

# Contribution to the performance evaluation of decentralized material flow controls on the example of a control prototype

DIPL.-INFORM. TOBIAS R. MAYER

LEHRSTUHL FÜR VERTEILTE INFORMATIONSSYSTEME, UNIVERSITÄT PASSAU

DIPL.-ING. SERGEY LIBERT

LEHRSTUHL FÜR FÖRDER- UND LAGERWESEN, TECHNISCHE UNIVERSITÄT DORTMUND

PROF. DR. MICHAEL TEN HOMPEL

LEHRSTUHL FÜR FÖRDER- UND LAGERWESEN, TECHNISCHE UNIVERSITÄT DORTMUND

The new control paradigm Internet of Things (IoT) in intralogistics propagates highly modular and flexible decentralized control systems based on agent technology and radio-frequency identification (RFID). The productive application of such systems demands the fulfillment of several requirements, which exist in the industrial reality of the facility logistics. The real-time requirements in decentralized control systems are examined within the scope of the Collaborative Research Center (CRC 696) “Logistics on Demand” (sub-project C5 – “Real-Time Logistics”). The systematic performance analysis of legacy and new controls is an important aspect of this research work. In this paper an approach for the evaluation of the performance of decentralized control is proposed. The collection and analysis of performance data is facilitated by self-developed measuring instruments.

## 1. Introduction

Recently applied research in the field automated material handling shows a strong inclination to developing decentralized control concepts (see [Win08], [HLS06] and [Wil06]). Improved scalability and adaptability as well as the reusability of system components (see [Lor03], [BSc01]) are often mentioned as the system advantages to be expected. Here increased profitability is to be guaranteed by shorter start-up periods and low effort expandability (see [GCK08]). However there is a gap between the vision of a desired decentralized material flow control and the realization of this vision in industrial practice. The use for performance evaluation instruments, proposed in this paper, helps to close this gap.

### 1.1. Internet of Things in Intralogistics

The control paradigm *Internet of Things in intralogistics* (see [GtH10]) propagates a decentralized approach to control material handling systems. The transported goods themselves take over the control and use the transport capability as well as other capabilities of the material handling plant. Modern technologies like radio-frequency identification (RFID) are used, having relevant information for transportation on the RFID-tag of the corresponding transportation good. A possibility to realize such distributed control systems is the usage of software-agents. The control system examined in this work is a multi-agent-system, where unit load agents are negotiating with each other and with conveyor agents to get access to plant resources. The concept provides for the distribution of control hardware and software along the mechanical components of a conveyor plant. This distribution is meant to produce a high modularity of the control. However, this makes additional demands to real-time capability and the performance of responsible control components.

### 1.2. Real-Time Problems and Control Performance as a Research Topic

The real-time capability of material flow control is an important requirement which is in contrast to the decentralized control principle (see [HLL07], [Ste03]). The reason for this confrontation lies in the nature of distributed systems with decentralized data management and processing. In these systems simple decisions can be made on the basis of local information. The duration of the decision making directly depends on the implemented algorithms and the computer performance. Furthermore, complex decisions like route search or the planning of transport resources require an additional communication among the system participants. Therefore, the duration of the decision finding also depends on the performance of the communication network and the used communication protocols.

For distributed data processing more and more industrial compact computers on PC-basis (IPC) are used. In contrast to a PLC-based control<sup>1</sup>, the real-time capability of such systems cannot always be guaranteed. In case of an IPC-based distributed control the reaction time of the control system depends on many factors. Among them is, for example, the performance of the individual computer as well as the complexity of the realized control functions or, in other words, the number of running programs. To equip the control hardware with a performance reserve would be a remedy but would also cause unreasonably high procurement costs. Undersizing on the other hand, can lead to the control system not being able to fulfill its tasks. In this context the determination of the performance of legacy and new planned material flow controls is becoming an important task. The heterogeneity of potential decentralized solutions and the broad range of hardware do not make this task any easier.

### 1.3. Performance of Material Flow Control

Due to the versatility of the term performance (see [Svo76], [Oss73], and [Fer78]) a universal definition has hardly any practical relevance. Our work puts the performance of decentralized computer and software systems in material flow control into the focus of attention.

In practice the control performance is often seen as the control quality with regard to the system function, which has to be implemented. Here, the maximal throughput of a material handling plant is a benchmark for evaluating and comparing the control quality and thus the control performance (see [Gud05]). In this case consideration of the control performance is limited to the analysis of the quality of implemented control strategies. In the planning phase the evaluation of the control strategy can be done by means of simulation (e.g. [Rad00], [RFo08]). The examination of the performance of a realized control does not take place until the integration test or even during operation.

Evaluating the performance of individual control components might lead to control performance to be seen from a different angle. Here the performance values for industrial computers and communication systems, such as computing performance or network throughput, are calculated experimentally (see [TVa99], [Kle96], [LMT99], and [Ros02]). However, conclusions regarding the quality of the whole control system are missing.

For a holistic consideration of control performance it is useful to establish a connection between time factors of control hardware and software and the quality of control decision. Within the scope of the Collaborative Research Center 969 “Logistics on Demand”, this task is dealt with in the part project C5 –“Real-Time Logistics”<sup>2</sup>. The developed Real-Time-Logistics Model (RTL-Model) provides a coherent representation of time factors in conveying plant mechanics and in the corresponding control system (see [LHN07], [LNH08]). The input values for this model can be received from data sheets of concrete automation devices or be measured under real operational conditions.

