Efficient Data Aggregation and Management in Integrated Network Control Environments

Patrick-Benjamin Bök, Michael Patalas, Dennis Pielken, and York Tüchelmann

Ruhr-University Bochum, 44801 Bochum NRW, Germany
`(boek,patalas,pielken,tuechelmann)@iis.rub.de`
`http://www.iis.rub.de`

Abstract. Due to the emerging growth of computer networks, broadly based measurements, monitoring and management become necessary, for example, to solve occurring problems. Lots of different concepts exist for each of the mentioned functionality. Therefore, distributed network control architectures integrating all of these functionalities are in the focus of current research. To take advantage of this architectures, advanced data aggregation and management schemes are required because an efficient access to the distributed data is critical in this case. In this paper, we present a data aggregation and management scheme that improves the performance of data handling in Integrated Network Control environments. The Integrated Network Control concept is enhanced by an multidimensional on-line analytical processing (OLAP) scheme. A performance analysis shows that the proposed scheme noticeably improves the overall performance of the Integrated Network Control environment.

Keywords: Business Intelligence, Data Aggregation, Hierarchical Communication, Network Control, Network Measurement, OLAP.

1 Introduction

Computer networks have to be monitored and managed efficiently with a focus on the performance of the network and its elements observed. Due to the increasing size and complexity of computer networks, broadly based measurements, monitoring and management become necessary especially if problems regarding the performance occur within the network. Lots of different concepts and tools exist for each of these functions. Because of the huge amount of different tools and concepts, Integrated Network Control architectures have been developed that try to subsume these tools and concepts into a controlled architecture. Instead of using much additional and expensive high-performance hardware, Integrated Network Control environments as [1] have the advantage to perform measurement, monitoring and management functions using one central system that distributes the functions of network control towards the observed elements, e.g. the traffic generating hosts. Thereby, these elements are utilized for network control purposes. To work efficiently, [1] uses a hierarchical group communication structure. An efficient data aggregation and management scheme is required to take

R. Szabó et al. (Eds.): AccessNets 2010, LNICST 63, pp. 269–282, 2011.

advantage of this communication structure because synchronization and analysis are of fundamental importance to the fully distributed approach of Integrated Network Control. Due to the fact that the measured or monitored data is distributed over the elements, accessing this data is critical because the network of observation should not be influenced significantly by the transfer of data caused by network control. However, an administrator must be able to access the data when it is required. This ranges from historical analysis to near-term monitoring. Therefore, the data has to be transferred to the administrator. Besides, the distributed data has to be merged and aggregated, respectively.

In the field of business intelligence, several approaches of data warehousing with on-line analytical processing (OLAP) exist. These are already in use for data analysis and monitoring purposes, also in distributed systems, but disregarding any performance aspects with reference to the used computer network. A brief review on related work is given in Section 2. On the basis of the related work, we propose a data aggregation and management scheme that improves the performance of data-handling in Integrated Network Control environments. Therefore, a central data warehouse including an OLAP-engine is integrated into the data-handling component of the Integrated Network Control Framework (INControl-F) [1]. The enhanced data management allows an efficient analysis of live and historical data, using composed multidimensional data models. Data aggregates are pre-calculated in a number of operational data store. The access by and the transfer to a data warehouse are managed by a specially designed data management component. Thereby, the performance of the hosts and the computer network is not influenced negatively.

The remainder of this paper is structured as follows. The Integrated Network Control architecture is outlined in Section 3. In Section 4, the proposed data aggregation and management scheme is presented. The general improvements induced by the proposed scheme are discussed in Section 5 and confirmed by the results of a performance analysis presented in Section 6. We conclude our contribution in Section 7.

2 Related Work

Data warehousing and accordingly different types of OLAP are used in several business segments, e.g. e-commerce or clinical reporting, as presented by Thomas et al. [2], Xiangdong and Xiao [3] as well as Hamm et al. [4]. Because of them being used for business purposes and not for performance sensitive network control purposes, the influence caused by computation on nodes and by data exchange using a network has been neglected. Therefore, these approaches cannot be used or adopted for network control purposes. To improve the efficiency of computation and analysis of data, Albrecht et al. [5] propose an advanced management of multidimensional data cubes. Therefore, a context-sensitive model that uses hierarchical data stores is introduced. Specialized methods and systems for distributed and parallel OLAP and data warehousing are presented by several researchers. Chen et al. [6] focus on the organization of distributed data sources in OLAP infrastructures. They present an approach that allows to harmonize the data of different distributed sources according to dynamic patterns and rules. Their prototype confirms the motivated improvements for data organization in distributed OLAP environments. Also Jianzhong and Hong [7] show improvements for the handling of the huge amount of data in parallel and distributed data warehouses. The authors present a scheme that improves the range sum query processing in such OLAP environments. Akinde et al. [8] increase the efficiency of OLAP query processing, that is very critical in distributed OLAP environments. They show that the storage of data without using an operational data store is impractical. Using an OLAP query translator, queries on the data warehouse can be spread to a number of distributed OLAP elements.

