Control-as-a-Service from the Cloud: A Case Study for using Virtualized PLCs

Omid Givehchi[1], Jahanzaib Imtiaz[1,2], Henning Trsek[1], and Juergen Jasperneite[1,2]

[1]inIT - Institute Industrial IT, Ostwestfalen-Lippe University of Applied Sciences, D-32657 Lemgo, Germany {omid.givehchi,jahanzaib.imtiaz,henning.trsek,juergen.jasperneite}@hs-owl.de
[2]Fraunhofer IOSB-INA Application Center Industrial Automation, D-32657 Lemgo, Germany

Abstract

Cloud computing has recently emerged as a new computing paradigm in many application areas comprising office and enterprise systems. It offers various solutions to provide a dynamic and flexible infrastructure to host computing resources and deliver them as a service on-demand. Since industrial automation systems of the future have to be adaptable and agile, cloud computing can be considered as a promising solution for this area. However, the requirements of industrial automation systems differ significantly from the office and enterprise world. In this paper we describe a case study that implements a concept of PLC as a service within a cloud based infrastructure and provides a performance evaluation with respect to legacy PLCs.

1. Introduction

Future automation systems should meet fast growing market demands by providing agile and flexible production lines. In-addition, competitive production costs and scalability of the automation system would be an important topic for the industry 4.0. Technical processes are often consist of distributed subsystems controlled by individual PLCs [2]. Each new hardware PLC brings additional costs and maintenance efforts. The virtualization of industrial controller functionality (PLC-as-a-service) [9] could be a promising approach. Though, some inherent challenges such as accessibility and security needs to be addressed for industrial automation applications [4]. In the presence of a comprehensive resource distribution policy, multiple instances of virtual PLCs can be created and allocated to control underlying physical subsystems [10].

The main research question for this work arises as: *"Is it possible to implement a control functionality as a service, employing a cloud infrastructure with a comparable performance guarantee as that of a dedicated hardware based PLC?"*

Virtualization is an enabler technology for cloud computing. Additionally, some other features such as service oriented paradigms, security [12] and resource management are employed in cloud computing to improve the virtualization core [10, 9]. In this paper we present a case study that investigates the potential of cloud computing (a technique mainly coming from IT industry) for control level applications. This is with a focus on the performance aspects, for being an important factor for the industrial automation systems.

This paper is organized such as: section 2 provides some background and state of the art introduction, which includes an overview of some possible cloud computing architectures for industrial automation systems. An approach to deliver control-as-a-service is described in section 3. Followed by a case study and evaluation results in section 4. The last section concludes the work and briefly address future work in this area.

2. Background

During recent years, industrial automation witnessed the new demands and trends in different areas. Field devices become more intelligent by embedding new functionalities inside IO devices or sensors and actuators. Similarly, industrial communication has been developed significantly by improving communication standards and protocols, e. g. Profinet [7] and OPC in the control and field level which makes devices more connected. These trends might require the definition of new automation architectures differing from the current hierarchical automation pyramid. Among these newly proposed architectures, Vogel-Heuser et. al. [15] introduced a new architecture as global information architecture for industrial automation.

This architecture is shown in Figure 1. In their proposal, time-critical parts of control and field levels still exist in a traditional way based on the well-known automation pyramid. However, the upper levels have been changed. Since these are basically providing non-physical functions and services, it is possible to migrate them into the cloud.

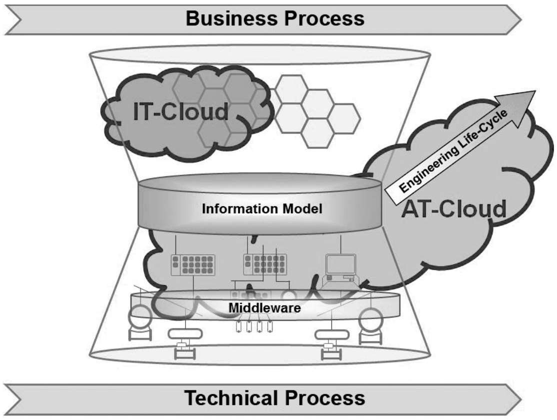

Figure 1. Global Information Architecture with use of Cloud Computing [11], based on [15]

The proposal in [15] described the new model as a motivation and introduction for future automation systems. They promised an improvement in engineering life-cycle and more flexible operations. However, there is not much research work related to cloud computing for automation, especially in control and field levels [5]. The cloud based applications are often operate in public domain. Since automation systems always have stringent reliability requirements, a private cloud model could be a solution. Unlike a public cloud, a private cloud has no connectivity with Internet and public networks and is accessible only from local network. Hence, it offers the highest degree of performance, privacy and security comparing with other cloud types [1].