### 1.4. Goal and Approach of this Work

The current work represents the continuation of the research plan in the context of the “Real-Time Logistics” project. The goal of this work is to develop an approach to determine and analyze time factors of control hardware and software in real automation systems. A special challenge is presented by the fact that material flow systems as well as the corresponding controls are highly individual. This fact impedes a quantitative performance comparison of different system implementations and is motivation for this work.

The later introduced approach shall not only be used for cross-architectural performance evaluation but also for identification of fundamental problems in distributed control systems. This does not mean logistic objectives such as throughput of transported goods but rather performance aspects of the control system itself. The material control system of the test-bed system uses not necessarily real-time capable technologies like Ethernet. Using such technologies in domains with time demands can lead to delays, which may endanger the reliability of the control system.

The structure of this paper is geared to the approach used for performance analysis of productive software systems. For determining the control performance an experimental object has to be found first. Here the control prototype, which was developed according to the principles of Internet of Things in intralogistics, is used as such an object. The experimental object is presented in the following chapter. After that the experiment planning is explained in single steps. This explanation is followed by the introduction of the self-developed tool which is used for collecting measured data and for visualizing the results. Subsequently the measurement results are discussed. A résumé and a forward look on the continuation of this work conclude the paper.

<sup>1</sup> PLC = programmable logic controller

<sup>2</sup> <http://www.sfb696.uni-dortmund.de>

## 2. The Test-Bed System

The control to be examined is installed in the conveyor plant for light transportation good at the Chair of Materials Handling and Warehousing<sup>3</sup> (FLW) of TU Dortmund. Below both systems – plant and control – are briefly introduced.

### 2.1. The Decentralized Material Handling System

The system to be controlled is an experimental plant used for testing different decentralized control concepts (figure 1). The plant is divided into two levels which are connected with each other. It consists of overall 37 powered conveyors which are equipped with approx. 80 sensors for control purposes. The total length of the conveyor lines amounts to more than 120 meters.

The unit loads are multi-purpose containers equipped with RFID-tags for identification. Ten identification stations at all important decision points of the plant guarantee non-contact writing and reading of the control relevant data<sup>4</sup>.



Figure 1: Continuous handling plant of the Chair FLW of TU Dortmund

The embedded control devices are industrial PCs (IPCs) with Cyrix Geode CPUs (266 MHz) and a RAM of 128 MB. The data storage of 1024 MB is realized by a Compact Flash memory card. The seven of such IPCs are allocated close to the technical process, along the conveyor lines. They are connected with each other via Fast-Ethernet (100MBit/s) and take over the part of PLC in classic control architecture. Sensors and drives are connected with the control devices via the K-Bus-System. A Linux operating system with a 2.4 x Kernel and real-time extension (Real-Time Application Interface, RTAI) is used. The utilized operating system enables the realization of control programs in high-level program languages like Java or C++.

### 2.2. Material Flow Control Package Royale

*Package Royale* is a decentralized control system, which realizes the vision of Internet of Things in intralogistics in a prototype implementation. Being developed by S. Feldhorst at the chair FLW of TU Dortmund, its architecture has been partially published in [FLH+09]. The control software is created in Java and bases on standard technologies like XML<sup>5</sup> and Web Services (WS). Furthermore it is characterized by a high degree of modularity and flexibility. For this purpose two different kinds of control components appropriately represent plant control and material flow control tasks.

A plant control component realizes the core functions of a particular conveyor element. These core functions (collecting sensor events, setting the conveying direction, etc.) are available as services to the other control components via Web Service. All Web Service calls are processed using the DPWS library<sup>6</sup> (Device Profiles for Web Service), which became OASIS standard in 2009.

Operating strategies are implemented in decentralized material flow control components, realized as software agents. Such an agent calculates control decisions and uses plant services to achieve its control goals. In the control system *Package Royale* one control agent is associated with each conveying element or transportation good. The operation of conveyor agents is cycle-oriented. In one control cycle the sensor states are checked, a

<sup>3</sup> <http://www.flw.mb.uni-dortmund.de>

<sup>4</sup> The RFID-Equipment is a UHF-system (865 MHz). The RFID-tags applied to the containers are passive transponders with a storage capacity of 512 Bit.

<sup>5</sup> eXtensible Markup Language (XML) is a meta language for the description of structured information.

<sup>6</sup> <http://www.ws4d.org>

control decision is made and instructions are sent to the corresponding plant control component. The duration of the control cycle defines the (local) reaction time of such a conveyor agent.

The runtime environment of control agents is called *computation device*. Whereas the components of the plant control always work on the relevant IPCs, the computation devices can be distributed among the IPCs or be centrally installed on one computer. Due to the low computer performance, initially only services concerning plant control are installed on the IPCs. The computation devices, on the other hand, run on a standard-PC with a 1866 MHz Pentium-M CPU und 1024 MB random access memory. In the following this computer is called agent computer. The same computer is used as a measuring station for all our experiments.

### 3. Systematic Approach

Jain proposes a systematic approach for performance evaluation which consists of several sequential steps [Jai91]. According to this procedure, after the determination of an experimental object, metrics have to be identified. In our case these metrics represent the quantifiable performance values of the control system and are described below.

#### 3.1. Metrics for Evaluating Control Performance

Metrics are quantifiable characteristic values used for evaluating and comparing different systems. With respect to the current investigation a distinction is drawn between metrics of the computer system and metrics of the control system.

*Metrics of the computer system* characterize the utilization of computer resources. They includes the degree of utilization of processors, random access memory and the network bandwidth. These metrics describe the utilization of a particular computer system while the control program is running. For a detailed utilization analysis the differentiation of the metrics with respect to individual executed computation tasks is necessary.