Besides organizing and accessing the data of data warehouses, the storage of data within the data bases of date warehouses is critical. He et al. [9] focus on the improvement of a storage algorithm for multidimensional OLAP (MOLAP) that uses multidimensional data cubes. The proposed algorithm reduces the time to maintain multidimensional data cubes because in the majority of cases this task is time-critical. Therefore, the performance of data cubes is improved on the basis of the hierarchies existing in multidimensional data cubes. A similar concept is presented by Shimada et al. [10], who simplify the dimension dependency in the storage process. The dimension dependency is a capital problem when data arrays are accessed in multidimensional data cubes. All of these researchers show different improvements for the aggregation and storage of data. However, they neglect the performance of systems and networks used for computation or transfer, respectively. Yeung et al. [11] discuss the problem of transferring data from distributed and operational data stores to a central data warehouse. Their concept allows updating the data warehouse, initiated by operational data stores if a modification of data occurs. It allows a centralized near-term analysis, but impairs the performance significantly with an increasing number of operational data stores to include. Xu et al. [12] introduce an on-demand pull mechanism to solve this shortcoming. Instead of using an autonomous push mechanism, data is pulled from the operational data stores by a central data warehouse. This concept works more efficient when using an increasing number of operational data stores. Besides, Hose et al. [13] improve the performance of OLAP by means of an *a priori* preparation of complex aggregation requests.

All of the mentioned approaches in this section present data warehousing concepts with on-line analytical processing (OLAP) that are already in use for data analysis and monitoring purposes in distributed systems. However, the impact of their concepts on the performance of a computer network or the system as a whole is ignored, which makes it impossible to use these approaches in Integrated Network Control environments.

3 Integrated Network Control Architecture

In most of the network control architectures, much additional hardware is required to perform network control. Network control includes functions and software for network monitoring, measurement and management. For these functionalities it is necessary to site active and passive elements in the network. In [1] an architecture is presented that utilizes up to every client of a network, instead of siting probing and steering elements across the network. To efficiently utilize every client for network measurement, monitoring and management purposes, an efficient communication structure is necessary because the emerging overhead, which is created by control and command flows as well as data migration flows, for example that consist of network captures, between a central component and the utilized elements (Fig. 1 (a)) may become critical. To avoid a negative influence on the network and its elements, a dynamic hierarchical group communication structure (see Fig. 1 (b)) is introduced.

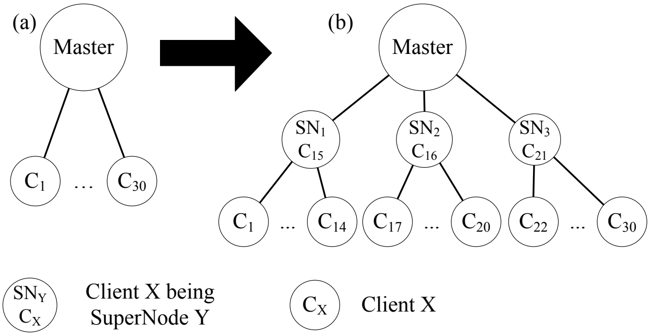

Fig. 1. Hierarchical Communication Structure

Instead of using a flat hierarchy, dynamic groups are formed as possible by selectable parameters (e.g. by the logical subnets). Thereby, a central controller (Master) that organizes all tasks is enabled to delegate tasks to group managers (Super-Nodes), which communicate with the clients of their group and forward the accordant tasks. Super-Nodes are chosen by the master on the basis of a score that, among other things, includes values for computing power, memory power, data transfer rate, uptime and past experience of availability. Also a fault detection mechanism allowing a fault-tolerant behavior of this structure is already included. Therefore, the availability of Super-Nodes is periodically checked by the Master and the structure is changed accordantly. The occurring problem of data consistency is guaranteed by the data aggregation and management scheme, presented in the next section. In the following, we use host synonymic with client and node as instrumented elements. The task forwarding is illustrated in Fig. 2. To improve the architecture's and its communication structure's efficiency, the Super-Nodes perform data aggregation methods on the data retrieved from the clients of their group (see Fig. 2), before it is forwarded to the Master. The proposed data aggregation and management scheme for Integrated Network Control purposes, which uses this communication structure will be discussed in the next section.

