Journal of the Indian Institute of Science
A Multidisciplinary Reviews Journal
ISSN: 0970-4140 Coden-JIISAD

REVIEWS

On the Scalability of a Cyber Physical System

Kumar Padmanabh

Abstract | Sensors and actuators are the fundamental constituents of a cyber physical system (CPS). Pervasiveness of the sensors, requirement of collaborative processing of data, historical importance of data, and the ever increasing different forms of user applications are not only making data management more complex, but are also bringing more challenges for the scalability of a CPS. On the other hand, the requirement of bringing actuators to complete the feedback loop of a CPS is making it even more complex. There are three stages of complexities associated with scalability: in the physical infrastructure of the data acquisition system, in the communication mechanism, and in the server and user applications. In all these stages, diversity, interoperability and individual capacity limitations of components play an important role in establishing the overall scalability limits of the CPS. In this paper we will systematically study how these parameters affect the scalability limits. Furthermore, we will also study the existing technologies (e.g. of big data and cloud) which can be used for addressing the challenges of scalability imposed at different stages of futuristic cyber physical systems.

Keywords: *Cyber Physical System, Sensor Network, Cloud Computing, Scalability.*

1 Introduction

This is an information intensive age. For last few decades, researchers have been involved in developing information technologies to bring positive differences in human life. These technologies have been developed with the understanding that either the data are readily available in the computing systems or there are dedicated efforts for data entry into software systems. For example, banking software requires someone to feed transaction data so that banking software could execute tasks of data management, analytics and reporting. However, in the last decade, with advancements in sensor networking, internet of things and other cyber physical systems, this paradigm has changed. Now a cyber physical system can generate relevant data automatically and feed them to the software. Moreover, in conventional mechanisms, the data was supposed to be shipped to different geographical locations, typically to the server for computing and analysis of the same. With a number of cyber physical systems in place, now computing is supposed to be present in all end devices, thus bringing another change in the pattern of computing.

The world is going to be instrumented more in the future. The ubiquity and pervasiveness of cyber physical components would be felt everywhere. The automatic generation of data and feeding of data through hyper-distributed computing would solve socio-economic problems, which would not be possible otherwise. Such a huge system would be complex, and would pose new challenges to the computing world. Out of these challenges and complexities, the scope of this paper is limited to that of scalability. The system architecture of a typical cyber physical system now needs to consider the following:

1. ***Heterogeneous sources of data:*** The source and format of data ought to be diverse in nature. Communication and computing systems need to take all diversities into account.
2. ***Volume of data:*** A cyber physical system is supposed to monitor systems, events and processes

Corporate Research,
Robert Bosch Engineering and Business Solutions Ltd.
Gold Hill Square, 690,
Hosur Road, Bommanahalli,
Bangalore 560 068,
India.
kumar.padmanabh@in.bosch.com

continuously. Each instance of monitoring will result in data flow into the system. With ubiquity of sensing and computing, the volume of data is bound to grow exponentially. The current system is not designed to handle this enormous volume of data.

3. ***The diversity of communication system:*** Each cyber physical system is supposed to have one or more types of communication system in place. The real value of a CPS is in the collaboration of end devices for information rendering, which cannot be done otherwise. For example, physical security system of a building can provide occupancy related data to the energy management solution. Similarly, the energy consumption pattern can supply occupancy level of a building to the security system, and thus add value to each other's core functionalities.
4. ***The diversity of user application:*** The same set of data might have different implications of criticalities to different applications. Additionally, each application requires data in different forms. In this scenario, the availability and accessibility of the data becomes challenging, in case one wants to scale up the system.

To understand the challenges of scalability in futuristic cyber physical systems, let us analyze one of the example deployments of a CPS.[1] In this example, 300 data centers located in different cities of India were supposed to get instrumented with various sensors. Before this deployment, the operating conditions of these data centers were maintained manually to honor the compliance and warranty requirements. For example, temperature and humidity levels of the datacenters were supposed to be in certain respective ranges. It was required that there was no AC leakage. In order to maintain these conditions, in the legacy system, an operator used to visit the data center and log the temperature, humidity, leakage, UPS status etc., in the physical logbook on an hourly basis. The human error and inefficiency resulted into undesirable downtime and unruly energy consumption. There were multiple objectives of instrumenting the data centers. Stakeholders wanted to monitor all these parameters online. It was required that existing fire monitoring systems be integrated into the proposed system. They wanted to generate alarm whenever a parameter under observation crossed the recommended range. They wanted to generate reports of the performance based on observed data. And most importantly, they wanted to minimize electricity consumption. Though it was a simple application from development point of view, it was very critical for operation of the datacenters, because several operation related decisions were supposed to be made based on the alarms and reports. For example, upon detection of leakage of AC, an alarm ought to be generated triggering a series of several workflows for different stake holders. Arrangement of different components of this deployment is depicted in Figure 1.