*Metrics of the control system* comprehend control-specific performance values. To describe and quantify the performance of material flow controls the term local reaction time is introduced. The local reaction time covers the time necessary to fulfill a single control task in a local decision node. The time period starts with the first occurrence of such a demand, includes all control-relevant activities and ends with the realization of the control decision. Thanks to a high degree of abstraction, this metric can be used for a number of control applications.

In the case of performance evaluation of a control software, the local reaction time covers the described period with the exclusion of the technical implementation (e.g. switching a drive or setting a direction). In this case the local reaction time is structured as shown in figure 2.

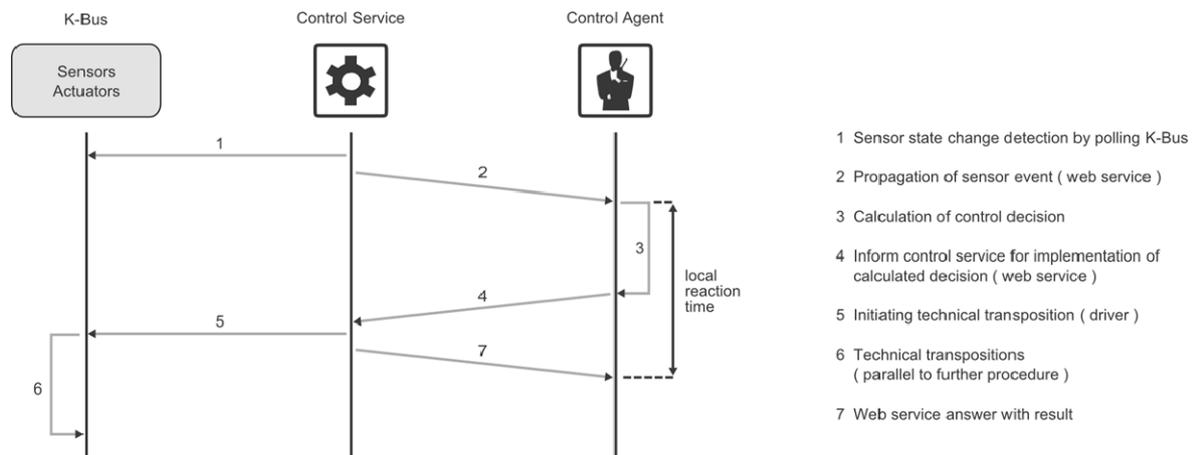


Figure 2: Structure of local reaction time

Figure 2 illustrates that, in case of a cyclic control procedure, the duration of the processing cycle corresponds with the local reaction time. In the following the term local reaction time is used synonymously with the duration of the calculation cycle of a control agent. For registration of the local reaction time the control agents have to remember the cycle period and to make the information available to the measuring system. The structure of the corresponding experiment is describes in the next paragraph.

#### 3.2. Experiment Structure

The structure of the experiment for gathering measurement data is characterized by the decentralized structure of the control which is to be examined. The measuring system is structured in two hierarchy levels (see figure 3).

The utilization values are gathered by means of data gathering software which is active on the IPCs as well as on the agent computer. Evaluation software, which is connected via network, periodically gathers the measurement data, enables visualization and supports the evaluation of the data. The data gathering pool and the evaluation pool are both part of the system *Monitor Royale*, a self-development, which is described in detail in paragraph 4. The interface between control agents and data gathering pool is realized via file system. A control agent enters the time values in a file. The data gathering pool reads the data every 1000 ms and additionally gathers the utilization values of the particular computer system. Each of these measuring objects comprise the aforementioned utilization rates and, due to variable agent cycle durations, a list of local reaction time values. Individual measuring objects are clearly marked with identification numbers, the timestamp of the executed measurement is used as such.