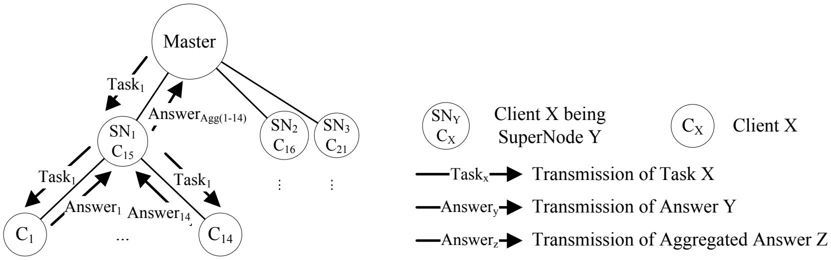

Fig. 2. Hierarchical Task Forwarding and Answer Aggregation

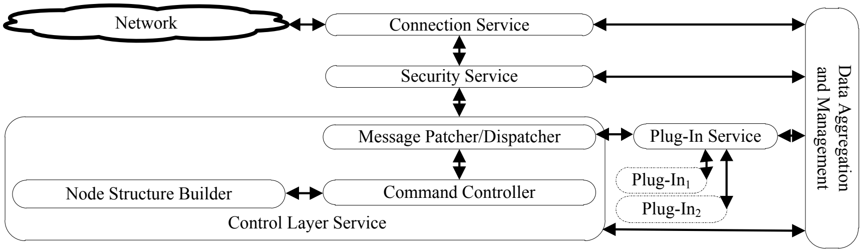

Fig. 3. Enhanced INControl-F Architecture

The enhanced Integrated Network Control architecture has been implemented within a framework called INControl-F, illustrated in Fig. 3. It consists of several components that provide the basic functionality required to build the communication structure and to enable the described features. The INControl-Framework runs on hosts of inspection within a network . Thereby, no additional hardware except for a managing Master is required. The *Connection Service* is a lightweight communication component that is responsible for any communication between Master, Super-Nodes and hosts. The *Security Service* ensures secure and confidential data transmission between the different levels in the communication hierarchy. It works transparent to any component. A *Control Layer Service* includes sub-components for patching and dispatching messages to or from components, the *Message Patcher/Dispatcher*, for controlling the framework, the *Command Controller*, and for the automatic build up of the hierarchical group communication structure, the *Node Structure Builder*. An *INControl-Plug-In* represents a measurement, monitoring or management function that has been included as an attachable component into the framework. To allow an easy integration of plug-ins, a *Plug-In Service* offers a stable API and essential functions to be used for communication by and with plug-ins. As illustrated in Fig. 3, a *Data Aggregation and Management* component is included in the architecture. It is responsible for the aggregation and management of all collected data. The novel data aggregation and management scheme used within this component is presented in the next section.

4 Data Aggregation and Management Scheme

The proposed data aggregation and management scheme for Integrated Network Control environments considers the hierarchical communication structure, its various layers and their specific demands. The lowest layer in the hierarchy contains the hosts of a network on which data is collected and initially aggregated. Due to the definition of the Integrated Network Control architecture, computationally intensive data analysis is not performed on the lowest layer. Thus, measured data will be transferred to the corresponding Super-Node, representing the next higher layer in the hierarchy and having sufficient capabilities to serve a quantity of hosts. Because a Super-Node is part of the network, it may also collect data. In our scheme we activate an operational data store on each node that acts as a Super-Node. An operational data store contains the collected data received from the lower layer and the Super-Node itself. Data aggregation and management functions have to be the same on each host because every host may become a Super-Node. Because of performance restrictions on each host or Super-Node, a Super-Node will not contain its own OLAP-Engine for multi-dimensional data analysis in our scheme. Instead, a central data warehouse that includes an OLAP-Engine to perform the multi-dimensional data analysis is attached to the Master. The data warehouse automatically builds up the multi-dimensional data cubes based on the available information of each INControl-Plug-In that generated the corresponding data. To minimize the network overhead created by the Integrated Network Control architecture, collected data from the operational data store can be transferred to the data warehouse in large intervals, for example, when the network utilization is low. Thereby, historical analysis becomes possible.

