High-Speed Access to RFID Data: Meeting Real-Time Requirements in Distributed Value Chains

Holger Ziekow[1,2], Benjamin Fabian[2], and Cristian Müller[2]

[1] International Computer Science Institute, Berkeley, U.S.A.
ziekow@icsi.berkeley.edu
[2] Institut für Wirtschaftsinformatik,
Humboldt-Universität zu Berlin, Berlin, Germany
{ziekow,bfabian,cristian.mueller}@wiwi.hu-berlin.de

Abstract. Using RFID data within operational processes requires fast data access. In distributed value chains, RFID data is not only captured locally, but also accessed from remote locations. However, retrieving data from remote RFID repositories may pose significant delays and slow down the operations. This paper analyses how companies can exchange RFID data in the presence of real-time requirements. We analyze results of performance experiments with globally distributed RFID repositories and propose novel architectures for speeding up data access.

1 Introduction

As companies continue to vertically integrate their processes, they need to exchange information across several steps in their value chains. RFID technology together with the EPCglobal Network promise to improve the exchange for object related data between organizations [1,2]. This network of information sources is based on the Electronic Product Code (EPC), which defines a numbering framework for uniquely identifying objects. RFID tags can store the EPC of the object they are attached to, which provides a unique reference number for managing data about this object. As objects move through the value chain, entities at different locations will collect data about them. Simple examples are records about the arrival of an object at a certain location, or the execution of a certain processing step. One important challenge is to integrate such object data across several value-chain steps and across different locations.

The EPCglobal Network facilitates discovery and retrieval of RFID data from locations across the globe. Core services in this infrastructure are EPC Information Services (EPCIS) [1]. These services provide access to object-related data captured at a certain location or value-chain step, respectively. While these services work well for track and trace applications, their applicability under real-time constraints is questionable. Downstream business operations may require fast access to data records from previous steps. For example, a manufacturer may route items trough its production lines based on object-related information.

R. Meersman, P. Herrero, and T. Dillon (Eds.): OTM 2009 Workshops, LNCS 5872, pp. 142–151, 2009.

Also, companies may quickly check some object-related data at the material intake to reject invalid deliveries right away. In such applications, it is inevitable that the process of accessing EPCIS does not slow down the operations. Even manual operations usually require system responses faster than 0.5 seconds in order to not delay the process [3]. This number can be much lower for automatic processes. Meeting real-time requirements is particularly challenging if the application needs RFID data from very remote locations.

In this paper we address the challenges of using EPCIS for time-critical operations. We specifically take the potentially global distribution of data sources into account. In Fig. 2 we present experimental results about accessing EPCIS around the globe. We propose a range of solutions for accelerating RFID data access in Fig. 3 and discuss their strength and weaknesses. The paper closes with an outlook on our future work.

2 Experiments on Retrieving Distributed RFID Data

We conducted large-scale experiments to test the retrieval of RFID data from remote sources. In our experimental setup, we did deliberately choose not to include the delay caused by an optional EPCIS-discovery phase – either using the standard Object Naming Service (ONS) [1], P2P-ONS [4], or EPCIS Discovery Services [5] – since our particular goals were to test (1) what minimal delays occur in accessing RFID data via EPCIS and (2) how the physical distribution influences the response time of the actual data exchange, even if all data sources are known in advance. Adding a discovery phase will only increase the access time for the whole process, and involves challenges out of the scope of this paper. Particular the aspect of physical distribution of the EPCIS is relevant in global supply chains. In this scenario, network delays due to long distance communication are inevitable. In our experiments we analyze the impact of this effect on real-time requirements for the actual data access.

For the experiments we deployed EPCIS on globally distributed locations and issued simple queries for RFID data (see Fig. 1). As EPCIS software we used the open source implementation provided by Fosstrak [6]. Using the infrastructure of PlanetLab [7] we deployed EPCIS on servers in USA, Germany, China, and Japan. In all regions participated with three to five servers. The variance in set up servers derives from the fact that the nodes on PlanetLab are not reliable. We lost several nodes, even during the short period of time our experiment lasted.

During our experiments we issued queries between the servers and recorded the response times. For a period of three days, we let every server issue one query per minute to the EPCIS on all other servers. We used a very simple EPCIS query that retrieves all data records about a single object (or EPC). All RFID data repositories stored records about two observations of the queried object and consequently the EPCIS returned two records for each request. Following the EPCglobal event model, each record includes the EPC, the event type (or action value, here "OBSERVE"), an identifier of the conducted business step, and a time stamp [8].

