Graph analysis of Fog Computing Systems for Industry 4.0

Nandor Verba, Kuo-Ming Chao, Anne James, Jacek Lewandowski, Xiang Fei
Faculty of Engineering, Environment and Computing
Coventry University
Coventry, United Kingdom
verban, csx240, csx118, aa6910, aa5861@coventry.ac.uk

Chen-Fang Tsai
Department of Industrial Management and Enterprise Information, Aletheia University
New Taipei City, Taiwan
au1204@mail.au.edu.tw

***Abstract*— Increased adoption of Fog Computing concepts into Cyber Physical Systems (CPS) is a driving force for implementing Industry 4.0. The modern industrial environment focuses on providing a flexible factory floor that suits the needs of modern manufacturing through the reduction of downtimes, reconfiguration times, adoption of new technologies and the increase of its production capabilities and rates. Fog Computing through CPS aims to provide a flexible orchestration and management platform that can meet the needs of this emerging industry model.**

Proposals on Fog Computing platform and Software Defined Networks (SDN) for Industry allow for resource virtualization and access throughout the system enabling large composite application systems to be deployed on multiple nodes. The increase of reliability, redundancy and runtime parameters as well as the reduction of costs in such systems are of key interest to Industry and researchers as well. The development of optimization algorithms and methods is made difficult by the complexity of such systems and the lack of real-world data on fog systems resulting in algorithms that are not being designed for real world scenarios. We propose a set of use-case scenarios based on our Industrial partner that we analyze to determine the graph based parameters of the system that allows us to scale and generate a more realistic testing scenario for future optimization attempts as well as determine the nature of such systems in comparison to other networks types. To show the differences between these scenarios and our real-world use-case we have selected a set of key graph characteristics based on which we analyze and compare the resulting graphs from the systems.

Keywords—Fog Computing; CPS; Industry 4.0; Graph Analysis;

I. INTRODUCTION

Cyber Physical Systems (CPS) implementation in Industrial Environments has emerged from an increased use of wireless medium based sensor and actuator technologies that have increasingly larger processing and communication capabilities. Together with concepts from Internet of Things (IoT) the requirements of Industry 4.0 were established. Proposals such as the framework in [1] and the business process centric approach in [2] that aim to answer these requirements deploy sensor and actuator networks together with orchestration protocols to provide a flexible industrial environment that can improve production parameters, overview and the flexibility of factory floors.

Concepts from Fog Computing propose that a further increase in reliability and reduction of latencies as well as an increased functionality can be achieved by migrating tasks and applications from cloud containers to local nodes and gateways. Service oriented approaches such as in [3] as well as Asynchronous messaging based ones as in [4] propose an abstracted view of devices and resources as well as the possibility of cluster or region level resource access from application or services.

Allowing applications to access resources and devices from any node on the network allows for application migration in the system which can be used to further improve runtime parameters and reduce costs and latencies as proposed in [5]. Migration poses a placement problem for Fog Systems which can be decomposed into an estimation and modeling problem as presented in [6] and a management problem as in [7]. The management problem can consider single shot systems with finite tasks being deployed, executed and having results returned in the case of more traditional Fog Computing approaches. For Industry and CPS systems we consider highly connected and continuously running systems that use resources and access devices from a varying set of gateways and locations.

We propose graph based CPS system analysis approach based on key parameters that allows for a more rigorous connectivity, reliability and categorization of IoT Fog Systems. Furthermore, the resulting generation or replication parameters allow for scaling and large systems to be generated based on small use cases which aid in optimization and load balancing algorithm development and testing. Finally, we analyze a set of use-case scenarios developed for workstations available at our industrial partner, based on which we determine average parameters and categorization of such systems. Using this data we compare virtual systems generation methods based on random, pseudo-random and measured parameter to analyze how accurately they reflect real systems and how they differ.

II. OPTIMIZATION AND ANALYSIS CHALLENGES

Optimization in IoT systems is made difficult by two main factors. The first is with respect to the complexity of these systems where migrating an application, service or resource leads to the alteration of the connection topology as well as the locally available resources the effect of which is difficult to model or estimate. The second hindrance in developing optimization solution for Fog computing is the lack of real-life information on data-sets, use-cases, application sizes, processing requirements, message rates and their impact on the deployed nodes. Most available use-cases such as in [8] show a high-level view of agile manufacturing systems which can't be used for optimization purposes. The state of the art solutions for this as in [9,10,11,12] are the proposal of example applications and use cases on top of which they build their optimization methods. The drawback of these approaches is that there is no guarantee that the proposed system parameters or use-cases resemble real-life solutions, reducing the utility of the proposed models and algorithms.

The solutions look at different aspects of optimization. The solution in [9] is a clustering and stage based method based on a simple delay model between components, while in [12] a simple topology reduction is attempted. The proposals in [10,11] show a more elaborate application and delay model considering processing delay as changing through the deployment locality as well as considering several different connection delay types. Although these models are extensive, the constants, rates and values that change through migrating are assumed instead of measured or deduced. This may cause certain optimization approaches to seem more advantageous than others as well as leading to inaccurate models.