The processing of a request is explained in detail in Fig. 4. First, a request is sent to the data warehouse. If a request relates to data that has not been transferred from the according operational data store to the data warehouse, yet, an aggregate of the requested data compressed with the aggregation factor k_{level} will be transferred to the data warehouse. The value of the aggregation factor k_{level} depends on the current utilization of the network and the corresponding Super-Node. These are checked periodically using statistical measurements from routers in the core network and from Super-Nodes. Thereby, it can be ensured that the network and the Super-Node will not be influenced negatively. Finally, the OLAP-Engine on the data warehouse is used as a MDX/SQL-converter and SQL-queries are sent to the corresponding Super-Node to request the data. Since the data structures in each operational data store are identical to the data structures in the data warehouse, an execution of short SQL statements becomes possible. Afterwards, the result is returned and added to the data warehouse, so that it is available for future requests. Using this concept, near-term analysis can be performed efficiently.

Nodes store their measured values in flat files, whereas the Super-Nodes store the data retrieved from nodes in an embedded database and the Master stores the data retrieved from the Super-Nodes in a framework external database. Because nodes and also Super-Nodes should not be overloaded with data to store

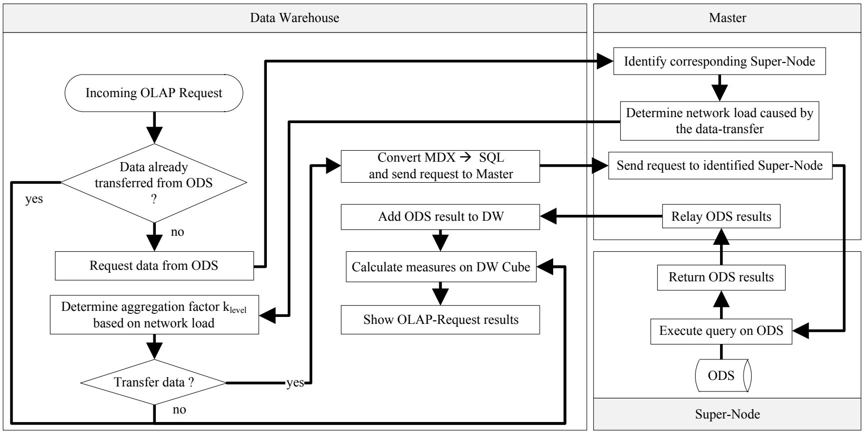

Fig. 4. OLAP Request Processing

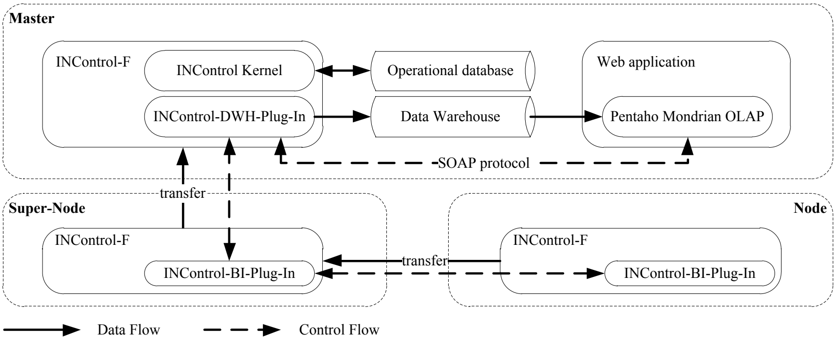

Fig. 5. Scheme Components within INControl-F

within their structures, data has to be deleted after some time. The failure of a Super-Node leads to a change of the Super-Node for a certain group of nodes. This may become a critical point if data has already been transferred to the Super-Node, but not to the Master. If the data would be erased on the nodes directly after they are transferred to the next higher level in the communication hierarchy, such a failure of a Super-Node or link failures would lead to the loss of data. To counter this problem, data transferred to the next higher level will only be erased if the completion of a transfer to the Master has been acknowledged. It can be assumed that the Master as high-performance element offers redundancy in data storage. Thereby, the loss of data can be eliminated. The data aggregation and management scheme extends the INControl-Framework with several sub-components, as illustrated in Fig. 5. The active instance of INControl-F on the Master is extended with a data warehouse plug-in (INControl-DWH-Plug-In) that connects INControl-F to a data warehouse. An operational database of the data warehouse is connected to INControl-F through the INControl-Kernel.