Fig. 1. Locations of servers involved in our experiments

Fig. 2 depicts the distribution of response times for all queries in our experiments. The results show that only few queries responded with short delays. Overall, 65% of the queries took longer than 0.5 seconds to respond. Such delays would even slow down manual processes. Note that we tested EPCIS with very simple requests, so we measure delays for EPCIS queries under rather ideal conditions. Consequently, our experiments show results along the lower bound for delays. More complex examples are likely to yield even longer response times.

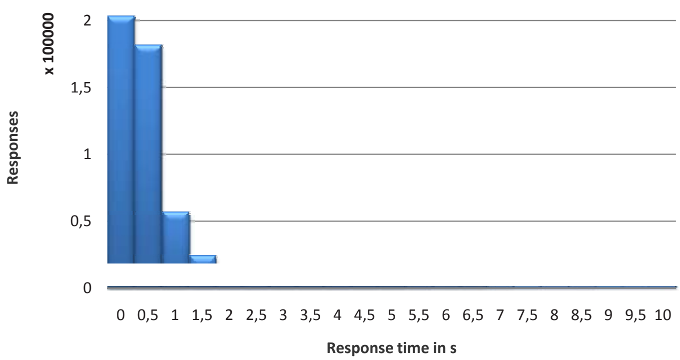

Fig. 2. Response times for EPCIS queries

Our results show that response times in a global setting are often too long for time critical processes. Consequently, time critical processes require measures to speed up data access. A more detailed look at our measurements reveals the cause of long response times and discloses where countermeasures can apply. Table 1 shows an extract of our measurements. It lists the minimum response times for queries between servers in different cities. Connections having response times always above 0.5 sec are highlighted.

A proportion of the delay certainly results from low performance of some involved servers. This is because servers in PlanetLab usually run several experiments by different research groups in parallel, which can result in a slowdown

Table 1. Response times for queries between different locations

from / to	Berlin	Boston	Tokyo	Toyohashi	Osaka	Hong Kong	Beijing
Berlin	70	267	**652**	**658**	**649**	**753**	**805**
New York	162	63	247	252	248	370	350
Boston	257	54	431	449	436	**595**	**579**
Tokyo	**692**	471	104	152	140	299	314
Toyohashi	**642**	439	86	41	73	224	**1024**
Osaka	**641**	254	81	77	48	213	394
Hong Kong	**1271**	**1092**	**597**	**736**	**719**	**655**	**735**

of some query clients and EPCIS servers in our particular experimental setting. The PlanetLab node in Hong Kong is an example where particularly the query client suffered from poor performance (see Table 1). For real world applications, we assume that high performing servers are available and the problem would diminish.

However, Table 1 also shows that the physical distribution of data sources has a significant impact on the response time. This is due to the increased network

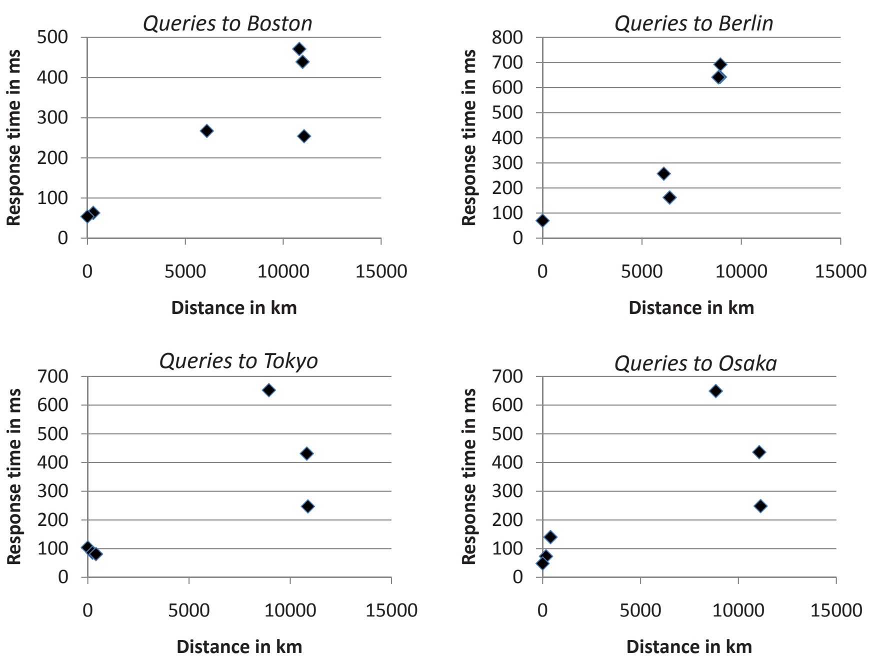

Fig. 3. Minimal response times in dependence on physical distance

delay for long distance communication. Note, that network delay is independent of the used server hardware and inherent in distributed data access. This effect is therefore of major relevance for accessing distributed RFID data. Fig. 3 provides more details. It shows the minimal query response times in dependence of the physical distance. Visualized are results for queries to Boston (top left), Berlin (top right), Tokyo (bottom left), and Osaka (bottom right). The charts show considerable variations between the depicted cases. Yet, in all cases we clearly see the impact of the physical distance.