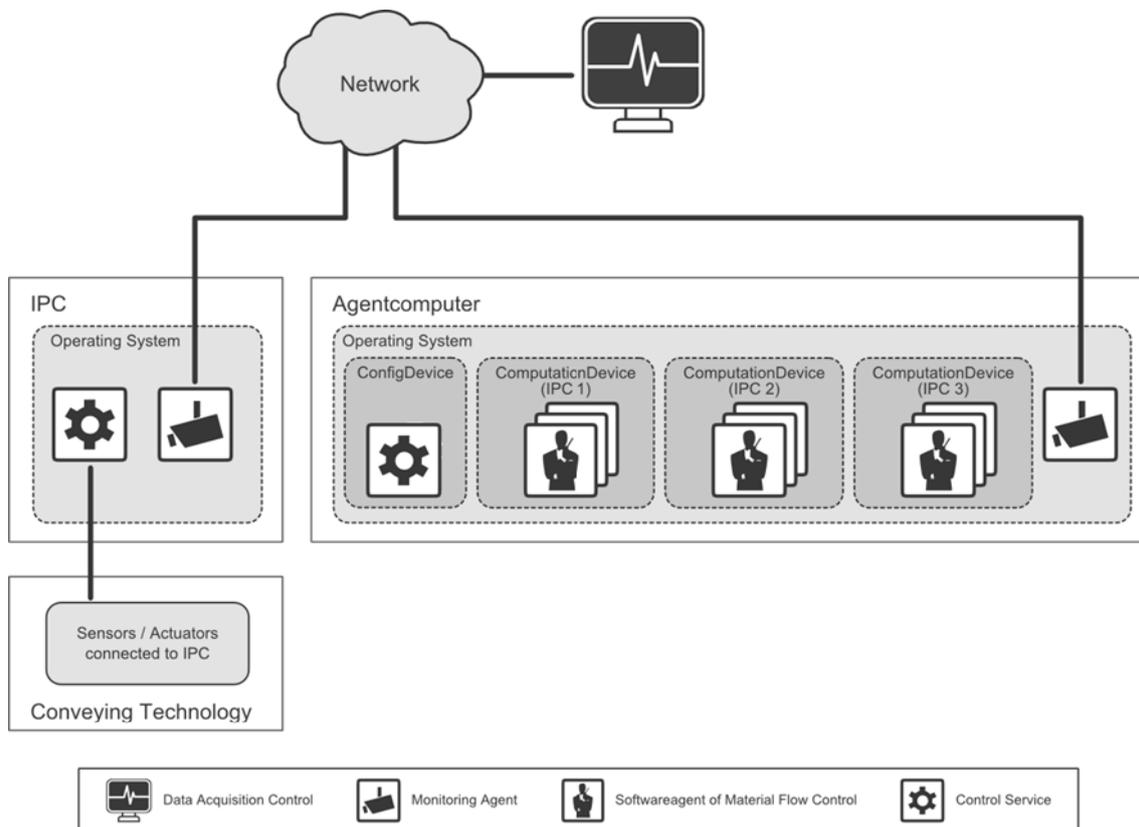


Figure 3: Experiment Structure of Measurement Data Collection

### 3.3. Examination Scenarios

For gathering load and performance values, examination scenarios have been developed, which represent typical core tasks of (decentralized) material flow control. These scenarios can be realized in different systems and be used for performance comparison. The number of active control entities in a system and the local throughput (number of transported piece goods per time unit) is used as a variable factor.

- *Scenario 0: Performance of Monitoring System*  
The measuring system itself can create a load in a computer system. In this scenario adequate performance values are determined, so that they can be taken into account for the following scenarios.
- *Scenario 1: Control Utilization Rate in Idle Mode*  
In this scenario the system behavior without a workload is examined. Decentralized control entities are active but there is no throughput.
- *Scenario 2: Start of Control Entities*  
In a decentralized control, control entities can be added to the system separately or in groups. In this scenario only a partial quantity of control entities are active at a particular time. At runtime a predefined number of control entities is activated and the utilization rate is determined.
- *Scenario 3: Switching Off Control Entities*  
This scenario is similar to scenario 2. With the exception that here a predefined number of control entities is deactivated while the system is still running. In doing so, the system performance is examined.

- *Scenario 4: Performance of Distributive Control (one package)*  
A basic function of the control of material handling plants is target-based control which is realized by material handling control elements. In this scenario control entities, responsible for this task, as well as a number of other entities which are part of this negotiation (e.g. route finding) are activated. The control performance is determined by this distributor element when freight is passing.
- *Scenario 5: Performance of Joining Control (one package)*  
Realizing priority strategies for all junctions of material handling control elements is another basic task of material flow control. The control performance of a joining element under a simple workload is examined in this scenario.
- *Scenario 6: Behavior Under Workload*  
In this scenario the workload of the handling plant varies during test drive. Control performance as well as utilization rate of network and computer resources for single local controls are determined and analyzed. This scenario includes the scenarios 1, 4 and 5.

## 4. Monitoring Software Monitor Royale

Users and administrators of computer systems can choose from a large number of different monitoring systems. Among them are simple programs such as Linmark<sup>7</sup> and Panalyser<sup>8</sup> as well as more complex ones like OProfiler<sup>9</sup> or Perfsuite<sup>10</sup> of the U.S. National Center for Supercomputing Applications. Despite this variety, there currently is no tool which can fulfill the requirements of this research work. Among others, there especially are the following requirements to be mentioned: availability of data via network, abstract data interfaces for recording individual data as well as functional extensibility. With regard to these requirements the monitoring system Monitor Royale<sup>11</sup> was developed and published as open-source-software. Monitor Royale was especially developed for performance examination in decentralized control systems and tested with the system Package Royale. Thanks to its generic architecture, Monitor Royale can be used for the



<sup>7</sup> <http://linmark.sourceforge.net>

<sup>8</sup> <http://panalyser.sourceforge.net>

<sup>9</sup> <http://oprofile.sourceforge.net>

<sup>10</sup> <http://perfsuite.ncsa.uiuc.edu>

<sup>11</sup> <http://monitorroyale.sourceforge.net>

**Figure 4: Visualization of system load information via Evaluation Station**

performance examination of other different soft- and hardware systems. The tool is developed in C++ with Linux-/Unix-systems as a target platform. Monitor Royale consists of two essential components: monitoring agent (data collection) and evaluation station (visualization and analysis). A monitoring system can consist of several components, according to requirements. In general a m-zu-n-relation can be realized. The program parts are loosely linked among each other. Communication is done per RPC (remote procedure call) via a TCP/IP-connection. Besides data recall via network the use of data files is possible.

The *Monitoring Agent* is a program component which is used in monitoring systems. Besides utilization rate information about processors, random access memories and networks, information about active processes are also recorded. The utilization rate information of the computer system is taken from the proc-data system<sup>12</sup> of the operating system. The performance data of the control are selected from the interface file. This file contains key-value-pairs for any kind of purpose, here cycle time values of control agents. The use of text files increases the software's flexibility. Each program with the ability to operate file systems is also able to send data to the monitoring system to be recorded.