Hardware platform developed for this purpose had 11 sensors (temperature, humidity, smoke, UPS status etc). Four sets of such platforms were supposed to get deployed in each quadrant of a data center. Thus 1200 units of hardware having array of 11 different sensors were supposed to pump data in real time to the software system responsible for supplying APIs to dashboard, alarms and reporting related applications. In this application, frequency of sensing was required to be once in every 10 second. One sample of data from all sensors along with all communication overhead was supposed to be almost 1KB. For operator-in-charge of the datacenters, the generation of alarms was essential in real time. The integrated view of data was more important than the individual data. Thus, entire application was producing 13.2 MB of data every 10 seconds. This is equivalent to 4.68 GB of data per day. It is to be noted that unlike other data, in sensor related data, each byte has its own significance. For example temperature data can be incorporated into just a single byte. Therefore, it is not pragmatic to extract information from this data and store the information rather than all the data. Moreover, since there is historical significance of data, the stored data might keep growing and amounting to 1.2 TB of data each year. Thousands of different software threads associated with different end applications and online analytics accessing the uniquely identifiable bytes individually out of 1.2 TB of data imposes different kinds of challenges. Thus, in a typical CPS based solution, it is required that following points are taken care:

a. ***Time synchronization:*** There is a mechanism in place to synchronize all devices so that all the sensors and actuators are working in the same frame of reference.
b. ***The data overflow:*** The base stations or data aggregation points have mechanisms for proper input queue management and it is ensured that data doesn't overflow.
c. ***Data storage:*** The system ensures that the stored data are available and accessible seamlessly.
d. ***Software thread management:*** The software threads which process raw data and help the

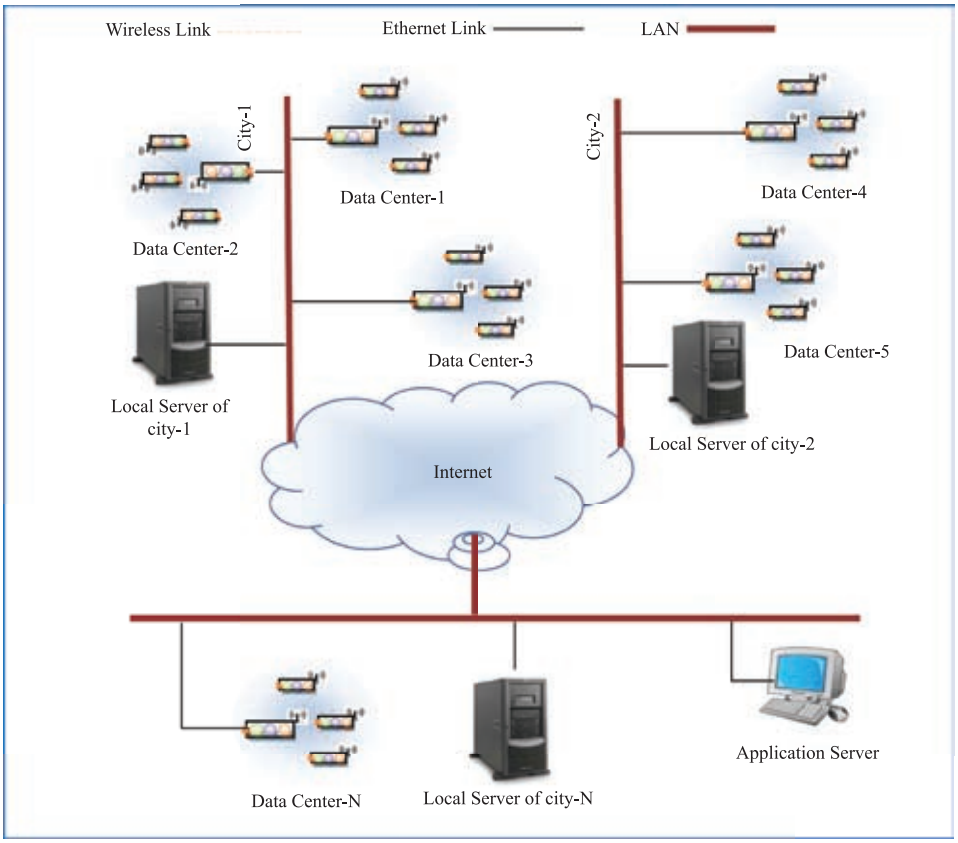

Figure 1: The arrangement of different components of CPS used in the deployment example.

system to infer different useful information before getting stored in the database, and the software threads which process different queries coming from huge number of user applications, are optimized for performance for real time applications. It has been proven that server side software thread management is scalable. However, software thread management in a Gateway based on embedded platform is challenging.

It is to be noted that if the above mentioned points are not taken care of, then it would become a bottleneck for the overall scalability of the CPS. In what follows, we will study the concerns of scalability in detail in section 2. Subsequently, in Section 3, we will analyze how these challenges can be addressed systematically. More specifically, we will study other existing technologies for their relevance in solving scalability related problems.

2 Scalability Related Concerns

Different components which constitute a typical cyber physical system are different from each other from technological point of view. The embedded systems, middleware and server components are required to interact with each other. The functionalities of these components are different. Therefore, they are uniquely designed. There are distinctive scalability related challenges associated with them. In spite of the uniqueness, each component has to deal with diversities, interoperability and inherent capacity limitation of the components (e.g. data rate, cable length and the volume of data). These factors contribute to overall challenges for the scalability. Figure 2 depicts a multidimensional view of the presence of different challenges.