Our experiments show that the physical distribution of EPCIS servers accounts for a significant proportion in query response times. This effect is of particular importance to applications in distributed value chains. Short response times are often crucial to avoid slow down of operations. Given our performance measurements, it seems currently unlikely that EPCIS-based solutions can support time critical processes in distributed value chains. High performing servers may solve parts of the problem. However, network delay is inherent in accessing remote data sources. This effect appears to be significant in accessing remote EPCIS. To overcome this problem, we are developing several candidate architectures that take locality into account and thereby ensure short query response time. We describe these solutions in the following section.

3 Solutions for High Speed Data Access

Our goal is realizing fast access to RFID data in distributed value chains. When an object arrives at a certain process step, the object-related data must be available in short time. To achieve this goal we must overcome the problem of network delay. In the following, we propose four new architectures that allow answering EPCIS-queries without long-distance communication. The underlying principle of all solutions is to place required RFID data physically close to the query sinks. The solutions vary in the assumptions they make on the underlying processes and in the distribution of load throughout the system. We discuss each solution and illustrate it along simple supply chains. For our illustrations we assume that each supply chain station requires captured RFID data from all previous locations. An example for such applications is the creation of E-pedigrees, for example in the pharmaceutical supply chain [9].

3.1 Pulling RFID Data from EPCIS

Solution 0 shows RFID data retrieval using currently specified services, only. In this solution all supply chain parties pull required information on demand from their predecessors in the chain (see Fig. 4).

As our experiments showed, this solution does not ensure short query response times. We include this solution in our discussion to clearly point out the differences to our alternative propositions.

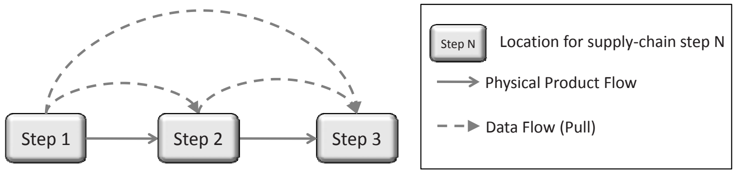

Fig. 4. Data flow in solution 0 (Pull)

3.2 Proactively Pushing RFID Data

In this solution, each supply chain station proactively pushes captured RFID data to locations that will need them (see Fig. 5). This ensures that the RFID data is already locally available when the corresponding object arrives at a certain process step. It is then possible to look for object-related information locally, resulting in very fast response times.

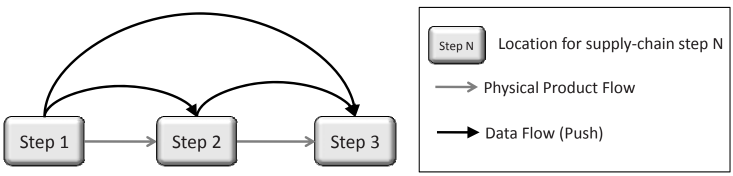

Fig. 5. Data flow in solution 1 (Push)

Because EPCIS do support push-based communication, one can implement this solution solely using EPCglobal standards. However, this solution makes very strong assumptions on the underlying processes. To setup the data distribution, one must know in advance (1) the exact path of each object and (2) the whole downstream information demand. That is, one must know which location will issue which query about which object. This can be viable in rigidly planned processes. However, the solution is infeasible for dynamic and flexible value chains.

3.3 Passing on RFID Data Hop-by-Hop along the Product Flow

In this solution, RFID data follow the product flow hop-by-hop (see Fig. 6). When a product is shipped, captured RFID data are sent along to the designed destination. Note, that this includes RFID data from the current and all previous steps. Thereby, all RFID data for an object reside on a close by server. We leave open if these servers run directly at the involved supply chain station or at a designated third party information broker.