*Evaluation Station* is a graphic user interface of Monitor Royale. This component is used to control the monitoring and to manage and visualize collected data. The visualization tool enables an online-pursuit of data collection. Figure 4 shows the user interface of the program. A number of integrated functions enable various forms of presentation like, for example, scaling time and/or value axis or selecting certain measurement data sectors. A simultaneous presentation of several measurement parameters in one diagram allows a comparing analysis. Beyond that, the different statistic information is calculated and presented as numeric values but also as diagrams. And the user-friendliness gains from the ergonomic surface design and the intuitive pictograms regarding program control.

## 5. Results

The examined distributed control system applies non-real-time capable technologies such as Ethernet and PC-based hardware in domains of the facility and material flow system with time demands. Determinated performance figures shall act as contribution to a better understanding of problems, that may arise through undesired delays while control system operations. The control system Package Royale was tested in all scenarios described, in certain scenarios the test drives took up to 2 hours. However, only those results will be presented and discussed in the following that have been generated during the scenarios 1 and 6. The reason for the choice of these two scenarios is the equivalent to normal operation of scenario 6, being compared to idle operation (scenario 1). Scenario 6 includes scenario 4 and 5 and represents a more general use case. Scenario 2 and 3 lead to a change of the transportation network topology and can be considered as rarely arising special cases. Therefore they are negligible for the identification of control system performance bottlenecks in normal operation.

Note that for ease of examination all control agents were running on the preliminary mentioned agent computer and thus, all relative values are related to its performance values.

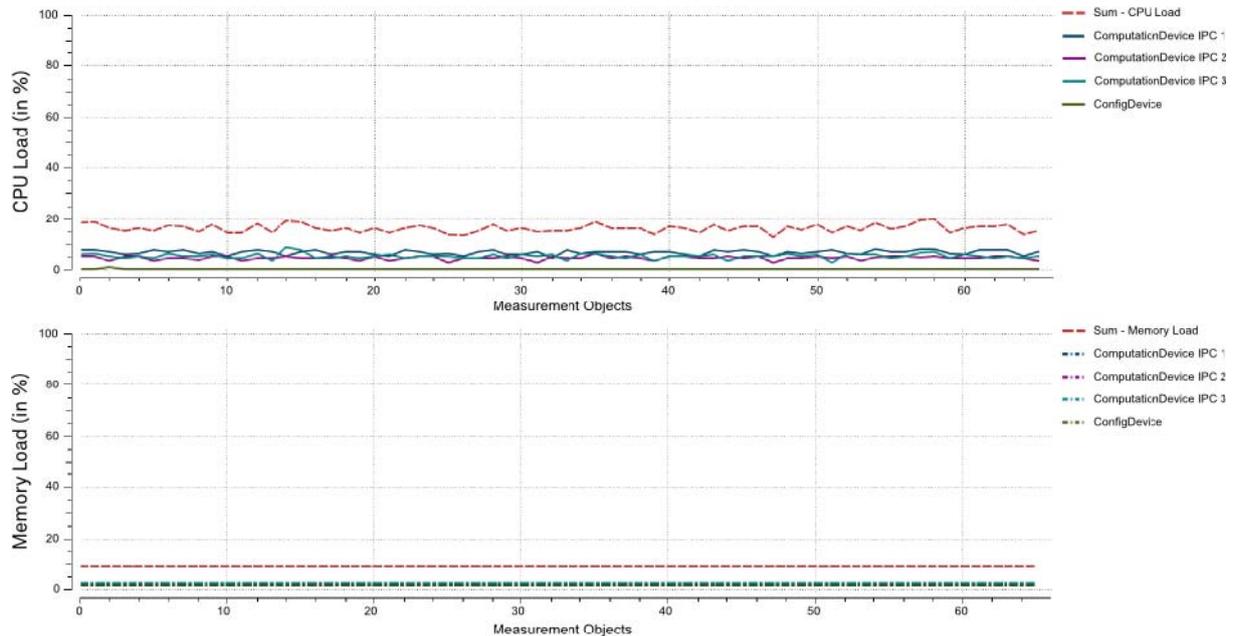
### 5.1. Control Utilization Rate in Idle Mode (Scenario 1)

In idle mode no objects are transported by the materials handling system. The activity of the control agents is reduced to running the control cycle including sampling process states. The adequately low utilization rate of the agent computer is shown in figure 5. Here, the series of measurement for the CPU-load (on top) and data memory utilization (at the bottom) are exemplary presented for a period of 60 seconds. The series of measurements at the bottom of both diagrams (approx. 6% for CPU-load and 2.6% for memory utilization) equate the three agent runtime environments (ComputationDevice) as well as a configuration program (ConfigDevice), which are simultaneously running on the agent computer. The resulting utilization rate of the agent computer amounts to 15 up to 20% for the CPU and 8.89% (91.22 MB) for the memory. The existing CPU-load is to be put down to the control cycles in the control agents. The measured cycle times of the agents are altogether below the defined recording threshold of 1 ms. The memory consumption of 91.22 MB is to be put down mainly to the virtual machines of Java (JVM). Besides regular management notifications like "still alive" messages or propagation of sensor events, there is too few network communication and therefore not presented here.

<sup>12</sup> In case of Unix-operating systems proc-data system means a directory structure, which shows the internal states of an operating system in form of files.