In this section we will study the unique challenges associated with (i) the mechanism of communication among end devices, (ii) the mechanism of data acquisition, (iii) the middleware which is responsible for the interaction of server components (applications) and (iv) the user applications.

2.1 *Challenges associated with the data acquisition system*

The data acquisition system has to deal with the way sensing is done. It also ensures that the data are made available for rest of the system. Since

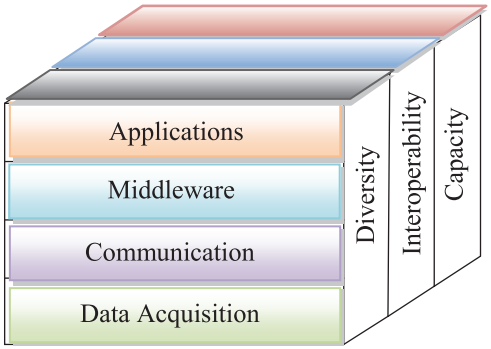

Figure 2: Multi dimensional view of challenges for the scalability. Diversity, Interoperability and Individual Capacity Limitations impose challenges at different levels starting with data acquisition till user applications through communication system and middleware.

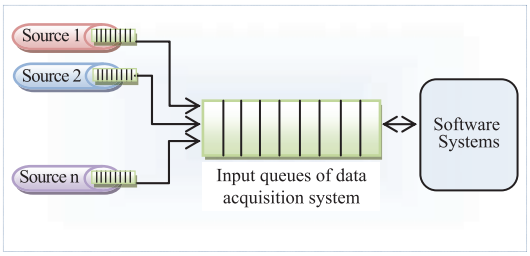

Figure 3: The receiver of data aggregator receives data from various sources and immediately puts it into the input queue before getting scheduled for the processing. There is a chance of data loss due to congestion in the input queues.

sensing is individualistic in nature, they don't play a direct role in scalability. Data reporting is an important process in which many smart sensors would like to report data to a system which connects it to the outside world. In sensor networking jargon it is known as base station. At the base station, since every sensor reports data, data acquisition system should be such that there is no scope of any data loss. Moreover, modern days analytical tools require N^2 or/and N^3 time to process N data points, however, the speed of the I/O channel has not increased according to storage capability.[8] Thus, the performance of base station is bound to get affected negatively by the congestion at the input queue as depicted in Figure 3.

Therefore, while receiving data in the Base Station, it should be ensured that tradeoff between the amount of time a packet resides in the queue and processing time required is optimum, so that queue doesn't grow infinitely. It has been proven through deployment[3] that queue at the data acquisition system should be designed carefully taking into account the (i) number of devices reporting data to the system (ii) packet length and (iii) the storage mechanism.

Timing is an important parameter in a cyber physical system. Unlike an enterprise network where timing is just a parameter of performance, in cyber physical systems timing is a measure of correctness. If a task is not able to get executed within a predefined time limit then system is considered to be "*incorrect*". In this perspective, an end device in CPS is supposed to execute task within the specified time. However, sense of timing is different from device to device due to different pattern of drifts and offsets in the clocks of the devices. Therefore, it is prudent that drifts and offsets are compensated with appropriate synchronization mechanism. With increasing number of end devices in the CPS, the diversity of electronics of the devices also increases. It results in two seemingly incorrigible challenges: (a) with increasing number of end devices, the correctness of synchronization will keep decreasing and (b) the complexity of algorithm will increase exponentially, because the number of variables to be optimized increases with the increase in the diversity. These two points are required to be handled separately.

2.2 *Challenges associated with communication systems*

Sensors and actuators are the end devices for providing data and acting according to the requirement of the applications respectively. These devices are supposed to get connected with each other using special-purpose-communication system, e.g. RS432, BACNET, LonTalk, industrial Ethernet, Z-Wave and ISA-100. These systems are based on industry standards developed for specific purposes. Eventually, due to this specific design constrains the number of connected device in one instance of networking, the permissible data rate, the number of hop-counts, transmission range cannot grow beyond the respective fixed limit. Therefore, there is an inherent limitation to the scalability of these standards in the terms of number of devices that can be connected in one instance of communication channel and length of communication channel itself. Some of these limitations are mentioned in Table-1.

While a wired network has limited number of output ports, the total number of connections is limited. On the other hand, wireless networking used for the connectivity of the end devices imposes different kinds of limitations. Due to interference, the number of devices in a particular geographical area is not allowed to grow beyond

Table 1: Limitations of the capacity of different communication standard that can be potentially used in cyber physical systems.

S. no	Standard	Max no. of connected devices	Remarks
1	RS 485	32	<4000 feet cable
2	BACNET	128	Master slave architecture
3	LonTalk	34	Master slave architecture
4	Industrial Ethernet	Virtually infinite	Unreliability increases with scale
5	Zigbee	256	Inherent wireless interference
6	Z-Wave	232	Mechanism of manual/software pairing a challenge

a certain numbers. Moreover, the metallic or industrial environment would affect the throughput negatively. This will be resulted in additional depreciation of the signal strength which affects the throughput, and hence the scalability negatively.