The active instance of INControl-F on a Super-Node or node is extended with an business intelligence plug-in (INControl-BI-Plug-In) that connects such an instance to the data warehouse plug-in of the Master or to the business intelligence plug-in of a Super-Node, respectively. Both plug-ins together represent the functions of the data aggregation and management scheme, which utilize the operational database and the data warehouse. The analysis of the data in the data warehouse is allowed through a web application on the basis of the public domain OLAP tool Pentaho Mondiran OLAP.

5 General Improvements

The data aggregation and management scheme described before leads to several improvements discussed in this section. Because of network control data being additional overhead for the network, it should not influence operative business applications using the network. Using the aggregation factor k_{level}, a smart varying of the granularity of data of interest to be transferred becomes possible. Thereby, the amount of data from the INControl-Framework that has to be transferred over the network to the central data warehouse can be reduced as confirmed in the next section. The aggregation of data within the given hierarchical communication structure is performed bottom-up, starting at the hosts using an aggregation factor of k_1. This factor defines how strong the data from a host will be aggregated before it is transferred to the Super-Node. As soon as the aggregated data of all nodes is available in the operational data store of a Super-Node, this already aggregated data will be auxiliary reduced using the aggregation factor k_2. Thereby, the total amount of data transferred from the operational data store of a Super-Node to data warehouse of the Master is reduced. Aggregated data will be available on the Super-Node until it is requested and transferred to the data warehouse or the recurring transfer to the data warehouse is performed. Thereby, the total amount of transferred data within the hierarchical communication structure can be significantly reduced as shown in the following. In Fig. 6 the ideal impact of the aggregation factor on each level in the communication hierarchy is illustrated. The total amount of data to be transferred from $i = \{0, ..., n\}$ clients to the operational data store of a corresponding Super-Node $j = \{0, ..., s\}$, with each client collecting an amount of $d_{i,j}(\Delta T)$ data within period ΔT, is

$$\sum_{i=0}^{n} \frac{d_{i,j}(\Delta T)}{k_1}$$

Hence, the total amount of data to be transferred from s Super-Nodes j to the central DW, considering the amount of data $d_j(\Delta T)$ measured within period ΔT on the Super-Node itself, is

$$\frac{\sum_{j=0}^{s}\sum_{i=0}^{n}\frac{d_{i,j}(\Delta T)}{k_1}}{k_2} + \sum_{j=0}^{s}\frac{d_j(\Delta T)}{k_1 \cdot k_2} < \sum_{j=0}^{s}\sum_{i=0}^{n} d_{i,j}(\Delta T) + \sum_{j=0}^{s} d_j(\Delta T)$$

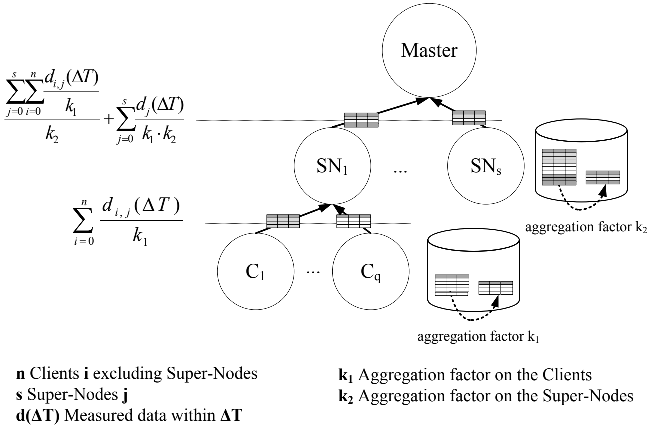

Fig. 6. Impact of the aggregation factor

which is smaller than without aggregation. Because of $k_1 > 1$ and $k_2 > 1$, the aggregation has a significant impact on the amount of values and data transferred. To demonstrate this, consider four Super-Nodes each having three hosts beneath. Thus, a group of hosts with the corresponding Super-Node consists of four measuring hosts. Each node or Super-Node creates a measured value once per second. The total amount of network control overhead created by the transferred data can be reduced according to the aggregation factors k_{level}. For example, nearly 75% of the amount of data to be transferred can be saved using the proposed scheme within the scenario described before if the aggregation factors k_1 and k_2 are chosen to be 2. Using these aggregation factors, a flexible reaction on the different demands becomes possible.