Several private cloud products are currently available from different vendors. Among them, VMware's vCloud suite was selected for our implementation, since it has some features such as VMDirectPath I/O pass-through and VMCI [3] which make it compatible with industrial networks. These features will be discussed further in the next section.

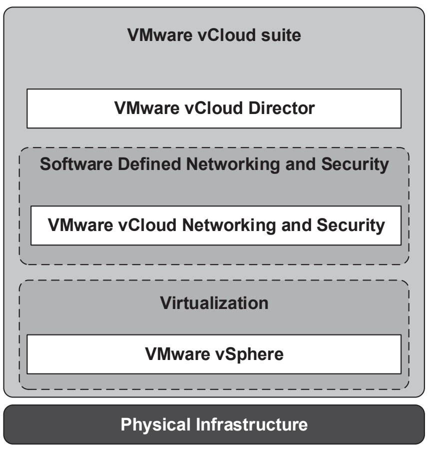

Figure 2. VMware's vCloud Suite Architecture [14]

VMware's vCloud suite architecture with components used for our implementation is shown in Figure 2. As illustrated, vCloud suite is a combination of different VMware components to offer a package of a cloud platform. VMware vSphere plays the role of virtualization hypervisor while vCloud Networking and Security provides the connectivity and security for the VMs. vCloud Director is used to pool the prepared resources, automatic provisioning of them, and deliver them to the users as service when they demanded [14].

3. Cloud-based Approach

The proposed solution approach aims at offering control-as-a-service for an industrial automation use case as illustrated in Figure 3 for a process module n. Based on application requirements for each specific module, PLC could remain in the shop floor as a physical device traditionally as shown for module 1 to gain more reliable control functions. Alternatively, PLC can be implemented as a virtual entity and be delivered to the field as a service from a cyber-physical system via the network as shown for module n.

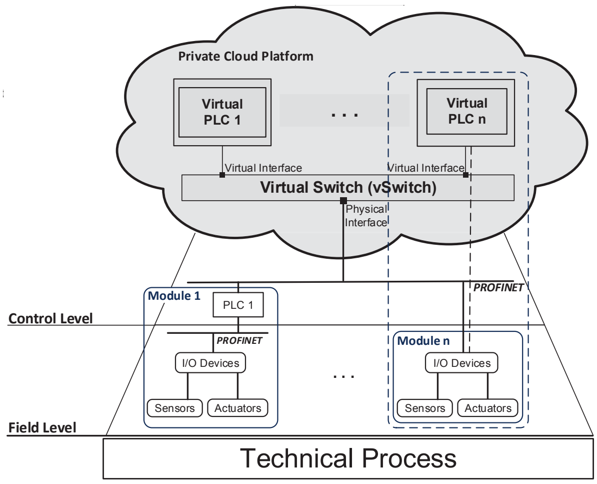

Figure 3. A Generic Cloud-based Control Approach

This system will be realized as a private Cloud platform with capability to offer virtual PLC appliances using VMware's vCloud suite [3] solution discussed in section 2 which is connected to the Profinet real-time Ethernet system (RT) via physical interface. In order to increase bandwidth, the number of physical interfaces could be increased by connecting more network adapters to the hypervisor computer. The used virtual switch (vSwitch) is directly managed by vCloud Networking and Security component shown in Figure 2. Network rules and policies such as access permissions for virtual machines defined for the cloud will be applied by vSwitch. For enabling industrial communications such as Profinet on vSwitch, Virtual Machine Communication Interface (VMCI) [13] is enabled for the adapter driver of VMs. VMCI is a high-speed interface which boosts the communication between VMs and the host computer [13].

A virtual PLC appliance is connected to each virtual interface. Technically, each appliance includes a virtualized operating system which runs a PC-based software PLC. In our implementation, Microsoft Windows 7 32-bit is selected as guest operating system for virtual machines since it is recommended by the PLC vendor. PC WORX SRT [8] Software PLC is running on VM to handle control tasks. This product is designed to support soft real-time(RT) control applications with a minimum task interval of 2 milliseconds. The installation and configuration of VM performs manually once on a vCloud Director template. This template will be automatically provisioned on vSphere by vCloud Director numerously to provide new instances of virtual PLC as it demanded by plant engineers.

4. Case Study

The evaluation of the proposed cloud platform is performed on a PC hardware with Intel(R) Core(TM) i5-2400 processor having four 3.10GHz CPU cores with enabled hardware virtualization support (Intel VT-x), 8 GB of RAM, 1 TB HDD, and three 100Mbps network adapters. For each VM, a single core 3.10GHz CPU, 1GB RAM, and 20 GB of SCSI Virtual HDD is allocated. VMCI is enabled for all the VMs. A clean installation of Windows 7 32-bit is running on each VM. To compare the evaluation results with a legacy setup, a Phoenix Contact hardware PLC (ILC 350 PN) is used as reference benchmark.