## 5.2. Behavior Under Workload (Scenario 6)

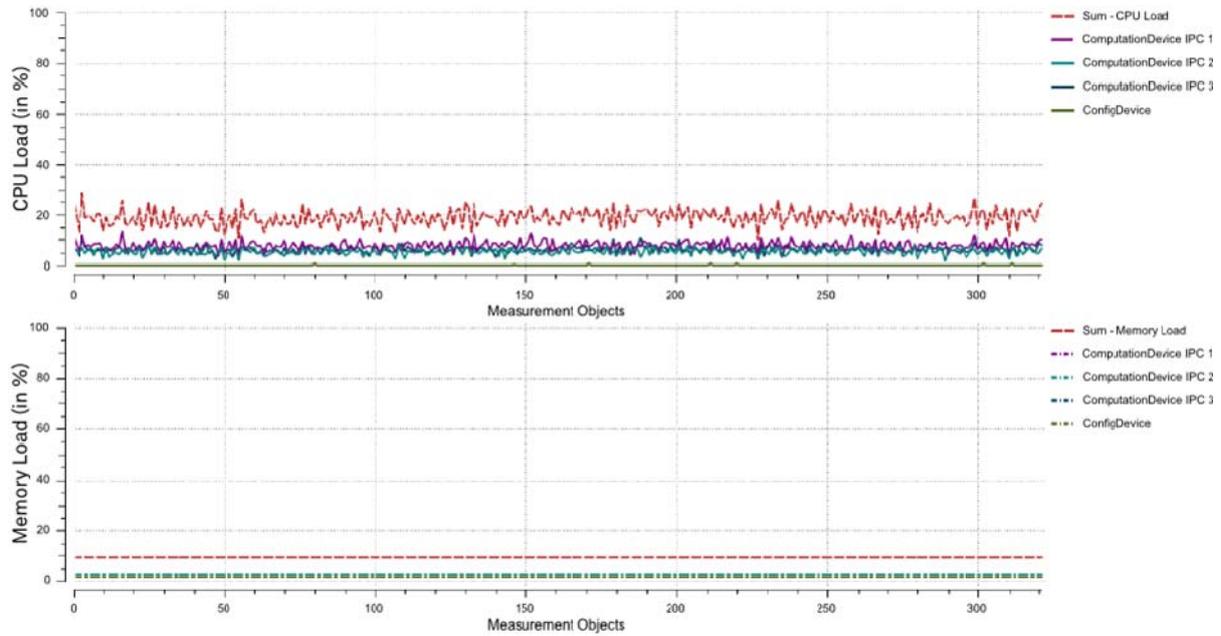
In scenario 6 the control system is examined under varying workloads. For this experiment only the lower cycle of the material handling plant is put into operation. Due to the existing transport system topology, the control has to realize a route decision function as well as a priority strategy. The number of objects rotating in the system has



**Figure 5: CPU- and memory utilization rate of agent computer for scenario 1 (no transported packages)**

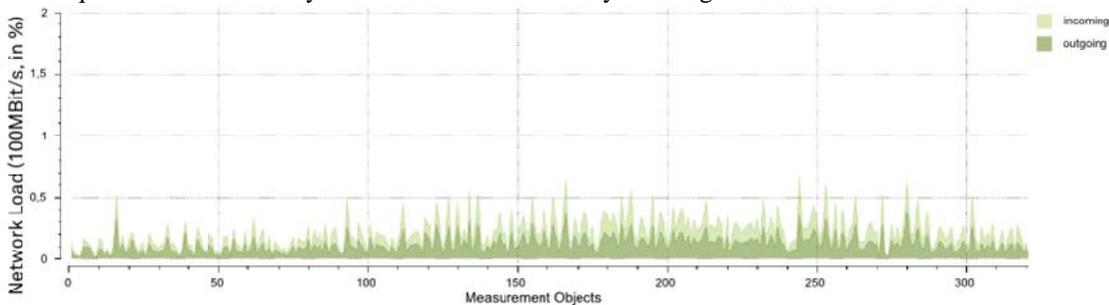
gradually been increased from 0 to 12. The measurement data presented in figure 6 show the status under maximal workload of 12 transportation goods over a period of approx. 300 seconds. In the experimentation the 12 goods were circling over the conveying system with a velocity of about 1 meter per second. With distances of about 5 meter between the goods this scenario simulates a packet flow of 720 packets per hour.

When comparing the utilization rate of the agent computer with the ones from scenario 1, only a small increase can be observed. More precise examinations showed that the increased process load can primarily be put down to the processing of Web Services. Therefore, the scalability strongly depends on the used Web Service processing library and the number of Web Service calls and. This, on the other hand, depends on the number of sensors whose triggering results in event propagation via Web Service. The memory utilization, however, is not different from that in scenario 1. This is explained by the fact that the initialization of all entities (control services, material handling agents) has already been completed. Increased memory utilization due to the increased number of control agents is not recognizable for 12 packages.



**Figure 6: CPU- and memory utilization rate for scenario 6 (12 transported packages)**

The network utilization rate of the agent computer for scenario 6 is small. The arithmetic average is 0.42% (52.5 KByte/s) for incoming and outgoing traffic altogether (figure 7). When comparing these values with the performance evaluation of Fast-Ethernet (see [RWN+95]), collisions during media access can be neglected. In this case the non-deterministic CSMA/CD-access procedure has no negative effect on the application with real-time requirements. Scalability can be further increased by subnet generation.



**Figure 7: Network utilization rate of the agent computer for scenario 6 (12 transported packages)**

The workload is reflected in the increased local reaction times of control agents. Figure 8 shows the recorded reaction times of an agent responsible for a distributive material handling element. The majority of cycle times can be found in the area between 150-160 ms. Striking values at the beginning of the recorded spectrum as well as values below recording threshold are representing the cycle time in idle mode. Discordant values greater than 520 ms can be ascribed to the work of Java Garbage Collector<sup>13</sup>. When ignoring the discordant values, the average local reaction time amounts to 167 ms. The negative effect of the slow reaction time and especially the discordant value was alleviated by proactive behavior of control agents. This means that control decisions are prepared in advance with the goal to reduce calculations at time-critical moments. While for example goods are transported to a decision point (joining or branching of material flows) the responsible unit load agents use this transportation time to calculate the next directions or the right of way at these points.

