol. 460, 2015

[2] Choi S., Kim B., Noh S., "A diagnosis and evaluation method for strategic planning and systematic design of a virtual factory in smart manufacturing systems.", International Journal of Precision Engineering and Manufacturing, 2015.

[3] Cotteleer M., Holdowsky J., Mahto M., "The 3D opportunity primer.", On: Deloitte University Press, 2014.

[4] Phani Kumari Paritala, Shalini Manchikatla, Prasad KDV Yarlagadda, "Digital Manufacturing- Applications Past, Current, and Future Trends", Procedia Engineering, Volume 174, 2017.

[5] Rainer Stark, Simon Kind, Sebastian Neumeyer, "Innovations in digital modelling for next generation manufacturing system design", CIRP Annals - Manufacturing Technology, 2017.

[6] Moneer Helu, Thomas Hedberg, Allison Barnard Feeney, "Reference architecture to integrate heterogeneous manufacturing systems for the digital thread", CIRP Journal of Manufacturing Science and Technology, 2017.

[7] Jonathan Downey, Denis O'Sullivan, Miroslaw Nejmen, Sebastian Bombinski, Paul O'Leary, Ramesh Raghavendra, Krzysztof Jemielniak, "Real time monitoring of the

CNC process in a production environment- the data collection & analysis phase", Procedia CIRP, Vol. 41, 2016.

[8] João A. Duro, Julian A. Padget, Chris R. Bowen, H. Alicia Kim, Aydin Nassehi, "Multi-sensor data fusion framework for CNC machining monitoring", Journal of Mechanical Systems and Signal Processing, 2016.

[9] Andrea Bonci, Massimiliano Pirani, Sauro Longhi, "A database-centric approach for the modeling, simulation and control of cyber-physical systems in the factory of the future.", IFAC-PapersOnLine, Vol. 49, Issue 12, 2016.

[10] Schulz WL, Nelson BG, Felker DK, Durant TJ, Torres R, "Evaluation of relational and NoSQL database architectures to manage genomic annotations", J Biomed Inform, 2016.

[11] Peng Yue, Zhenyu Tan, "GIS Databases and NoSQL Databases", Reference Module in Earth Systems and Environmental Sciences, 2017.

[12] Anatoly Sorokin, Gene Selkov, Igor Goryanin, "A user-defined data type for the storage of time series data allowing efficient similarity screening", European Journal of Pharmaceutical Sciences, Vol. 46, Issue 4, 2012.

[13] D Mourtzis, M Doukas, C Vandera, "Mobile apps for product customisation and design of manufacturing networks", Manufacturing Letters, 2014.

[14] LI Bo-hu, ZHANG Lin, WANG Shi-long, TAO Fei, CAO Jun-wei, JIANG Xiao-dan, SONG Xiao, CHAI Xu-dong, "Cloud manufacturing: a new service-oriented networked manufacturing model", Computer Integrated Manufacturing Systems, 2010.

[15] Hassan, Qusay, "Demystifying Cloud Computing", The Journal of Defense Software Engineering, 2011.

[16] Vaibhav A Gandhi, C K Kumbharana, “Comparative study of Amazon EC2 and Microsoft Azure cloud architecture”, International Journal of Advanced Networking Applications, 2014.

[17] T.Madhuri, P.Sowjanya, “Microsoft Azure v/s Amazon AWS Cloud Services: A Comparative Study”, International Journal of Innovative Research in Science, Engineering and Technology, 2016.

[18] Radu F. Babiceanua, Remzi Seker, “Big Data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook”, Computers in Industry, 2016.

[19] Xiaoqing F. Liu, Md Rakib Shahriar, S.M. Nahian Al Sunny, Ming C. Leu, Liwen Hu, “Cyber-physical manufacturing cloud: Architecture, virtualization, communication, and testbed”, Journal of Manufacturing Systems, 2017.

[20] Dazhong Wu, Xi Liu, Steve Hebert, Wolfgang Gentzsch, Janis Terpenny, “Democratizing digital design and manufacturing using high performance cloud computing: Performance evaluation and benchmarking”, Journal of Manufacturing Systems, Vol. 43, Part 2, 2017.

[21] Yongkui Liu, Xun Xu, Lin Zhang, Long Wang, Ray Zhang, “Workload-based multi-task scheduling in cloud manufacturing”, Robotics and Computer-Integrated Manufacturing, Volume 45, Issue C, 2017.

[22] Hehua Yan, Qingsong Hua, Yingying Wang, Wenguo Wei, Muhammad Imran, “Cloud robotics in Smart Manufacturing Environments: Challenges and countermeasures”, Computers & Electrical Engineering, 2017.

[23] David Golightly, Sarah Sharples, Harshada Patel, Svetan Ratchev, “Manufacturing in the cloud: A human factors perspective”, International Journal of Industrial Ergonomics, Vol. 55, 2016.