Jiang Li et al.[2] and P. Gupta et al.[5] found the respective bounds on the number of devices that can be connected. With the limited transmission range available at each node, nodes which are sufficiently separated can reuse the same frequency of transmission concurrently without any interference. Therefore, the total amount of data that can be transmitted in one hop kind of arrangement has been found to be proportional to the total area.[5] Thus capacity of an ad hoc wireless network was found to be $O(n)$, where n is the number of nodes in the network. Since, the number of hop counts required in larger area is $O(\sqrt{n})$, the capacity of multi-hop wireless network was found to be $O(\frac{n}{\sqrt{n}})$. P. Gupta et al.[5] demonstrated that there exists an upper bound on the global scheduling of the nodes by $\Omega(\frac{1}{\sqrt{n}\log n})$. Moreover, as the traffic of a communication system of a CPS converges towards the server side, the last mile in this communication chain turns out to be a bottleneck.[14] In a broadband system multiple fiber links solves this problem. However, a CPS should be designed strategically to take care of the problem of the last mile.

In summary, the physical layer, the data link layer and the network layer of a communication system impose their respective limitations on the maximum number of devices that could be supported, and thus make the overall scalability more challenging to handle.

2.3 *Scalability challenges with the middleware*

Middleware is a piece of software, which connects the infrastructure of CPS to the outside world without showing its presence to the end users. It gathers data, processes the data and infers relevant information, provides mechanism of storing them in a suitable database and makes the information available to the outside world systematically via. APIs (i.e. application programming interface). It streamlines the process of data acquisition, data management and API management. It is not pragmatic for a middleware to handle infinite infrastructure of CPS. Each aspect of the middleware imposes their individual limitations on the scalability such as following:

Abstraction: The end devices in the cyber physical system come from different vendors. Method of data acquisition, data model and format, the underlying hardware platform, methods of data conversion and communication mechanism are different. While abstracting the data and before presenting it to APIs, the middleware needs to deal with these kinds of heterogeneities that affect the scalability negatively. In summary, the scalability of the CPS is inversely proportional to the heterogeneity.

Data processing: The end devices in the CPS are supposed to generate data or function as an actuator. These data need to be processed collectively. Information inferred from one type of data is supposed to be used with other type of data for inferring another kind of information. For example, as discussed in Section 1, the relative humidity inferred from the sensors was used with the leakage detection to confirm whether the leakage is from AC or otherwise. Increased number of similar nodes imposes one kind of challenge for scalability. On the other hand, heterogeneity puts a different kind of challenge. It is required that these challenges are dealt with separately.

Dynamicity: Communication parameters are continuously changing in a dynamic CPS. Collective measures of node mobility, node failure, communication failure etc. are known as dynamicity of a network. While designing a middleware the eventual dynamicity that affects the scalability negatively must be taken into consideration. It has been proven that dynamicity of the network is $O(n^2)$ where n is the number of end devices in the network. Since dynamicity increases exponentially with increase in number of devices, it poses a bigger challenge for scalability.

Programming: All devices in the CPS cannot have the same kind of programming mechanism. There must be a provision that all components of

the middleware are customized to address diversity of programming mechanism of the connected devices.

Adaptability: Adaptability of a middleware to a diverse system depends on the adaptability of the data processing algorithms and other components of middleware. Issues of adaptability of a middleware increase with the increase in number of end devices and applications, and hence scale.

Security and other QoS: Security system puts additional limitation on the performance of a computing system, and therefore, a secured system is lesser scalable than unsecured system from computing point of view. On the other hand, QoS ought to degrade with scale. There must be a mechanism in the middleware to tradeoff between security, QoS and scalability.

2.4 *Issues of scalability at application servers*

So far we have analyzed respective issues of scalability in data acquisition systems and the middleware. Ultimately the data will arrive into the application server which hosts a number of applications built for data and alarm management and for generation of commands for actuators. It is prudent to analyze the role of user applications on the overall scalability, though at this stage data are already available in processed form so the information rendering doesn't seems to impose any challenge on the scalability at this stage. However, different applications require data in different forms.

Each CPS can support a limited number of data format and limited number of applications. Therefore, it is the diversity of applications which dictates the terms of scalability in the application server. Additionally, processed data should be stored in such a way that there is lesser computational effort required to feed the application. A particular application can handle limited number of query from application in fixed amount of time. Therefore, the way a query processing is designed impacts the scalability of application server and hence the same of the CPS. On top of this the analytics engine also plays an important role. Thus, there are three important factors affecting scalability (i) mechanism of storage (ii) mechanism of query processing by the applications and (iii) algorithms for analytics.

As far as storage is concerned data would be stored in different physical locations, and it is the load balancing algorithm which gives a seamless experience to the user through applications. Network stripping and Server Stripping are some of the techniques[9] in which underlying complexities of communication, authentication, and scheduling of data components stored in various physical locations are taken care.