For real-time data warehouse architectures, an essential requirement is to send data to a central data warehouse as fast as possible. Thereby, the data warehouse has to manage a large number of connections. On the one hand, this loads the server. On the other hand, real-time traffic is created, even when the network is almost near to its limit. If monitoring requests are rarely required, these are disadvantages that cannot be tolerated because network control is an additional application to observe the vitality of a network. The proposed scheme overcomes these disadvantages. The Super-Nodes reduce the operational burden of the data warehouse. The data warehouse does not have to address all the hosts individually. Instead, only the aggregated data from the Super-Nodes have to be managed. Additionally, initial data aggregation of collected data can be performed on the nodes. The dynamical variation of the aggregation factors k_{level} based on the current utilization of Super-Nodes, hosts and network, holds the requirement that none of these elements should be affected by the Integrated Network Control architecture at any time.

6 Results

As shown in this section, the results of the performance analysis and are compliant to the general improvements motivated in the former section. For the performance analysis, a basic scenario is constructed that consists of five nodes (Dell OptiPlex 760, E7000-Series Core2Duo, 2GB DDR2 RAM, Windows Vista SP2) including two Super-Nodes that are both connected to the Master and, thereby, building an unbalanced tree. The INControl-Framework on every node controls a monitoring tool that creates a measured value once per second. For the performance analysis, it does not matter what the content of such a value is. After thousand measured values the test is stopped and the values are transferred to the Master, passing the corresponding Super-Nodes. The OLAP view within the data warehouse on the Master is configured to use three measures over five dimensions and eleven dimension levels overall to restructure the 5000 values received from the nodes. In the basic scenario, no aggregation of data is performed and k_1 and k_2 are set to 1. Beside aggregation, compression is applied before transferring the data from the nodes to their Super-Nodes and from them to the Master. In comparison to the basic scenario, the measured values are aggregated at the Super-Node using our scheme within three other scenarios, according to the configured measures mentioned before. Therefore, k_2 is set to 1 for each of the following scenarios. In the first aggregation scenario the measured values are aggregated with $k_1 = 2$. It is to note that this does not mean to build the average value, but create a value based on the defined measures for each aggregation interval. The two other aggregation scenarios aggregate the measured values with $k_1 = 5$ and $k_1 = 10$, respectively. The number of measured values transferred to the Master in each scenario is presented in Fig. 7. As mentioned before, the total number of values transferred to the Master if aggregation is not applied is 5000 measured values. After aggregation is applied with $k_1 = 2$ on the measured values, the number of transferred values is nearly the half than without aggregation. Because not in every moment of the aggregation interval enough measured values are available, the number of transferred values is not exactly the half, but only a few percent more. This is also applies to the other aggregation scenarios. Accordingly, the number of transferred values falls down nearly linear if the aggregation factor is set to 5 or 10, respectively. In contrast to the number of values transferred to the Master, which is the overhead normally created by network control, the amount of compressed data representing