When considering highly connected complex systems, the common approach is the use of graphs to model the connections between entities. This has been done to model WWW connection as in [13] as well as to optimize Wireless Sensor Networks (WSN) as in [14] for connection reliability, zero single node failures and other parameters. Increasingly there are attempts at using these methods on Fog Systems as in [15] where a tree based system is used or in [16] where graph repartitioning methods are proposed. These proposals have the same drawbacks of lacking real deployment data on which to test their algorithms on real-world systems where the clustering factor, connectivity and distribution of nodes might vary greatly. Finally, these solutions don't consider the existence of a physical and virtual connection set where the physical one looks at where application, devices and resources are deployed or orchestrated, while the virtual one only looks at which components interact with each other. This view would allow Node mapping between one graph to another which is the core of the Fog Computing placement problem.

III. USE CASE DESCRIPTION

The presented use cases are based on the 4 physical workstations and proposed automation and control systems that are in concurrence with the requirements of our partner and those presented in Industry 4.0.

A. *Physical Systems*

1) *Metrology Workstation (Dimention Measurements)*

The Dimension Testing Metrology station contains a CMM machine, alongside some smaller measurement devices, and a environment monitoring station for accurate temperature and humidity control which is essential for accurate measurements as well as a monitoring screen and a parts organizing station.

This workstation is designed to measure tolerances on finished components as well as bending and torsion. The key factors here are linked to quality assurance, environmental monitoring and Energy Control and monitoring.

2) *Metrology Workstation (Metallurgy)*

The Metallurgy Metrology workstation contains a Hot Mold Machine, a Polishing Controller, Digital Microscopes, a part organizer and monitor.

This workstation is used to take weld pieces, mount them into plastic molds, polish and analyze these for integrity. The key factors here are part monitoring, tests logging and quality control.

3) *Metrology Workstation (Stress Testing)*

The Stress testing workstation contains a Compression testing, Burst testing and Stretch testing instruments as well as parts organizer and monitor.

This station is used to test the integrity of welded tubes under pressure through the burst tests, as well as component characteristics through the compression and stretch or pull tests. The key components are regarding parts monitoring and tests logging together with energy monitoring and quality control.

4) *Assembly Line*

The Assembly line contains several ABB Robot arms with 2D vision capabilities together with welder units, a conveyor belt with position sensors, controls and bar code readers, an input and output part organizer, safety proximity laser curtains and emergency stop buttons.

The assembly line is used to weld and assemble components going through the line based on their part numbers. The key components are part monitoring as well as quality control through the metrology stations, safety and energy monitoring and control.

B. *Application Use-Cases*

The design of the application use-cases are based on the existing hardware and sensor environment as well as guidelines presented in [17]. The main purpose of these systems is to map flow based energy control, part monitoring access and environmental control on top of existing hardware with a realistic composite application approach. Each scenario has a different approach to the topology of the connections. The part Logging system is designed to be a more connected design while the energy monitoring and access control scenarios are more hierarchical or resemble fractal and tree based graphs.

1) *Part Logging and Flow Monitoring*

This system is designed to monitor the progress of parts through the assembly and metrology environments as well as gather data on parts production rate and use per environment as well as receive controls from the energy optimizer on where to assign parts.

The virtual connections of the system can be seen in fig. 1 where each component or application is shown with its respective cloud, storage, local access and device connections.

We can see from the graph that the applications are highly connected between each other while the devices usually belong to one controller/ orchestrator or reader with no direct machine to machine (M2M) communication between devices.

The use-case contains a main parts flow monitor and a status monitor connected to a local component for each room which then communicates with each individual machine type controller and reader. We have a local repository for parts status monitoring for each workstation as well as local access. Finally, there is a cloud monitoring connection for saving data and advanced analysis.

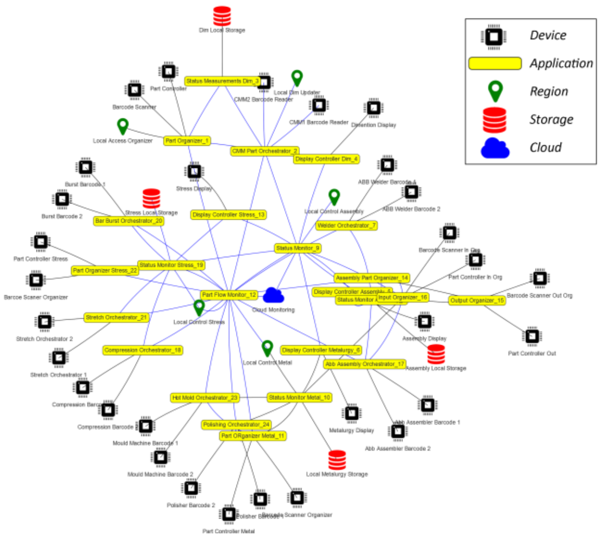

Fig. 1. Parts and Flow Monitoring subsystem

2) Energy Monitoring and Control

The system is designed to monitor the energy use of devices and machines for each workstation and the factory as well. It also controls the power supply of machines based on parts flow and existing optimization scenarios. These parameters are shown on displays and saved to a cloud source.