Large deployment would result in peta-bytes of data. The current mechanism of storing data and application logic separately is not a best solution. A part of the processed data is required to be moved near applications, and another part of the application would be required to be move near data. It is really a trade-off between these movements which defines the overall performance; trade-off is done based on the cost of movement. There are two types of cost incurring to the host of the CPS: (i) communication cost of data transfer and (ii) the computational cost. At higher levels, a trade-off between the movement of data and the movement of application is equivalent of optimizing the cost of movement. Technically, it is a trade-off between communication and computation. If a CPS system is designed for peta-scale, then it will require 1000–10000 disks and thousands of computational nodes. With such a large number of disks and computational nodes, failure of the same would be inevitable. Redundancy is required to prevent data loss and provide seamless availability the way RAID5 disk arrays do today.[10] Though redundancy addresses the problem of failure, optimization of partition in temporal and spatial domains can always improve the performance.

In this section, we highlighted the challenges associated with scalability during data acquisition stage, communication stage, middleware stage and in the application server. The solution designer will not only have to deal with the individual limitations of different technologies but the overall limitations of the CPS arising due to the interaction of these components. In following sections we will discuss the relevant technologies developed for other purposes, which could address some of these challenges.

3 The Technologies for Addressing the Concerns

The technologies developed independently to solve other problems of IT and computer science might be used readily to address the challenges of scalability. From solution point of view, the challenges mentioned in last section can be broadly classified into two categories: (i) *The Diversities*: how to deal with diversities in end devices, platform and communication technologies and (ii) *The Volume*: how to deal with huge volume of data where each byte has

its own importance. We will discuss them in the forthcoming paragraphs.

3.1 *Dealing with the diversities*

In cyber physical systems, it is the data that are valuable for end users, Platform and communication technologies are ideally hidden from the end users, and hence are trivial. However, for a large deployment, mechanisms of extracting data from the communication packets coming from various devices are complex. In a couple of earlier efforts[3] and,[4] the problem of diversified communication system was solved by having multiple communication interfaces and corresponding software components. However, separate communication interface and software processing is not a scalable solution. To reduce the number of software components this problem was solved by K. Padmanabh et al.[3] In this approach, though physical communication, interfaces remained the same, however, the data packets emerging from different end devices were directed into a common queue. Since queue length is a natural limit for scalability and pragmatically a queue cannot have an infinite length, incoming packets ought to be dropped when already filled.

A proper buffer management policy ensures that the queue never overflows. In this example,[3] there is a filtering algorithm that filters the packets and then puts them into the queues of corresponding processing elements (PE) as depicted in Fig 3. The buffer management policy, even in worst possible case of packet arrival, ensures that the queue does not grow beyond a bound. Let us assume that there are N numbers of end devices transmitting data at the rate of '*r*'. If '*s*' is the speed of computing, then queue length developed in a time T would $l = NrT - sT$. Therefore, the queue should not grow faster than l/T.

Different kinds of "embedded software systems" which handle the process of transmission and reception of data packets of the end devices are also known as platforms. Additionally, the end devices might be using different communication standards according to the requirement of a cyber physical system. For example, in a CPS the end devices might be running on embedded Linux, tinyOS and Contiki using ZWave, Zigbee or WiFi as a communication medium. Whenever a data packet is created, the respective information of individual platforms and occasionally communication standards are embedded into the data packets. Thus, when end devices start pumping data to a common middleware the data packet should be processed according to platform and communication standards. Therefore, the equivalent number of processing elements (as depicted in Figure 4) should be created to download the data. Communication related overheads are required to be processed according to the respective communication standard to abstract network identity.

In this diverse environment of platforms and communication standards, seamless flow of information from end devices to the application through different components and processes is required. User should not feel any difference due to specific platform or standards; rather, it is required that the user should not feel the presence of diversity. The platform related abstraction can be done at the middleware.[3] However, journey of a data packet from the input queue till the user applications are different for different communication systems. Therefore, handling multiple communication protocols in the same software system is not straightforward. In an earlier effort,[11] this issue has been addressed partially. This product from Cisco known as Mediator is playing an important role in building management systems. Mediator can convert data packets of multiple communication technologies into IPv4 packets. For example, it has the capability to convert data packets of LonTalk, Modbus, BacNet into IP packets. Thus, the middleware and applications at the server are completely aloof from the complications of using multiple communication standards in a single system. Since, it has been proven that the internet protocol is highly expandable, challenges of scalability arising due to the diversity of communication technologies can be handled using a protocol translator that converts data packets of multiple networking technologies into IP packets.

Mediator has the special requirement of converting every protocol into IP. However, in other building management systems, it is an OEM that decides the protocol standards. The end customers might use devices of different standards in their

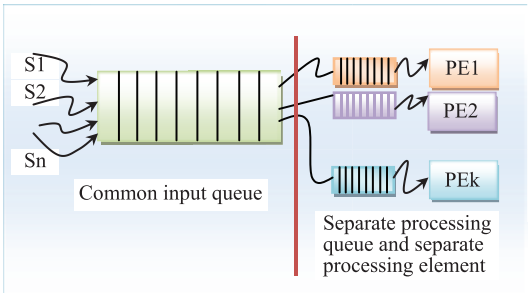

Figure 4: The common input queue for all sources and separate queues for individual processing elements (PE).

deployment to optimize cost and performance. For example, in building automation, a customer might use HVAC from OEM-A running on BacNet and fire safety system from OEM-B that runs on RS-485. If a common software/middleware has been planned to provide web services and it is required that fire safety system should interact with HVAC, then a protocol translator is required.