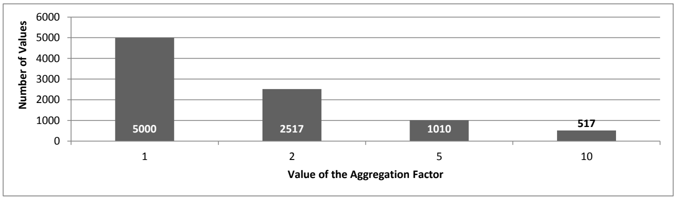

Fig. 7. Amount of measures transmitted depending on aggregation

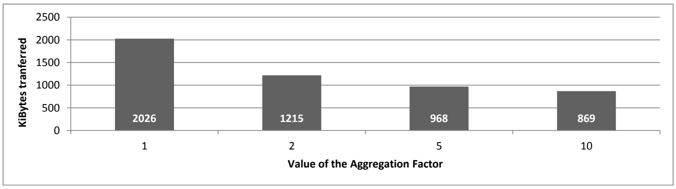

Fig. 8. Amount of compressed data transmitted depending on aggregation

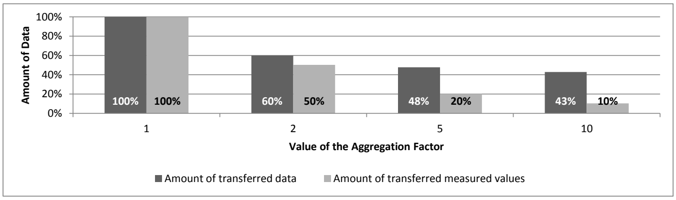

Fig. 9. Aggregation Improvements

these values that are transferred to the Master does not decrease linear to the aggregation factor. As shown in Fig. 8, the amount of data transferred for about the half of the values than the original number of values is nearly 60%. In the other scenarios, one can see that the amount of data is not linearly reduced like the number of values. In the case where the aggregation factor is set to 10 , the number of values are nearly 10% of the original ones, but the amount of data is still above 40% of the original one with no aggregation activated. This results were expected because compression functions work more efficient if the number of measured values is higher, so the efficiency of the compression algorithm must decrease with a decreasing number of measured values. The comparison of the number of transferred values and the amount of data transferred shows huge differences with an increasing aggregation factor and the constant number of measured values created on the nodes (see Fig. 9). The number of transferred values decreases linear to the aggregation factor k_{level}, whereas the amount of transferred data is nearly constant after setting the aggregation factor to 10. From this results we can derive that the efficiency of our scheme is not only driven by the aggregation factor, but also by the number of measured values that drive the efficiency of the compression algorithm. To verify this, additional large scale tests have been performed. After the frequency of creating measured values was increased, obviously more measured values were retrieved and the compression algorithm worked more efficient if k_1 was set to 5 and 10.

As confirmed, the performance of the network can be increased the proposed data aggregation and management scheme. But also the influence of the scheme on the performance of nodes, Super-Nodes and the Master, which are elements that run this additional components is of special interest. The average memory usage is nearly 10 MB for each node, 16 MB for each Super-Node and only 50 MB for the central Master. The memory usage on the nodes is not

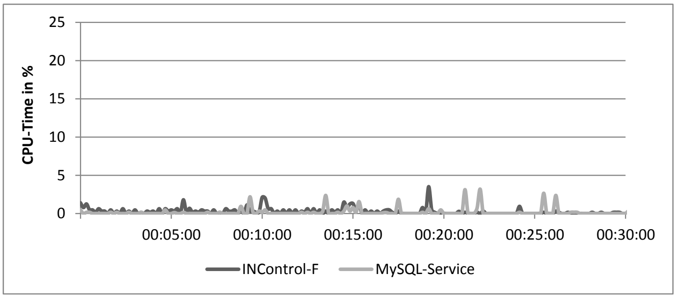

Fig. 10. Additional CPU Usage (Master)

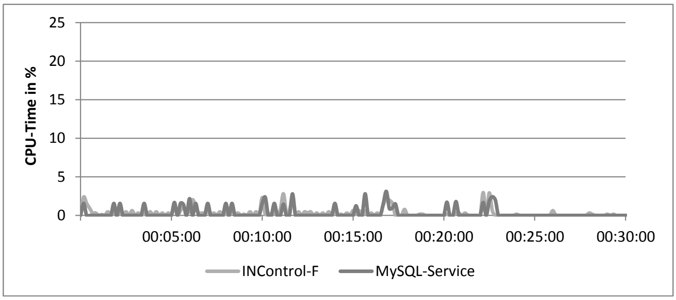

Fig. 11. Additional CPU Usage (Super-Node)

influenced by the aggregation of data because this is controlled by the Super-Nodes or the Master, respectively. So the memory usage on the nodes is negligible. Also the Super-Nodes that control and perform the aggregation of measured values use an acceptable amount of memory. Finally, the Master uses a higher amount of memory because the OLAP-Server that uses Apache Tomcat and the MySQL database as additional components are running on the Master. This is also acceptable because the Master as central component is a high-performance element. The Integrated Network Control architecture was designed with the constraint that utilized elements, excepting the Master, must not be affected by their additional features. The analysis of the memory usage shows that the additional memory usage is acceptable. Beside memory usage, additional CPU utilization may become critical. The detailed results of the analysis of additional CPU utilization are shown in Fig. 10 and Fig. 11. The maximal additional CPU utilization of a simple node is less than 1%. Thus, the results confirm that also the additional CPU utilization is obviously negligible. If the data of more nodes is managed by the Master, the additional CPU utilization of the simple nodes is the same. Also the utilization of the Super-Node's CPU does not increase significantly. Only the CPU utilization of the Master ascend because of the increasing amount of values and data to handle.