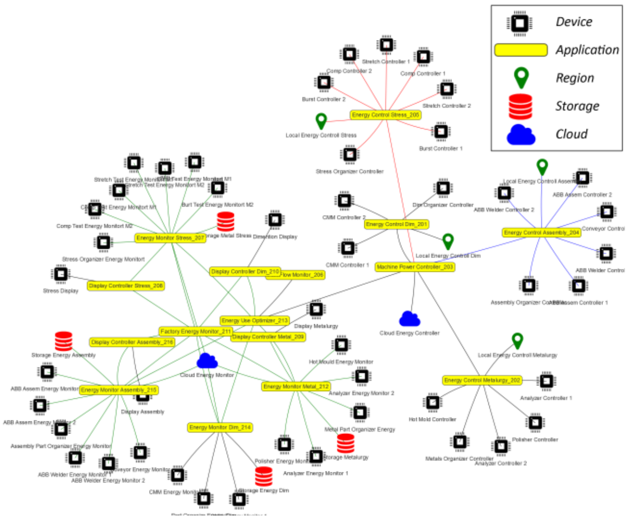

Fig. 2. Energy Monitoring, Optimization and Control

The virtual connections of the system can be seen in fig 2. We can see from the diagram that the connections in this scenario are much less clustered and more hierarchical than in the previous scenario especially for the left half, which is the control region, while the right is the monitoring and optimization part.

The presented use case contains a cloud connected main power controller connected to local controllers that have local access and that in hand orchestrate the individual devices. This component is linked with the Energy Optimizer which is connected to the flow monitor and Main Energy Monitor.

The main monitor is linked with local monitors that save data to local storage and show info on local displays while saving data for further analysis on a common Cloud Energy Monitor endpoint.

3) Access, Safety and Environment Control

This system is designed to take care of controlling and logging access on machine and rooms as well as controlling safety and environmental variables inside the rooms. Cloud logging and control as well as local access and displays are connected to these components.

The graph of these connections can be seen in fig. 3 where we can see that the graph has a similar structure to the one in fig. 2, but containing more local access points and a much more hierarchical system which is designed for layered safety in the case of access and security.

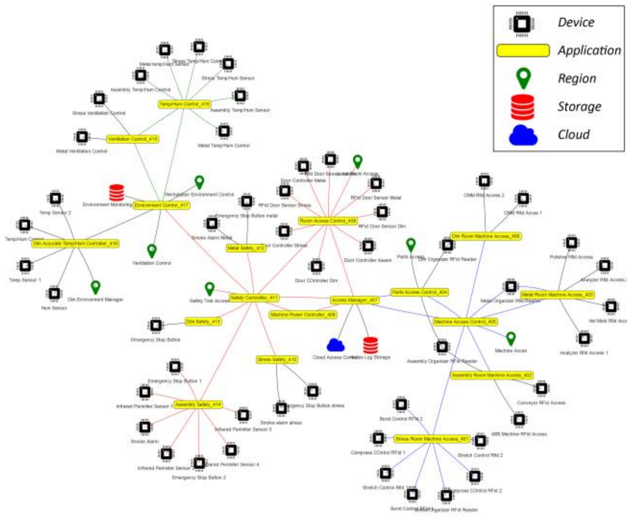

Fig. 3. Acces, Safety and Environmental Monitoring and Control

This scenario contains a main access manager that controls the room access, parts access and machine access modules that in hand orchestrate the room modules and their devices. The access manager is linked to the safety controller which in hand is linked to the environmental controller to initiate safety protocols if needed. The safety controller is linked to individual room components that in hand control the safety devices and sensors available. The environmental components orchestrate ventilation, temperature and humidity control through factory level components. It also has specialized units for the high precision environment control requirements of the Dimension measurement workstation which increases the number of sensors and splits the humidity and temperature as well as the ventilation.

4) Combined System

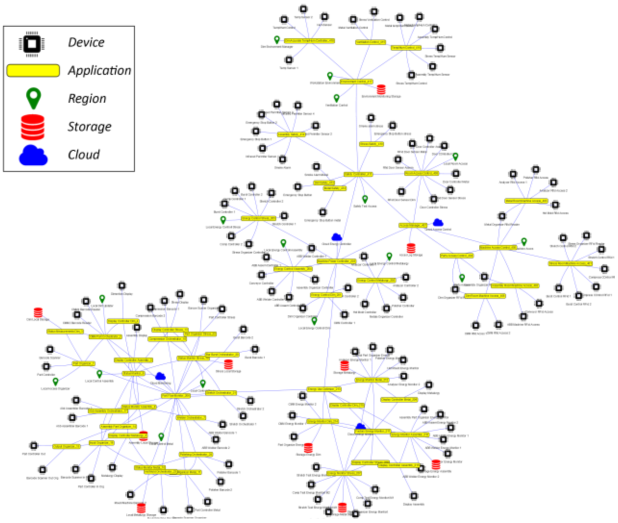

Fig. 4. Combined System

The combined system seen in fig. 4 looks at connecting the separate systems for a fully functioning factory floor. This is done by linking certain main components in these systems through a layered architecture design.