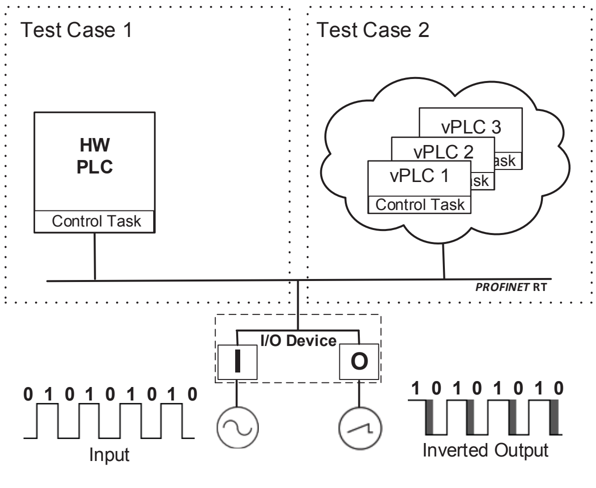

Figure 4. Case study setup

Both hypervisor computer and hardware PLC are connected to a Profinet RT network including an I/O Device (Phoenix Contact TPS-1 Board) as drawn in Figure 4. A IEC 61131 project is developed to read the input of the I/O device generated by a signal generator periodically with a fixed interval of T_I and to write the inverted value to the output. This task repeats cyclically in determined time intervals and the output will be measured and recorded by an oscilloscope. In Figure 5, the input signal interval is shown with T_I while output interval is shown with T_O. Typically, outputs will be influenced by some delays and a variation for output called ΔT will be observed for each output as shown in equation (1).

$$T_O = T_I + \Delta T \tag{1}$$

Since in this setup, network is not fully synchronized, cycle update time should be considered when frames start to be transmitted on the network. This time value is shown as T_{CD} in equation (2) and is the Profinet cycle update delay. This delay would be $2T_{CD}$ in the worst case where a frame waits for a whole cycle time at its upstream and downstream times. The T_{CD} of devices in the setup is maximum 1 ms in the worst case. $T_{Control}$ is the control task interval time plus execution time performed by the PLC to process each control task. The minimum supported value for the used PLCs is 2 ms. Due to simple inverting operation, execution time expected as a small value comparing with entire $T_{Control}$ value. T_N is the network delay for a frame to send over the network. In our case, each Profinet frame is 64 Bytes and its T_N is 7 μs which is a small value. The sum of these delays will be defined as End-to-End delay for the system as shown in equation (2). This delay will cause the mentioned variance ΔT for each output.

$$T_{E2E} = 2T_{CD} + T_{Control} + 2T_N \tag{2}$$

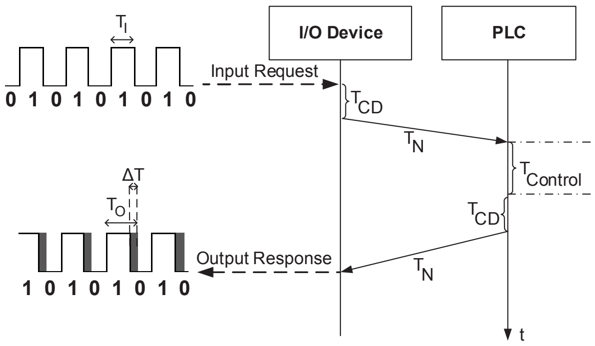

Figure 5. Sequence diagram of measurement

The mean value of ΔT for N experiments defined with $\overline{\Delta T}$ in equation (3) is considered as a metric to evaluate the performance of the system between a hardware PLC and a cloud-based virtual PLC. To ensure that the results are not highly variable due to a short measurement run, the experiments has been recorded for N=1000 times with a confidence of 90%.

$$\overline{\Delta T} = \frac{1}{N} \sum_{j=1}^{N} \Delta T_j \tag{3}$$

As shown in Figure 6, for the first test scenario, a hardware PLC is used to run the control task and $\overline{\Delta T}$ is recorded for different input intervals. The same test case is done for virtual PLC where three instances are running on the cloud platform simultaneously to control three different processes and output for one of them is captured to calculate the $\overline{\Delta T}$. Results of the evaluation show a reduction of performance for cloud-based scenario comparing with a hardware PLC. Especially for shorter input intervals the workload of the system will increase numerously, as it is observed from CPU usage of PLCs and network traffic, the $\overline{\Delta T}$ value will be higher which shows less performance for the system. The performance difference between virtual PLC and hardware PLC is basically a function of resource sharing on the cloud platform. These shared resources mainly include computing and computer I/O resources which are allocated to separate virtual entities from the hypervisor concurrently. This will result a performance overhead for the running services on the VMs [6] as shown in in Figure 6 with higher $\overline{\Delta T}$ value for virtual PLC. However, the performance reduction could be acceptable for soft real-time applications since intervals for changing sensor values or T_I is higher for this kind of applications. Furthermore, for hard real-time applications, there are some available real-time hypervisors which can be replaced by our used non real-time hypervisor to guarantee a high performance resource sharing [16] that will be focused by future works.