It is important to reiterate here that since the owner of a CPS would be more interested in performance and cost of the system, he wouldn't mind putting devices from multiple OEMs that runs on different protocols. In this situation, a protocol translator that translates any protocol to any other protocol is required. Since the owner might not be restricted to use IP only and might use BacNet as main underlying protocol would not work, in this situation, any protocol to IP as described in mediator[11] might not work. L. Malhotra et al.[12] provided a software solution of this problem. This is a protocol translator in disguise that deals with application layer packets only. It translates the application layer packets of one protocol into the same of other protocol without operating at lower communication layers. It was demonstrated that with the right configuration, the end application developers can develop application as if they are using just one set of protocol of their choice. For example, if BacNet, LonTalk, ISA-100 and Zigbee are being used in the system, then if the end developers decide to deal with only Zigbee then software system can be configured in such a way that all devices would appear as Zigbee devices. For example, a BackNet object would appear as a Zigbee device with the PAN number and node ID, and master slave architecture would be converted into a Zigbee Mesh. In this way, these devices running on the different communication standards can be incorporated into a CPS without affecting scalability.

Thus, in this section we discussed how diversity of end devices, embedded software platforms and communication standards should be handled for overall scalability of the CPS. We explained how a proper design of the lower layer of middleware that handles the incoming data packets would avoid loss of data in a large system. Subsequently, we discussed how a protocol translator would allow diversity in the CPS thus helping the system to scale without affecting performance.

3.2 *Scalability of data*

According to an estimate, there are already half a billion devices connected to internet in USA. Globally it is supposed to be 14 billion devices. In the same study it was found that by 2020 there would be around 100 billion devices connected to internet. If these devices produces same volume of data that needs to be processed and managed as described in section-1 then every device would be producing 4.5 MB of data on an average per day. So accordingly it would be 45000PB of data generated daily from these cyber physical systems globally. It is to be noted that we are discussing those devices that would be connected to internet; however, there would be additional devices which would not be connected to internet but still be pumping the data to the local servers. Cyber physical data are different from tradition multimedia data. Unlike multimedia data, following are the characteristics of a CPS data:

1. ***The significance of each byte:*** Each byte of data has its own significance. Loss of a byte means the loss of information. For example, information on temperature can be embedded in a single byte of data. Thus, due to this specific nature of CPS data it cannot be always compressed.
2. ***The significance of delay***: Time required to process a set of data is not only a performance measure, rather the *correctness* or *incorrectness* of the system can be defined using this parameter. For example, if a system is not able to process a data within a prescribed time limit then it can be termed as an incorrect system.
3. ***The historical importance:*** It is not only the instantaneous importance, a data can have historical importance and therefore current data can be combined with historical data to generate a unique report. Combination of historical data increases the chances of larger set of data getting processed together.
4. ***The collaboration:*** Data from multiple sources are required to collaborate among themselves, and collaborated data needs to be processed together. Thus, combination of multiple types of data may yield results which would not have been possible by processing them separately, thus giving a possibility of processing much larger data together.

The system which consumes multimedia can afford to lose few packets, they are NOT affected much by small delays and don't require to collaborate among different data. However, as explained above we need to understand CPS data in different way. Following subsection would throw some more light on how these data could be handled.

3.2.1 Role of big data analytics

Eventually in futuristic CPS more volume of data are required to be processed in minimum amount

of time. Since collaboration of data is required, therefore, distributed processing would not be always possible. Data are required to be there at a single site. It results in two types of challenges: (i) how these data would be shipped to the computing unit and (ii) how collaborative processing could be done efficiently. With appropriate gateways or local server end devices could be connected to the cloud directly. It has been proven that cloud infrastructure could be made scalable to any extent. There are technological supports available for processing of big data.

The above mentioned problems could be solved using cloud and big data framework. MapReduce framework has been evolved for big data. In MapReduce framework, the huge volume of data is essentially split into multiple chunks and processed separately in Mappers, and then the results are combined in Reducers. Splitter, Mapper and Reducer can be implemented in multiple instances of virtual machines in different forms in the cloud infrastructure. Though Mapreduce framework was originally developed for offline data analytics of Big Data, it has been proven that if the basic building blocks are used in adaptive way then near real time results can be produced.[12] Thus, with capability of MapReduce framework to produce results in bounded time, it can be used in offline as well as online analysis to support real time applications.