The main connected components that are the part flow controller with the energy use optimization that connects to the machine part controller which then relates to the Safety Controller and the Access Manager.

IV. Analysis Parameters

When considering the analysis of IoT systems, there are several parameters that need to be examined that may be interesting for two reasons. The first reason is for replication and scaling of these systems when testing how optimization algorithms perform with larger datasets. The second reason is to identify characteristics of these systems that can be used to better select and create new optimization approaches. Finally, as proposed in [18] these parameters can be used to calculate or estimate latencies, reliability and redundancies of entities and the system. For the analysis, we consider the system as a graph $G = (V, E)$ where V denoted the vertexes, nodes denote the applications, storages, cloud entities, and regional access points while E denotes the Edges or connections between these.

We consider $V_i \in V_k \in V$ where k denotes the type of Node and i denotes the number or id of the node and V_k denotes the set of all nodes of the same type. For the edges, we denote $E_{i,j} \in E$ where *i and j are* the id of the connected Nodes and $E_{i,j}$ denotes the edge itself. For the edges, we consider $E_{i,j} = E_{j,i}$ due to the undirected and unweighted nature of our graph.

A. Replication Parameters

The replication parameters are simple properties of the graphs that look at key parameters we can use to replicate the structure of the graph to allow replication and scaling of certain use-cases.

1) Application Resource Use

This parameter looks at what is the average number and distribution of device, region, storage and cloud connections from applications. We denote the resource use of an Application by R_{App}^{Type} where *Type* is the resource type and *App* is the application id. Equation (1) defines this resource use as a sum of all connections of an application to a type of device.

$$R_{App}^{Type} = \sum_i^{E_{App,i}} V_i \in V_{Type} \tag{1}$$

2) Clustering of Applications

This component looks at how certain applications group together into clusters and what is the average size and number of these clusters and how interconnected they are. We define a cluster $Clust_i$ where Application $V_i \in Clust_i$ as defined by a clustering algorithm like K-Means[19] or DBSCAN[19].

3) Connection Locality

This factor looks at what are the chances of one application connecting to resources and devices from the same gateway and how many external resources and applications it uses. We are interested in the distribution of these types of connections. We define three types of locality $\{Local, Cluster, External\}$ where $V_i \in \mathrm{Loc}_k^{Local}$ so that all elements V_i are on the same gateway, $V_j \in \mathrm{Loc}_k^{Cluster}$ so that all elements V_j are part of the same Cluster and $V_l \notin \mathrm{Loc}_k^{External}$ so that:

$$\mathrm{Loc}_k^{External} = V - (\mathrm{Loc}_k^{Local} \cup \mathrm{Loc}_k^{Cluster}) \tag{2}$$

4) Inter-App communication

This parameter looks at the average number and distribution of connections between applications deployed on the system. Together with clustering and locality, this component helps create a more realistic environment. We consider the connections of an Application by C_{App}^{Area} where *Area* denotes the region to which the application connects to which can be local or cluster level. Equation (3) defines these connections as a sum of all connections from each region coming or going to the application.

$$\begin{aligned} C_{App}^{Area} &= \sum_i^{E_{iApp}} V_i \in \mathrm{Loc}_k^{Area}\text{, where } V_{App} \in \mathrm{Loc}_k^{Area} \\ C_{App}^{Ext} &= \sum_i^{E_{iApp}} V_i \in \mathrm{Loc}_k^{Ext}\text{, where } V_{App} \notin \mathrm{Loc}_k^{Ext} \end{aligned} \tag{3}$$

B. Graph Parameters

The graph parameters are designed to show certain characteristics of these systems that can be translated to parameters of interest, such as reliability, latencies, clustering and interconnectivity. These characteristics are used in [20] to analyze a varying range of systems such as the World-wide web, social networks, citation interconnectivity and others.

1) Connectivity

Connectivity checks if there is a route $\mathrm{route}(i, j)$ from any node V_i in the graph to any other node V_j in the system. After verifying connectivity, we look at how many distinct connected graphs we can find in our system. This parameter aids in clustering of these connected graphs as well as shows separate subsystems. Our use-cases are all connected graphs so this parameter, while important in the analysis, in our case is overlooked when discussing results.