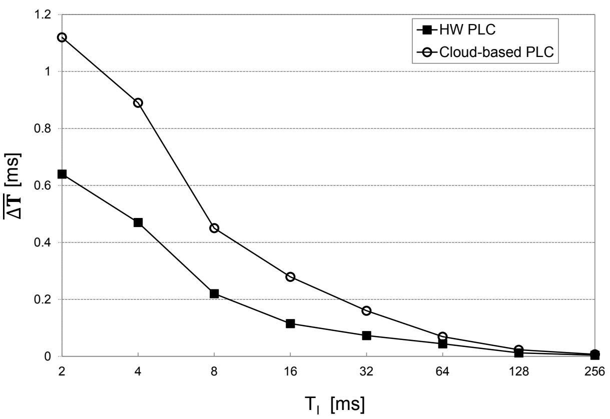

Figure 6. Performance comparison between Hardware PLC and Cloud-based PLC

5. Conclusion and Outlook

In this paper, a virtual PLC solution based on cloud computing has been analyzed in order to estimate their potential for a deployment on the control level of automation systems. The analysis is based on an implementation of a cloud-based PLC for the control level. The performance of the proposed solution is evaluated in a case study and compared to a legacy PLC. Even though the comparison of the results shows a lower performance of the cloud-based system, the obtained difference is less than 3 ms. After our first observations, the solution looks promising for soft real-time applications. However, more experiments from different perspectives are needed, especially in terms of security issues.

Our future research will mainly address hard real-time cloud platforms by using available real-time hypervisors e. g. RT-XEN [16] as virtualization component of the cloud. Furthermore, the achievable performance with respect to the number of virtual PLCs will be investigated.

References

[1] R. Buyya, J. Broberg, and A. M. Goscinski. *Cloud computing: Principles and paradigms*, volume 87. John Wiley & Sons, 2010.

[2] P. Gaj, J. Jasperneite, and M. Felser. Computer communication within industrial distributed environment a survey. *IEEE Transactions on Industrial Informatics*, 9(1):182–189, Mar 2013.

[3] S. Gallagher. *VMware Private Cloud Computing with VCloud Director*. John Wiley & Sons, 2013.

[4] L. Garber. The challenges of securing the virtualized environment. *Computer*, 45(1):17–20, Jan 2012.

[5] O. Givehchi, H. Trsek, and J. Jasperneite. Cloud computing for industrial automation systemsa comprehensive overview. In *Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on*, pages 1–4. IEEE, 2013.

[6] N. Huber, M. von Quast, M. Hauck, and S. Kounev. Evaluating and modeling virtualization performance overhead for cloud environments. In *CLOSER*, pages 563–573, 2011.

[7] J. Jasperneite and J. Feld. PROFINET: An Integration Platform for Heterogeneous Industrial Communication Systems. In *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*, volume 1, pages 8–pp. IEEE, 2005.

[8] PhoenixContact. Controller - pc worx srt - 2701680 datasheet, 2013.

[9] L. Ren, Y. Zhang, Y. Luo, and L. Zhang. A virtualization approach for distributed resources security in network manufacturing. In *Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on*, pages 1524–1528. IEEE, 2010.

[10] M. Rosenblum and T. Garfinkel. Virtual machine monitors: Current technology and future trends. *Computer*, 38(5):39–47, 2005.

[11] J. Schlick, P. Stephan, and T. Greiner. Kontext, Dienste und Cloud Computing - Eigenschaften und Anwendungen cyber-physischer Systeme. *atp edition*, 04, 2013.

[12] S. Subashini and V. Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1):1–11, 2011.

[13] VMware. Getting started with vmci. 2007.

[14] VMware. Vmware vcloud suite datasheet, 2012.

[15] B. Vogel-Heuser, G. Kegel, K. Bender, and K. Wucherer. Global information architecture for industrial automation. *Automatisierungstechnische Praxis (atp), Oldenbourg-Verlag, Muenchen*, 2009.

[16] S. Xi, J. Wilson, C. Lu, and C. Gill. Rt-xen: towards real-time hypervisor scheduling in xen. In *Embedded Software (EMSOFT), 2011 Proceedings of the International Conference on*, pages 39–48. IEEE, 2011.