<sup>13</sup> The Java Garbage Collector is an integrated routine of the Java Runtime Environment with the task to optimize memory resource usage while executing Java applications. The execution of this routine makes demands on computer resources and affects therefore the execution speed of running applications. The execution time as well as its duration depend on the current runtime situation and used algorithms of the Garbage Collector; they are therefore not deterministically predictable. It is considered for applications with realtime demands to preemptively initiate the Garbage Collector manually in (for realtime demands) uncritical moments.

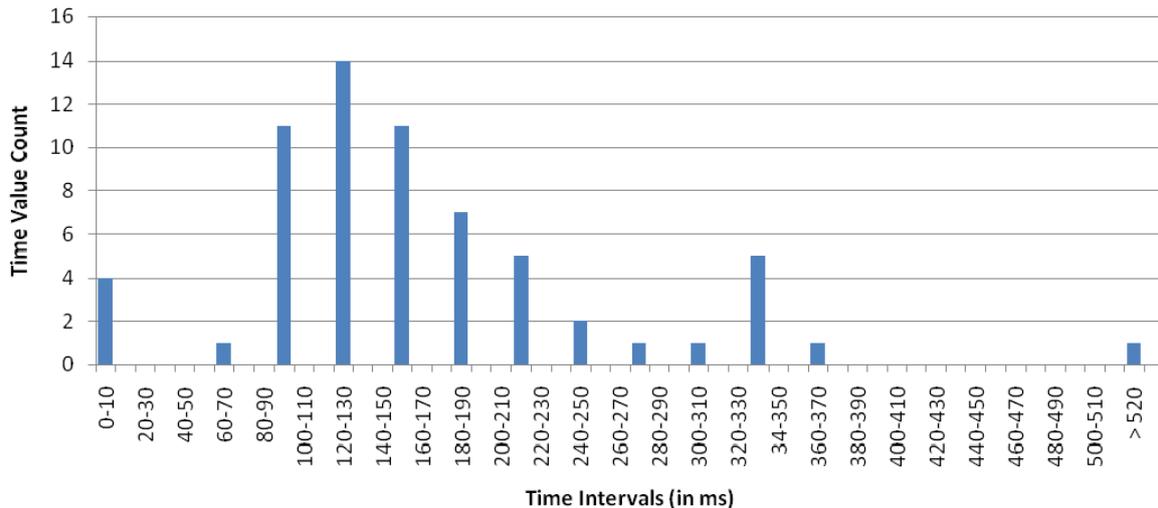


Figure 8: Cycle time distribution of control agents for material handling element U1

### 5.3. Result Summary

The here described approach had not only the measurement of performance figures as the goal. The approach, measuring tool and gathered performance values support the analysis of control solutions regarding problems and performance bottlenecks, which can be easily identified by means of graphically illustrated measurement results. The experiments could not confirm the assumption that bottlenecks are caused by non-deterministic communication technologies. Outlying cycle times of control agents could be quantified and lead back to the work of the Java Garbage Collector. This behavior endangers the reliability of the control system and needs to be considered in the architectural system design. The main part of the cycle times (average 167 ms) is caused by the dispatching of Web Service messages (see figure 2). An optimization of used programming libraries or adoption of a more efficient communication protocol will improve the overall reaction time of the control system.

## 6. Conclusion and Outlook

This work presents a tool-based approach for evaluating and comparing performances of decentralized control systems. This approach includes a proposal of performance metrics, the definition of variable parameters for the examination as well as the preparation and execution of test scenarios. The monitoring system Monitor Royale has been used for distributed data acquisition, which is available as open-source-software. Due to the configurable data interface it can be deployed for acquisition and visualization of measurement values in different scenarios and applications. This proposed approach has been exemplary applied to the decentralized material flow control Package Royale. This control was developed at the Chair for Handling and Warehouses of TU Dortmund in context of the research area Internet of Things and is run in a decentralized hardware environment. The experiments made statements about quality and performance of the examined control possible. Moreover, the reasons for the observed control behavior under workload were understood.

The introduced approach considers general core functionalities of material control systems being developed following the Internet of Things in intralogistics paradigm. The measurement figures of the examined control system Package Royale, that have been gathered with the presented approach, are suitable for a performance evaluation and comparison under different usage conditions. The experiments have revealed time bottlenecks in the behavior of the control system and these results are the basis for a following optimization. The usage of this approach with a specific control system has shown its general applicability. The goal of an approach for cross-architectural performance evaluation and comparison of material flow control systems has been therefore achieved.

This paper has its origins in the sub-project “Real-Time Logistics” of the collaborative research center 696. A further step of development is the integration of the proposed examination methods into the real-time-logistics model with the aim to describe and analyze the time behavior of decentralized control systems in an intralogistic plant. The time and performance data gathered with the system Monitor Royale serve as input values for the RTL-model. To be able to realize this plan, further experiments for gathering measurement data in real control systems are necessary.