2) Average Path Length and Graph Diameter

Average path lengths look at what is the average distance between two nodes while the graph diameter looks at the maximum distance. These parameters can be used to determine simple average and maximum latencies and hops within a network while comparing them to node and vertex counts can help us determine QoS parameters. The minimum distance from node V_i in the graph to any node V_j can be ca computed through the Dijkstra's algorithms and is denoted with $route_{Min}(i, j)$ while the average path in a system is defined as in (4) and the diameter or maximum shortest route is defined in (5)

$$AVG_{Route} = \frac{\sum_i^V \sum_j^V route_{Min}(i,j)\ , i \neq j}{size(V) \times (size(V) - 1)} \tag{4}$$

$$Diameter = \max_{V_i \in V} (\max_{V_j \in V} route_{Min}(i, j))\ , i \neq j \tag{5}$$

3) Clustering and Clustering Coeficient

The clustering coefficient looks at the average number of triangles $Tri(V_i)$, or three node pairs with each node being a member of a system. This number is divided by the total number of possible triangles, adjusting for the size of the graph. The sum of these values is the Clustering Coefficient (CCF) of the graph as can be seen in (6). This information can be used to determine how tightly coupled a cluster is. This parameter could be useful in determining the optimization of subsystems using divide and conquer techniques in optimization, especially latency optimization.

$$CCF = \sum_i^V \frac{Tri(V_i)}{size(V) \times (size(V) - 1)} \tag{6}$$

4) Graph Degree Distribution

The graph degree distribution (GDD) looks at how many nodes have a certain number of connections in a system compared to the maximum possible number of connections. The number of nodes that have a certain degree can be calculated based on (7) where k is the edge count.

$$GDD[k] = \sum_i^V (\sum_j^E E_{i,j} = k) \tag{7}$$

This gives a view on how the connections differ between systems and gives us the main comparison factor when categorizing our system as well as verifying generated systems.

5) Graph Betweenness Centrality Distribution

The graph betweenness centrality (GBC) of a node is calculated by counting the number of shortest paths $route_{Min}(i, j)$ that contain a node and compare it to the maximum and minimum values present in the system. Our implementation looks at all the paths whose length is equal to the shortest one. The equation (8) shows how the centrality of one node is calculated.

$$GDC(V_i) = \sum_j^V \sum_l^V V_i \in route_{Min}(j, l)\ , i \neq j, l \tag{8}$$

(6)

The distribution looks at how many nodes have these values between a certain range. This parameter is key in determining high importance nodes in the system as well as critical single points of failure. This characteristic is also important when comparing systems and verifying our generated graphs.

C. Network based Categorization

There are several network types based on their connection typology as suggested in [20], each having their real world equivalent and their set of attributes. We analyze our use-cases and compare them to the behavior of known models such as random-graphs, Markow graphs, non-scalable networks, small-world models, Barabas-Albert and other growth models.

With each network having its own characteristics, they require different approaches when certain optimization or analysis attempts are made such as clustering and single point of failure rerouting.

The analysis and categorization approximation of our system will allow for model specific method to be applied which may reduce run-times and reduce the diminishing returns we see with similar systems, such as in [21].

V. Results and Comparisons

A. Replication Data Analysis

The data analysis for the 4 virtual scenarios from the application resource use and locality point of view can be seen in Table I. broken down to device, storage, cloud and local interfaces and computed through the equation in (2). Each component has a local and external factor which looks at the locality of these connections with the local being the gateway hosting most resources while the external represents other gateways.

The connections between application are described in Table II. Where they are broken down to local connections, cluster connections and external connections based on (3) and (4). These are important when designing systems when considering approaches that focus on connections remapping SDN based router rewiring and other similar methods.

The clustered connections refer to the clusters in fig. 5 and looks at all the connections that are not to the same Gateway but are in the same cluster, while the external ones look at all connections to external gateways not on the cluster while the total shows all the connections.

TABLE I. RESOURCE USE PA4RAMETERS

Type	Prop.	Scenario: Energy Loc.[a]	Ext.[b]	Total	Parts and Flow Loc.[a]	Ext.[b]	Total
Device	Min	0	0	0	0	0	0
	Max	7	0	7	2	0	2
	Avg	2.87	0.0	2.87	1.25	0.0	1.25
Cloud	Min	0	0	0	0	0	0
	Max	1	1	1	1	1	1
	Avg	0.12	0.18	0.31	0.04	0.04	0.08
Storage	Min	0	0	0	0	0	0
	Max	1	0	1	1	0	1
	Avg	0.25	0.0	0.25	0.16	0.0	0.16
Local Access	Min	0	0	0	0	0	0
	Max	1	0	1	1	1	2
	Avg	0.25	0.0	0.25	0.2	0.08	0.29
		Access and Sec.			Combined System		
		Loc.[a]	Ext.[b]	Total	Loc.[a]	Ext.[b]	Total
Device	Min	0	0	0	0	0	0
	Max	8	1	8	8	1	8
	Avg	2.94	0.05	3.0	2.3	0.01	2.32
Cloud	Min	0	0	0	0	0	0
	Max	1	0	1	1	1	1
	Avg	0.05	0.0	0.05	0.07	0.07	0.14
Storage	Min	0	0	0	0	0	0
	Max	1	0	1	1	0	1
	Avg	0.11	0.0	0.11	0.17	0.0	0.17
Local Access	Min	0	0	0	0	0	0
	Max	2	0	2	2	2	2
	Avg	0.38	0.0	0.38	0.26	0.05	0.32

[a] Loc-Belonging to Local Gateway

[b] Ext-Belonging to External Gateways

Determining the number and size of the clusters for the analysis that was used for the app data in Table II. was done using a Density-Based Clustering Scan (DBSCAN) on the graphs.