[24] Xiuhong Chen, Xianglei Huang, Chaoyi Jiao, Mark G. Flanner, Todd Raeker, Brock Palen, “Running climate model on a commercial cloud computing environment: A case study using Community Earth System Model (CESM) on Amazon AWS”, Computers & Geosciences, Vol. 98, 2017.

[25] Yingfeng Zhang, Fei Tao, “Chapter 5 – Cloud Computing-Based Manufacturing Resources Configuration Method”, Optimization of Manufacturing Systems Using the Internet of Things, 2017.

[26] Xi Vincent Wang, Lihui Wang, Abdullah Mohammed, Mohammad Givehchi, “Ubiquitous manufacturing system based on Cloud: A robotics application”, Robotics and Computer-Integrated Manufacturing, 2017.

[27] Beata Mrugalska, Magdalena K. Wyrwicka, “Towards Lean Production in Industry 4.0”, Procedia Engineering, Volume 182, 2017.

[28] J.L. Michaloski, Byeongeon Lee, Frederick Marion Proctor, Sidney Ly, “Quantifying the Performance of MT-Connect in a Distributed Manufacturing Environment”, ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2009.

[29] H. Atluru, Amit Deshpande, “Data to Information: Can MTConnect Deliver the Promise?”, NAMRC, 2009.

[30] Pei Lei, Lianyu Zheng, Chao Li, Xichun Li, “MTConnect Enabled Interoperable Monitoring System for Finish Machining Assembly Interfaces of Large-scale Components”, Procedia CIRP, Vol. 56, 2016.

[31] B. E. Lee, J.L. Michaloski, Frederick Marion Proctor, Sid Venkatesh, N. Bengtsson, “MTConnect-Based Kaizen for Machine Tool Processes”, ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2010.

[32] Ben Edrington, Bingyan Zhao, Adam Hansel, Masahiko Mori, Makoto Fujishima, “Machine Monitoring System Based on MTConnect Technology”, Procedia CIRP, Volume 22, 2014.

[33] Martin Hardwick, “Roadmap for Deploying Semantic GD&T in Manufacturing”, Procedia CIRP, Vol. 52, 2016.

[34] Jay Lee, Hossein Davari Ardakani, Shanhu Yang, Behrad Bagheri, “Industrial Big Data Analytics and Cyber-Physical Systems for Future Maintenance & Service Innovation”, Procedia CIRP, Vol. 38, 2015.

[35] L.E.S. Oliveira, A.J. Álvares, “Axiomatic Design Applied to the Development of a System for Monitoring and Teleoperation of a CNC Machine through the Internet”, Procedia CIRP, Vol. 53, 2016.

[36] Athulan Vijayaraghavan, Will Sobel, Armando Fox, David Dornfeld, Paul Warndorf, “Improving Machine Tool Interoperability using Standardized Interface Protocols: MTConnect™”, 2008 International Symposium on Flexible Automation, Atlanta, GA, 2008.

[37] "Amazon Web Services" [Online]. Available: https://aws.amazon.com/ [Accessed 1st March 2017].

[38] "Types of Cloud Computing" [Online]. Available: https://aws.amazon.com/types-of-cloud-computing/ [Accessed 1st March 2017].

[39] "Complete Digital Factory Integration and the IIoT" [Online]. Available: http://www.cisco.com/c/dam/en_us/solutions/industries/docs/manufacturing/smart-box.pdf [Accessed 1st March 2017].

[40] "GE on AWS" [Online]. Available: https://aws.amazon.com/solutions/case-studies/ge/ [Accessed 1st March 2017].

[41] Dave Edstrom, "Getting Started with MTConnect: Writing Client Applications. A High-Level Architecture Overview and Tutorial for Software Application Developers" [Online]. Available: https://www.automation.com/pdf_articles/gettingstartedwithmtconnectwritingclientsrevisionjuly2013.pdf [Accessed 1st March 2017].

[42] "Unified Architecture" [Online]. Available: https://opcfoundation.org/about/opc-technologies/opc-ua/ [Accessed 1st March 2017]

[43] Craig Resnick, "MQTT is Gaining Ground as a Messaging Protocol for IIoT" [Online]. Available: https://industrial-iot.com/2016/10/mqtt-protocol-for-iiot/ [Accessed 1st March 2017].

[44] "One click to deploy a RabbitMQ cluster handling over 1 million msg/sec" [Online]. Available: https://cloudplatform.googleblog.com/2014/06/rabbitmq-on-google-compute-engine.html [Accessed 1st March 2017].

[45] D Caro, "Industrial data communications protocols and application layers", Woodhead Publishing Series in Electronic and Optical Materials, 2016.

[46] "Digital Manufacturing Commons" [Online]. Available: http://www.projectdmc.org/ [Accessed 1st March 2017].

[47] "MTConnect" [Online]. Available: http://www.mtconnect.org/ [Accessed 1st March 2017].