## References

- [BSc01] Bussmann, S.; Schild, K.: An Agent-based Approach to the Control of Flexible Production Systems. In: Proceedings of the 8th IEEE Int. Conf. on Emergent Technologies and Factory Automation (ETFA 2001), 2001. S. 481-488.
- [Fer78] Ferrari, D.: Computer systems and performance evaluation. Prentice Hall, Englewood Cliffs, 1978.
- [FLH+09] Feldhorst, S.; Libert, S.; ten Hompel, M.; Krumm, H.: Integration of a Legacy Automation System into a SOA for Devices. In Proc. of 14th IEEE International Conference on Emerging Technologies and Factory Automation, IEEE, IES 2009.
- [GCK08] Günthner, W. A.; Chisu, R.; Kuzmany, F.: Internet der Dinge - Zukunftstechnologie mit Kostenvorteil. In: Fördern und Heben, Ausgabe 10, 2008, S. 556-558.
- [GtH10] Günthner, W. A. (Hrsg.); ten Hompel, M. (Hrsg.): Internet der Dinge in der Intralogistik. Springer, 2010.
- [Gud05] Gudehus, Timm: Logistik: Grundlagen - Strategien - Anwendungen. 3. Aufl., Springer, 2005.
- [HLL07] ten Hompel, M.; Libert, S.; Liekenbrock, D.: Analyse der Echtzeitproblematik bei der Steuerung von Stetigfördersystemen. In: Crostack, H.-A.; ten Hompel, M. (Hrsg. Bd.): 1. Kolloquium des Sonderforschungsbereich 696, Dortmund : Verlag Praxiswissen, 2007, S. 235-255.
- [HLS06] ten Hompel, M.; Libert, S.; Sondhof, U.: Dezentrale Steuerung für Materialflusssysteme. In: Logistics Journal, 2006, S. 1-9.
- [Jai91] Jain, R.: The Art of Computer Systems Performance Analysis. John Wiley & Sons, Inc., 1991.
- [Kle96] Klehmet, U.: Mathematisch-heuristische Leistungsbewertung des Medienzugangsverfahrens beim Profibus. In: Automatisierungstechnik, Ausgabe 3, 1996, S. 108-119.
- [LHN07] Libert, S.; ten Hompel, M.; Nettsträter, A.: Modell zur Beschreibung von Zeitfaktoren in der automatisierten Stetigfördertechnik. In: Logistics on Demand, 2007, S. 167-188.
- [LNH08] Libert, S.; Nettsträter, A.; ten Hompel, M.: Das RTL-Modell für dezentrale Materialflusststeuerung. In: Crostack, H.-A.; ten Hompel, M. (Hrsg. Bd.): Berichte aus dem SFB 696, Dortmund : Praxiswissen, 2008, S. 137-161.
- [LMT99] Lian, F.-L.; Moyne, J.; Tilbury, D.: Performance Evaluation of Control Networks: Ethernet, ControlNet, and DeviceNet. In: IEEE Control Systems Magazine, Band 21, Ausgabe 1, 1999, S. 66-83.
- [Lor03] Lorentz, K.: Ein Beitrag zur verteilten und adaptiven Materialflusststeuerung für Stückguttransport. Dissertation, Magdeburg Universität, Fakultät für Maschinenbau, 2003.
- [Oss73] Osswald, B.: Leistungsvermögensanalyse von Datenverarbeitungsanlagen. Toeche-Mittler Verlag, Darmstadt, 1973.
- [Rad00] Radtke, A.: Beitrag zur Entwicklung optimierter Betriebsstrategien für Sortiersysteme. Dissertation, Universität Dortmund, Fakultät Maschinenbau, 2000.
- [RFo08] Follert, G.; Roidl, M.: Evaluation of Routing Strategies for Decentralized Self-Organisation in Large Scale Conveyor Systems. In: Kimberly E. et al. (Hrsg. Bd.): Progress in Material Handling Research: 2008, Charlotte (NC, USA) : The Material Handling Institute of America, 2008.
- [Ros02] Rostan, M.: Zykluszeit ist nicht alles: zur Performance von Feldbussystemen in der Fertigung. : SPS/IPC/DRIVES Nürnberg, 13. Fachmesse und Kongress, Heidelberg, 2002, S. 99-108.
- [RWN+95] Rindos, A.; Woollet, S.; Nicholson, L.; Vouk, M.; Nog, S.; Kotz, D.: A Performance Comparison of TCP/IP and MPI on FDDI, Fast Ethernet, and Ethernet. Technical Report, Dartmouth College, Hanover, 1995.
- [Ste03] Steusloff, H.: Verteilte Echtzeitsysteme und eingebettete Systeme - Über Systeme und Technologien. : Verteilte Echtzeitsysteme, Fachtagung der GI-Fachgruppe 4.4.2 „Echtzeitprogrammierung und PEARL (EP)“, Berlin : Springer, 2003.
- [Svo76] Svobodova, L.: Computer Performance Measurement and Evaluation Methods: Analysis and Application. Elsevier, New York, 1976.
- [TVa99] Tovar, E.; Vasques, F.: Analysis of the Worst-Case Real Token Rotation Time in PROFIBUS Networks. In: Proceedings of the Fieldbus Conference (FeT '99), 1999.
- [Wil06] Wilke, M.: Wandelbare automatisierte Materialflusssysteme für dynamische Produktionsstrukturen. Dissertation, TU München, Fakultät für Maschinenwesen, 2006.
- [Win08] Windt, K.: Ermittlung des angemessenen Selbststeuerungsgrades in der Logistik - Grenzen der Selbststeuerung. In: Nyhuis, P. (Hrsg. Bd.): Beiträge zu einer Theorie der Logistik, Berlin [u.a.] : Springer, 2008, S. 349-372.