TABLE II. APPLICATION PARAMETERS

Property	Parameters: Energy Local	Cluster	External	Total
Min	0	0	0	2
Max	4	6	1	8
Average	1.375	1.25	0.125	2.75
	Parts and Flow			
Min	0	0	0	2
Max	5	11	2	15
Average	2.0	1.5	0.33	3.83
	Access and Security			
Min	0	0	0	1
Max	6	4	1	8
Average	1.66	0.55	0.11	2.33
	Combined System			
Min	0	0	0	1
Max	6	14	3	15
Average	1.64	1.21	0.32	3.17

The configuration of the scan requires a minimum number of points for a cluster which for us is 8 and an epsilon which is a maximum distance between two peers which in our graph is 1. The minimum points value is determined by the structure of the graph. A more highly connected graph would require higher values to return distinct clusters rather than one big cluster.

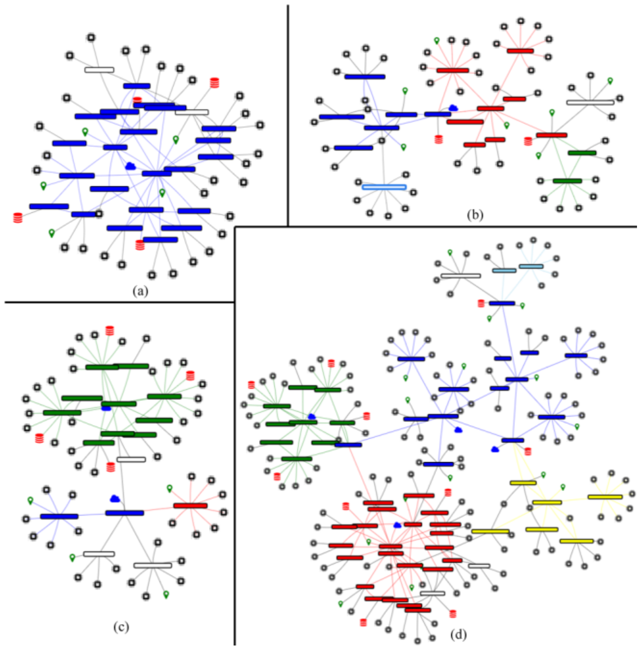

Fig. 5. DBSCAN Clusters

The resulting clusters can be seen in fig. 5 where (a) is the Parts and Flow Monitoring system, (b) is the Access Safety and Environmental Control and monitoring subsystem, (c) is the Energy Monitoring Optimization and Control subsystem and (d) is the Combined System. Individual application clusters are coloured the same and applications that are not part of any cluster are colored white.

This clustering method made for an average cluster size of 7.42, a maximum of 22 and minimum of 1. The method resulted in an average of 2.25 applications not being assigned a cluster. We can see that it works well in (d) and (b) where the density of nodes is more uniform and the results are weaker in (a) where the tightly coupled nature of applications results in one big cluster. In (c) due to the varying density we see that the top part of the graph is well clustered while on the bottom it identifies two small clusters and two unassigned nodes.

B. *Network Analysis*

The subsystems are analyzed based on the parameters in section IV. B. where the connectivity path length and diameter are the more basic properties of the system. For our tests, all the systems are made up of connected graphs, but this test would allow a fast clustering and easier group based optimization in cases such as the combined system if there were no connections between subsystems. The average diameter is 7 hops, while the average path length is 4.15. The maximum dimeter is in the combined system with 9 as well as the highest average path length of 5.23. We can see that diameter and average path length (APL) increase with the size of the cluster and are reduced with the increase of clustering as in (c) with a Clustering Coefficient

(CCF) of 0.01 having an APL of 3.84 and the more tightly clustered (a) with a CCF of 0.09 has an APL of 3.29.

We looked at the CCF of the applications on not just systems but also that of the subgraphs. The average CCF of the systems is 0.0425 varying between 0.016 and 0.09. If we consider the clusters by themselves the average CCF of clusters that have a size larger than 2 is 0.208 with values between 0.09 and 0.46.