3.2.2 Relevance of metadata

The collection of all analytical results tagged into the raw data is known as "*metadata*" of a CPS. In many cases, interest lies only in the metadata, and thus main data could remain unimportant after the final processing. Thus producing *metadata* is an important steps in coping up with a Big Data and hence scalability. The query processing on the main data and further analytics could be built just with the *metadata*. Each analytics is supposed to generate a unique set of metadata. There could be two sources of metadata: firstly, the metadata generated at the source of along with the main data, and secondly, during the analytics of the data. In typical cyber physical systems, metadata can be generated at the end devices, or in the gateways or in the server. Analytics can run at all three places. There are different kinds of metadata generated at different moment of time during the journey of data packets from its source to the final user applications. They are broadly classified into three categories:

1. ***Descriptive metadata:*** It helps in facilitating resource discovery and its identification. Geographical origin of data, time stamps, and related information can be classified in this category. This type of metadata is required context information, which is further required to infer other relevant information from the raw data.
2. ***Administrative:*** It helps in resource management within a collection. URL/URI, types, source and location of data could be used for administrative purpose and hence belongs to this category.
3. ***Structural:*** This kind of metadata is generated after analytics. This is very important and completely dependent on end application.

First two types of metadata could be generated at source itself, however, structural metadata is required to be produced at the server. For the same set of data there could be different sets of metadata according to the requirement of applications.

3.2.3 The role of database in scalability of a CPS

A database provides mechanism of systematic storage, query processing and reporting. Query processing is a computing intensive process, and therefore can handle only a fixed amount of data. Therefore, a database plays an important role in scaling up a CPS. There are two strategies for scaling any application from database point of view:

1. ***The vertical scaling:*** For vertical scaling, the applications are moved towards larger computing infrastructure so that all computing and analytics could be done at the same place. Though it reduces the communication costs, the most obvious limitations of vertical scaling are its exponentially increasing cost with increasing scalability.
2. ***The horizontal scaling:*** It offers more flexibility but definitely at the cost of increased complexity. This type of scaling can be done in two stages: firstly by grouping data based on functions and subsequently spreading functions across databases.

As mentioned earlier, the data of a cyber physical system are supposed to be unstructured. A consistent schema cannot be created so that storing and query processing become easy. Moreover, in a typical database operation, one single transaction involves multiple changes in the state of different objects. Unless and until there is a change in all states, the transaction is not considered to be valid. Complexity of managing these intermediate states increases with increase in the volume of data.

Table 2: Comparison of different unstructured database from scalability point of view.

S. no	Database/Properties	Types	Scalability	Availability	Security	Manageability
1	Apache Couch DB	Document	2	2	2	3
2	Mongo DB	Document	2	2	2	3
3	Terra Store	Document	3	2	1	1
4	Dynamo DB	Key-Value	3	1	2	3
5	VoldMort	Key-Value	3	2	2	1
6	Amazon Simple DB	Key-Value	3	3	3	1
7	Google Big Table	Big Table	3	3	3	3
8	Cassandra	Key-Value	3	3	2	3
9	Hadoop Hbase	Big Table	2	2	2	2

3 = Best, 2 = Medium, 1 = Lowest.

Since data from a cyber physical system are unstructured data, the traditional RDBMS technique will not be effective. In recent times, with advancement in cyber physical system, the number of proprietary and open source database has been developed for unstructured data. Some of these databases that could be potentially used in CPS environment are Apache CouchDB, Mongo DB, Terra Store, Dynamo DB, VoldMort, Amazon Simple DB, Google Big Table, Cassandra and Hadoop Hbase. They are broadly classified into "*document style*", "*key value pair*" and "*big table*".

Apache Couch DB is a schema less document oriented database with primary goal to be highly scalable and fault tolerant. *The Mongo DB* has emerged as one of the best open source database for web applications supporting document style of database. It is closer to MySQL in the sense that it has query optimizer, ad hoc queries and customizable network layer. *Terra Store* is a new document oriented database. It provides elasticity and scalability without compromising consistency.

Dynamo DB is a key-value pair type of database available from Amazon. It supports distributed hash tables (DHTs), therefore it can provide lookup tables similar to key-value. *Voldemort* is being used by LinkedIn system. This is also a distributed key-value storage system. *Amazon Simple* DB has a web services interface to create and store multiple datasets query easily and return the results faster. *Cassandra* is a widely adopted and popular structured key-value based distributed database system. It uses Apache Dynamo for distributed database and the concept of big table for storing values within every column. This database optimizes column store and hence retrieves all the columns for a given row is low latency.

Google Big Table is a proprietary distributed database of google primarily meant for structured data. It is designed to scale easily to peta-bytes of data across thousands of commodity servers. Google's web indexing, Google Earth and Google Finance are based on this database concept. *Hadoop Database (HBase)* has evolved from google big table. However, it is an open source distributed column oriented store. Therefore, it has all good feature of big table, such that it provides RESTful web service, there is no single point of failure, and fault tolerance is incorporated in it. The random access of data is at par with MySQL. It is designed for large quantities of sparse data; however, it is not meant for high amount of binary data.

Choosing a particular database is very subjective and depends on many considerations. Table 2 presents the different attributes that might be relevant for a CPS and identifies how it is scalable with respect to each other.

Thus in this section, we studied about how to deal with diversities which impose a challenge for the scalability. Secondly we studied about different aspect of *metadata* that could help in reducing the volume of data to required information only. Finally we studied about databases that could handle huge volume of unstructured data getting generated out of a CPS.