This solution ensures short response times by providing all required RFID data at a close-by server. However, it makes some assumption on the underlying

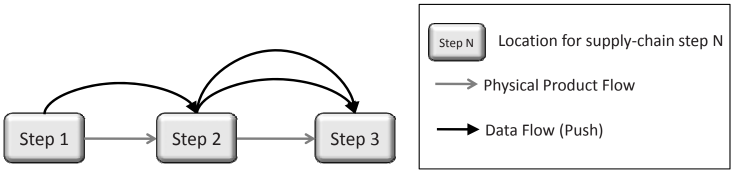

Fig. 6. Data flow in solution 2 (Hop-by-Hop)

business process: (1) each supply chain station must know the immediate next hop in chain. This is generally a practically viable assumption. However, third party logistic providers may introduce some intermediate stations that are unknown to the sender. (2) In order to send the right information, it is necessary to know the whole downstream information demand in advance. Alternatively, one can simply pass on all available information. However, this would cause considerable overhead. (3) Passing all information along the chain may conflict with confidentiality policies. It is required that all downstream players cooperate and are trusted with the information.

Another disadvantage of this solution is the distribution of load throughout the system. Servers downstream in the supply chain handle increasingly high data volumes. This raises questions about how supply chain partners share the burden of running the required servers or the payment for third-party information brokers.

3.4 Using Anycast and Distributed Mirrors

In this solution, each supply chain station pushes captured RFID data to distributed mirrors, e.g. run by one or several information brokers (see Fig. 7). Each query addresses the nearest server using anycast protocols [10]. A simple way of determining the nearest server is to use the physical distance as metric. Mapping servers into a multidimensional cost space may yield better results, but comes at the cost of keeping the servers network coordinates up to date [11].

The advantage of this solution is that it poses no restrictions on the underlying business processes. That is, it requires no knowledge about the product flow or the downstream information demand. Another advantage is that one can implement anycast transparently without changing EPCIS standards.. Query applications can simply use the EPCIS standard without noticing if they access the original data source or a close-by mirror.

On the downside, this approach is somewhat limited with respect to reducing the response times. This is because it makes no assumptions on the origin of future queries. Consequently, one needs a large number of mirrors to ensure low network delay for any possible query sink. With only few mirrors available around the world, it is still possible that certain supply chain locations must access data from a remote mirror. The challenge is finding a good balance between expected delay and the number of mirrors. Favoring certain regions in the placement of mirrors can offer opportunity for optimization. For example, a

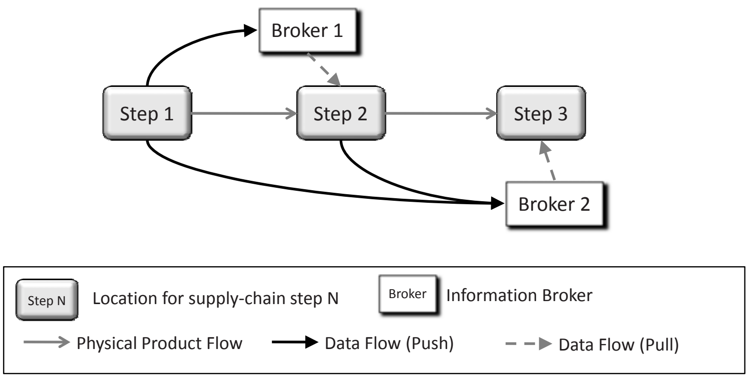

Fig. 7. Data flow in solution 3 (Anycast and Distributed Mirrors)

company may mirror its RFID data at the geometrical centroid (barycenter) of its most important clients.

3.5 Querying One Supply Chain Step in Advance

This solution compromises between push and pull in retrieving RFID data. Each supply chain station N informs its predecessor N-1 about its data demand (transmits the query). When station N-1 ships an object to station N, it pulls all required object data from distributed information sources and pushes them to station N (see Fig. 8 left). Note that to avoid confidential data for station N passing trough station N-1, a trusted third-party information broker (like in [12]) can come into play (see Fig. 8 right).

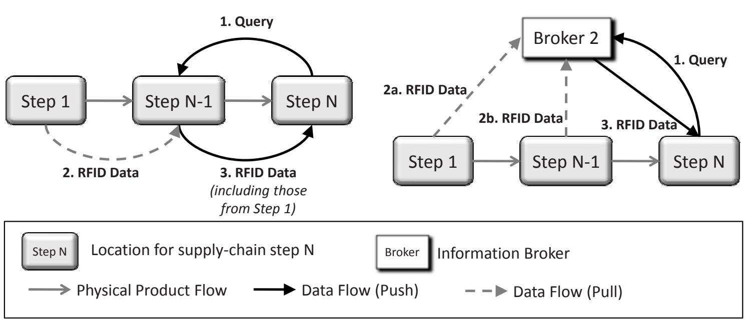

Fig. 8. Data flow in solution 4 (Delegated Query Execution)

The advantage of this solution is that is makes very little assumptions on the underlying business processes. Most of the information flow is pull-based. Thus, upstream supply chain stations need not know the downstream information demand in advance. The only limitations are that (1) each supply chain station must know its predecessor and (2) that the predecessor cooperates. However, these are relatively viable assumptions.