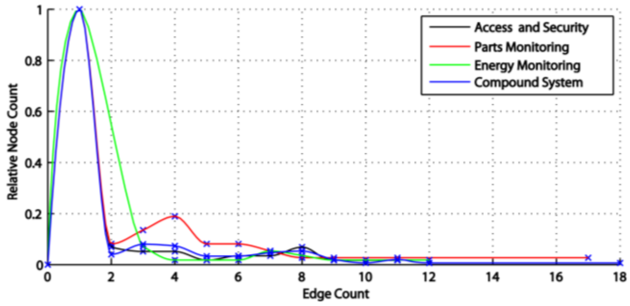

Fig. 6. Graph Degree Distribution of Systems

The Graph Degree Distribution of the systems can be seen in fig. 6. The number of nodes displayed is relative to the maximum number of nodes to allow a comparison between the graphs. For the systems, the highest node count values were at 1 connections, which is due to the device and resource links which are usually used by one application. The maximum values for these are 58 for access (5.b), 37 for parts monitoring (5.a), 56 for Energy (5.c) and 150 for the combined system (5.d). The highest number of edges are on the combined system with 18 and the second is on the Parts monitoring with 17. Every Node has at least one connection as the connectivity of the graphs show as well.

The Graph Betweenness of the systems is shown in fig. 7. The centrality value is a relative value to the maximum available on the system which is scaled to account for network size differences. The relative node count is scaled to the max values as well.

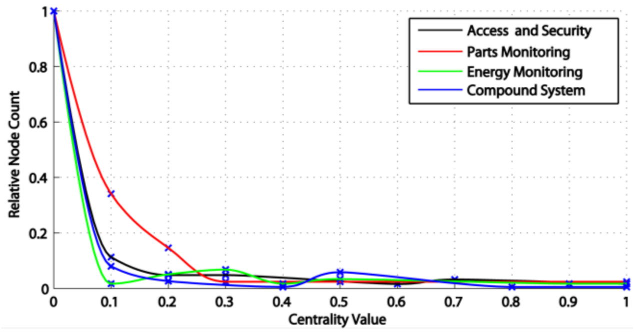

Fig. 7. Graph Betweenness Distribution of Systems

The node in (d) with the highest absolute centrality has a value of 40745 possible shortest paths crossing this node. This high number is also due to our implementation of the algorithm where we calculate the minimum distance between two nodes and consider all paths of the same lengths. This values are 3763 for the Energy Monitoring, 4012 for Access Control and 3886 for Parts Monitoring. The devices and resources often have a value of 0 residing at the edge of the network, not providing connection between any two components.

Based on the betweenness data as well as the graph degree distribution and structure of the system we can show some similarities with existing models. The Access Control and Energy Monitoring Systems have similar structures and the data in fig. 6 and 7 show that they have similar properties in structure to hierarchical and fractal networks with certain outliers and density variations. A closer look at these systems shows that their distribution and betweennness, especially that of the Access Control are like a Barabási-Albert model with an initial degree, m_0=1. The Parts Monitoring system has a different architecture with similar properties to a Random Graph when we look at the applications connections and the lack of clustering, as well as the outliers in fig 5. and fig 6. If we look at the Combined system, the plotted data as well as its structure suggest that it has similar attributes to the Random Network that models the World Wide Web (WWW), having clusters form and a varied type of connections.

C. Replication Analysis

When looking at the parameters used to generate use cases we can consider certain properties of interest. The increased adoption of connection locality and clustering can be seen in fig 8. Part (a) shows a completely random system with just the node numbers and average connection data being used. Part (b) adds connections types, distribution and locality, while part (c) adds the remaining factor of clustering.

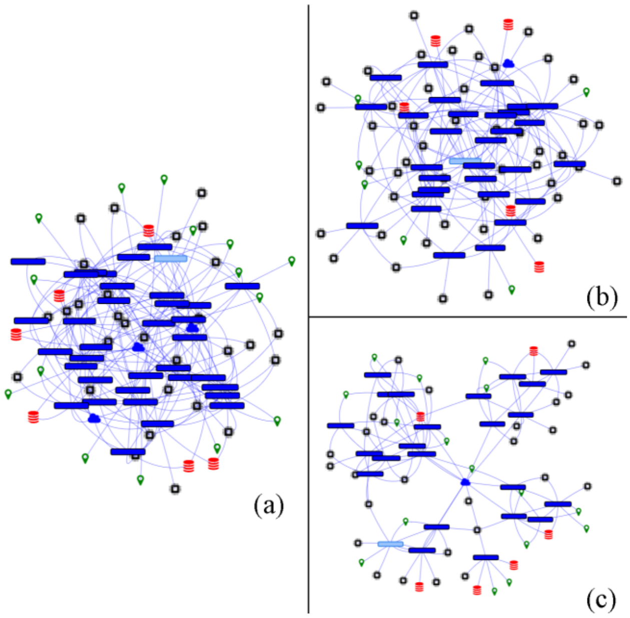

Fig. 8. Replicated Systems

We can see from the data in fig. 8 that as we adopt more parameters the systems resemble more those presented in fig.5. The system in (b) is similar to the Parts Monitoring use-case with the exception that devices are more interconnected due to the lack of locality data. System (c) contain all considered parameters and is like the Combined use-case and the Energy Monitoring one. If we consider even more realistic systems, we can devise the generation of Random Networks or Barabási-Albert models as the basis of the connection.