4 Conclusion

In this paper we studied the different aspects of the challenges associated with the scalability of a cyber physical system. The study was divided into two parts: initially, the challenges of scalability associated with different building blocks of a CPS were identified and explained, and subsequently, the relevant technologies which could address these challenges were explained. It was found that challenges

exist in data acquisition systems, middleware, and at the application server. At each of these stages scalability faces challenges in the form of diversities, capacity limitations and interoperability. It was explained how an intelligent middleware with specialized scheduling and management can address the challenges of diversities of platforms. Additionally, it was suggested that a protocol translator could solve problem of diversities of the communication systems. We also identified that with right form of metadata and with cloud analytics with adaptive Mapreduce framework, the above mentioned challenges could be handled. It is not pragmatic to provide a universal solution of scalability. Each specific use case of CPS should be studied separately from scalability point of view, so that the issues are quantified to provide exact solution.

Received 9 July 2013.

References

1. Sunil Kumar Vuppala, Animikh Ghosh, Ketan A. Patil, Kumar Padmanabh, "A Scalable WSN Based Data Center Monitoring Solution with Probabilistic Event Prediction," aina, pp. 446–453, 2012 IEEE 26th International Conference on Advanced Information Networking and Applications, 2012.
2. Jinyang Li, Charles Blake, Douglas S.J. De Couto, Hu Imm Lee, and Robert Morris, "Capacity of ad hoc wireless networks", In Proceedings of the 7th ACM International Conference on Mobile Computing and Networking, pp. 61–69, Rome, Italy, July 2001.
3. Kumar Padmanabh, Lakshya Malhotra, Vanteddu Adi Mallikarjuna Reddy, Amrit Kumar, Sunil Kumar Vuppala, Sanjoy Paul: MOJO: A Middleware That Converts Sensor Nodes into Java Objects. ICCCN 2010: pp. 1–6.
4. Intel Connected Services Gateway Reference Design: http://download.intel.com/embedded/applications/connecteddevices/323866.pdf
5. P. Gupta and P.R. Kumar. The Capacity of Wireless Networks. *IEEE Tiransactions on Information Theory*, 46(2): pp. 388–404, March 2000.
6. K. Römer. "Programming Paradigms and Middleware for Sensor Networks," *GI/ITG Fachgespraech Sensornetze*, Karlsruhe, 26–27 Feb, 2004.
7. Y. Yu, B. Krishnamachari, and V.K. Prasanna. Issues in designing middleware for wireless sensor networks. IEEE Network, 18(1), 2004.
8. Jim Gray, David T. Liu, Maria Nieto-Santisteban, Alex Szalay, David J. DeWitt, and Gerd Heber. 2005. Scientific data management in the coming decade. *SIGMOD Rec.* 34, 4 (December 2005), pp. 34–41. DOI=10.1145/1107499.1107503
9. Watson, Richard W. "High performance storage system scalability: Architecture, implementation and experience." In *Mass Storage Systems and Technologies, 2005. Proceedings. 22nd IEEE/13th NASA Goddard Conference on*, pp. 145–159. IEEE, 2005.
10. Dick Korea, "RAID Configuration and Parity Check", DTIDATA http://www.dtidata.com/resourcecenter/2008/05/08/raid-configuration-parity-check/
11. Cisco Network Building Mediator, http://www.cisco.com/en/US/products/ps10454/index.html
12. Lakshya Malhotra, Kumar Padmanabh, Sanjoy Paul, "System and method for facilitating communication between different protocol stacks via virtual communication devices", US Patent No-20130007199, http://www.google.com/patents/US20130007199
13. Fen Zhang, Cao Junwei, Xiaolong Song, Hong Cai, Cheng Wu, "AMREF: An Adaptive MapReduce Framework for Real Time Applications" 9th International Conference on Grid and Cooperative Computing (GCC), pp. 157–162, 2010.
14. WikiPedia, "The Last Mile" https://en.wikipedia.org/wiki/Last_mile
15. Szewczyk, Robert, Joseph Polastre, Alan Mainwaring, and David Culler. "Lessons from a sensor network expedition." In *Wireless Sensor Networks*, pp. 307–322. Springer Berlin Heidelberg, 2004.

Dr. Kumar Padmanabh is working in Corporate Research lab of Robert Bosch Engineering and Business Solution Bangalore, where, he is currently leading various initiatives of "Internet of Things". Prior to this he worked in General Motor's Research Lab in Bangalore in intra-vehicular communication network. Earlier he worked in Infosys Labs for five years in the technology domain of Wireless Sensor Network in the area of enterprise building automation especially on the various aspects of energy management. He received a degree of Ph.D. in Sensor Networking from IIT Kharagpur in year 2006. He is specialized in Internet of Things, Sensor Network and Cloud Computing applicable to home and enterprise building automation. So far he has published more than 35 research papers and a book chapter on Zigbee. He has been the lead inventor of 15 US and Indian patent applications. He is a winner of grand challenges for technologists-2010 award given by MIT Technology Review and Infosys Thought Leadership Award of year 2010. He was one of the finalist of Indian Mathematics Olympiad-1994. In his leisure time he writes fiction and so far he is able to publish two books.