The above-discussed solutions show a spectrum of possible solutions for accessing RFID data. The spectrum ranges from very flexible solutions with potentially long response times to solutions that ensure fast response times but make strong assumptions on the underlying business processes. Table 2 provides an overview of strengths and weaknesses of the different solutions.

Table 2. Strengths and weaknesses of the solutions

Solution	Strength	Weakness
0	- Uses existing EPCglobal standards only. - Is very flexible and poses no limitations to processes.	- Has potentially very slow response times.
1	- Uses existing EPCglobal standards only. - Has fast response times.	- Is very inflexible and poses strong limitations to processes.
2	- Can ensure fast response times (Potentially zero network delay). - Reduces network load trough reuse of event data as it is passed on. - Distributes load.	- Requires that the whole chain supports the service. - Poses some limitations to processes by demanding that the next hop of the product flow is known. - Requires that the downstream information demand is known.
3	- Reduces response times. - Conforms to EPCglobal standards.	- The possible reduction of response times is limited (i.e. zero network delay requires mirrors on all computers in the world).
4	- Can ensure fast response times (potentially zero network delay). - Does not require that the whole chain supports the service.	- Poses some limitations to processes by demanding that of the product flow is known one hop in advance.

4 Conclusions

In time-critical applications, the potentially global distribution of RFID data sources may constitute an important bottleneck for fast data lookup and storage. In this paper, we presented experiments on PlanetLab confirming this intuition. In our opinion it is therefore inevitable to design mechanisms and architectures to answer time-critical queries locally.

We presented and discussed four important solutions to mitigate this problem. Our future work will concentrate on evaluating and comparing these solutions by analytical means, simulation, and extended experiments on real-world testbeds like the PlanetLab, focusing on trade-offs between flexibility and system performance.

References

1. EPCglobal: The EPCglobal Architecture Framework – Version 1.2 (September 2007), http://www.epcglobalinc.org/standards/architecture/
2. Wamba, S., Boeck, H.: Enhancing Information Flow in a Retail Supply Chain Using RFID and the EPC Network: A Proof-of-Concept Approach. Journal of Theoretical and Applied Electronic Commerce Research 3, 92–105 (2008)
3. Ziekow, H., Ivantysynova, L.: Design Guidelines for RFID-Based Applications in Manufacturing. In: 16th European Conference on Information Systems, Galway, Ireland, pp. 2580–2591 (2008)
4. Fabian, B., Günther, O.: Distributed ONS and Its Impact on Privacy. In: IEEE International Conference on Communications (ICC 2007), Glasgow, Scotland, pp. 1223–1228. IEEE Press, Los Alamitos (2007)
5. BRIDGE: BRIDGE Project WP2 – Serial-Level Lookup Service (2009), http://www.bridge-project.eu/index.php/workpackage2/en/
6. Fosstrak: Fosstrak – Free and Open Source Software for Track and Trace (2009), http://www.fosstrak.org/
7. PlanetLab: PlanetLab – An Open Platform for Developing, Deploying, and Accessing Planetary-Scale Services (2009), http://www.planet-lab.org
8. EPCglobal: EPC Information Services (EPCIS) Version 1.01 Specification (September 2007), http://www.epcglobalinc.org/standards/epcis/
9. EPCglobal: Pedigree Ratified Standard – Version 1.0 (January 2007), http://www.epcglobalinc.org/standards/pedigree/
10. Abley, J., Lindqvist, K.: Operation of Anycast Services, Request for Comments, RFC 4786 (December 2006), http://www.ietf.org/rfc/rfc4786.txt
11. Zhang, B., Ng, T.S.E., Nandi, A., Riedi, R., Druschel, P., Wang, G.: Measurement-based Analysis, Modeling, and Synthesis of the Internet Delay Space. In: 6th ACM SIGCOMM Conference on Internet Measurement (IMC 2006), pp. 85–98. ACM Press, New York (2006)
12. Ziekow, H.: In-Network Event Processing in a Peer to Peer Broker Network for the Internet of Things. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2007, Part II. LNCS, vol. 4806, pp. 970–979. Springer, Heidelberg (2007)