VI. Conclusions And Future Work

Graph based analysis and optimization of IoT and CPS Systems are key tools in improving the latency, reliability and other QoS parameters of such systems as well as allowing for a better integration and optimal use of the technology in Industry. This is needed to fulfill the QoS requirements of Industry 4.0.

This paper presents a set of Industry based use-case scenarios as well as an overlaying application systems. Using these systems, we chose and analyze key replication and analysis parameters that can be linked with QoS characteristics.

The experimental analysis show that different systems have a varying set of parameters and architectures while a compound system may mimic behaviors of Random Network Models. The replications tests show how replication systems and use-case architecture similarities are improved by considering our proposed parameter.

Future work will look at using these parameters to aid in rerouting the graphs to improve latencies as well as connecting the physical and virtual systems to aid in the placement problem. Finally, we will look at system requirements to achieve certain reliability parameters, removing single points of failures and other improvements to the QoS qualities of the system.

Acknowledgment

The authors would like to thank our partners, especially Dr. Zahid Usman at the Institute for Advanced manufacturing and Engineering (AME) with collaboration with Unipart Manufacturing Group (UMG) for their support with developing the use cases and allowing our team to survey their factory floor.

References

[1] Wang, Shiyong, et al. "Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination." *Computer Networks* 101 (2016): 158-168.

[2] Saldivar, Alfredo Alan Flores, et al. "Industry 4.0 with cyber-physical integration: A design and manufacture perspective." *Automation and computing (icac), 2015 21st international conference on*. IEEE, 2015

[3] Vogler, Michael, et al. "DIANE-dynamic IoT application deployment." *Mobile Services (MS), 2015 IEEE International Conference on*. IEEE, 2015.

[4] Nandor Verba, Kuo-Ming Chao, Anne James, Daniel Goldsmith, Xiang Fei, Sergiu-Dan Stan, Platform as a service gateway for the Fog of Things, Advanced Engineering Informatics, ISSN 1474-0346

[5] Sarkar, Subhadeep, Subarna Chatterjee, and Sudip Misra. "Assessment of the suitability of fog computing in the context of internet of things." *IEEE Transactions on Cloud Computing* (2015).

[6] Aazam, Mohammad, et al. "IoT Resource Estimation Challenges and Modeling in Fog." Fog Computing in the Internet of Things. Springer International Publishing, 2018. 17-31.

[7] Delicato, Flávia C., Paulo F. Pires, and Thais Batista. "The Resource Management Challenge in IoT." Resource Management for Internet of Things. Springer International Publishing, 2017. 7-18.

[8] Scheuermann, Constantin, Stephan Verclas, and Bernd Bruegge. "Agile factory-an example of an industry 4.0 manufacturing process." *Cyber-Physical Systems, Networks, and Applications (CPSNA), 2015 IEEE 3rd International Conference on*. IEEE, 2015.

[9] Oueis, Jessica, Emilio Calvanese Strinati, and Sergio Barbarossa. "The fog balancing: Load distribution for small cell cloud computing." *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*. IEEE, 2015.

[10] Deng, Ruilong, et al. "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption." *IEEE Internet of Things Journal* 3.6 (2016): 1171-1181.

[11] Zeng, Deze, et al. "Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system." *IEEE Transactions on Computers* 65.12 (2016): 3702-3712.

[12] Vogler, Michael, et al. "Optimizing Elastic IoT Application Deployments." *IEEE Transactions on Services Computing* (2016)

[13] Kleinberg, Jon, et al. "The web as a graph: measurements, models, and methods." *Computing and combinatorics* (1999): 1-17.

[14] Savazzi, Stefano, Vittorio Rampa, and Umberto Spagnolini. "Wireless cloud networks for the factory of things: connectivity modeling and layout design." *IEEE Internet of Things Journal* 1.2 (2014): 180-195.

[15] Jingtao, Su, et al. "Steiner tree based optimal resource caching scheme in fog computing." *China Communications* 12.8 (2015): 161-168.

[16] Ningning, Song, et al. "Fog computing dynamic load balancing mechanism based on graph repartitioning." *China Communications* 13.3 (2016): 156-164.

[17] Lee, Jay, Behrad Bagheri, and Hung-An Kao. "A cyber-physical systems architecture for industry 4.0-based manufacturing systems." *Manufacturing Letters* 3 (2015): 18-23.

[18] Voutyras, Orfefs, et al. "Social monitoring and social analysis in internet of things virtual networks." *Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on*. IEEE, 2015.

[19] Xu, Rui, and Donald Wunsch. "Survey of clustering algorithms." *IEEE Transactions on neural networks* 16.3 (2005): 645-678.

[20] Newman, Mark EJ. "The structure and function of complex networks." *SIAM review* 45.2 (2003): 167-256.

[21] Heller, Brandon, Rob Sherwood, and Nick McKeown. "The controller placement problem." *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012.