Summarizing, the results show that the proposed data aggregation and management scheme significantly reduces the network traffic overhead created by Integrated Network Control structures, especially for the INControl-Framework, while the performance of the probing and steering elements is not affected and, thereby, the computational overhead created by the proposed scheme is negligible.

7 Conclusion

Efficient data aggregation and management is crucial for distributed network control environments. Therefore, we presented a data aggregation and management scheme that is fitted to the special purpose of distributed network control. The proposed scheme improves the performance of data handling in these environments. Adapting concepts and schemes of data warehousing and on-line analytical processing that are already in use for performance insensible applications, a component has been designed that is integrated into the data management of the Integrated Network Control Framework. Because the scale of aggregation of data is driven by the utilization of any involved element, the introduced overhead can be automatically adjusted and minimized. With this enhancements, an efficient analysis of live and historical data from network measurement and monitoring becomes possible without influencing the network or its hosts negatively. Thereby, the network management can be performed using a broad number of data sources. The improvements of the proposed scheme have been verified within several testbed scenarios. The results of the performance analysis confirm that the data management of Integrated Network Control environments is improved by the proposed scheme.

References

1. Bök, P.-B., Pielken, D., Tüchelmann, Y.: Towards an integrated network control architecture. In: Parallel and Distributed Computing and Networks Conference (2010)
2. Thomas, M.S., Nanda, D., Ali, I.: Development of a data warehouse for non-operational data in power utilities. In: Power India Conference, p. 7. IEEE, Los Alamitos (2006)
3. Xiangdong, Z., Xiao, J.: Process control analysis system based on data warehouse. In: International Conference on Artificial Intelligence and Computational Intelligence, AICI 2009, November 2009, vol. 1, pp. 283–287 (2009)
4. Hamm, C.K., Kennedy, M.R., Wu, T.T., Phillips, J.S.: An operational data store for reporting clinical practice guideline adherence in chronic disease patients. In: Proceedings of the 17th IEEE Symposium on Computer-Based Medical Systems, CBMS 2004, June 2004, pp. 78–83 (2004)
5. Albrecht, J., Bauer, A., Deyerling, O., Gunzel, H., Hummer, Lehner, W., Schlesinger, L.: Management of multidimensional aggregates for efficient online analytical processing. In: International Symposium Proceedings Database Engineering and Applications, IDEAS 1999, August 1999, pp. 156–164 (1999)

6. Chen, Q., Dayal, U., Hsu, M.: A distributed olap infrastructure for e-commerce. In: Proceedings of the 1999 IFCIS International Conference on Cooperative Information Systems, CoopIS 1999, pp. 209–220 (1999)
7. Jianzhong, L., Hong, G.: Range sum query processing in parallel data warehouses. In: Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2003, August 2003, pp. 877–881 (2003)
8. Akinde, M., Bohlen, M., Johnson, T., Lakshmanan, L.V., Srivastava, D.: Efficient olap query processing in distributed data warehouses. In: Proceedings of the 18th International Conference on Data Engineering 2002, p. 262 (2002)
9. He, H., Zhang, Y., Li, L., Ren, J., Hu, C.: An improved storage algorithm for multidimensional data cube. In: Fourth International Conference on Innovative Computing, Information and Control (ICICIC), December 2009, pp. 841–844 (2009)
10. Shimada, T., Tsuji, T., Higuchi, K.: A storage scheme for multidimensional data alleviating dimension dependency. In: Third International Conference on Digital Information Management, ICDIM 2008, November 2008, pp. 662–668 (2008)
11. Yeung, G.C.H., Gruver, W.A., Kotak, D.B.: A multi-agent approach to immediate view maintenance for an operational data store. In: IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th, July 2001, vol. 4, pp. 1869–1874 (2001)
12. Xu, W., Li, M., Wu, S., Zhu, S., Wang, Z., Miao, K., Wang, Y.: Incremental data feed maintenance of a data warehouse system derived from multiple autonomous data sources. In: International Conference on Control and Automation, ICCA 2005, June 2005, vol. 2, pp. 1108–1113 (2005)
13. Hose, K., Klan, D., Sattler, K.-U.: Online tuning of aggregation tables for olap. In: IEEE 25th International Conference on Data Engineering, ICDE 2009, March 2009, pp. 1679–1686 (2009)