Herakles: A System for Sensor-Based Live Sport Analysis using Private Peer-to-Peer Networks

Michael Brand, Tobias Brandt, Carsten Cordes, Marc Wilken, Timo Michelsen
University of Oldenburg, Dept. of Computer Science
Escherweg 2, 26129 Oldenburg, Germany
{michael.brand,tobias.brandt,carsten.cordes,marc.wilken,timo.michelsen}@uni-ol.de

Abstract: Tactical decisions characterize team sports like soccer or basketball profoundly. Analyses of training sessions and matches (e.g., mileage or pass completion rate of a player) form more and more a crucial base for those tactical decisions. Most of the analyses are video-based, resulting in high operating expenses. Additionally, a highly specialized system with a huge amount of system resources like processors and memory is needed. Typically, analysts present the results of the video data in time-outs (e.g., in the half-time break of a soccer match). Therefore, coaches are not able to view statistics during the match.

In this paper we propose the concepts and current state of Herakles, a system for live sport analysis which uses streaming sensor data and a Peer-to-Peer network of conventional and low-cost private machines. Since sensor data is typically of high volume and velocity, we use a distributed data stream management system (DSMS).

The results of the data stream processing are intended for coaches. Therefore, the front-end of Herakles is an application for mobile devices (like smartphones or tablets). Each device is connected with the distributed DSMS, retrieves updates of the results and presents them in real-time. Therefore, Herakles enables the coach to analyze his team during the match and to react immediately with tactical decisions.

1 Introduction

Sport is a highly discussed topic around the world. Fans, the media, teams and athletes discuss about performance, tactical decisions and mistakes. Statistics about sports are an important part of these discussions. Some statistics are calculated manually by people outside the playing field (e.g., counting ball contacts). Other statistics are retrieved automatically by computer-based analysis systems. However, such systems require expensive computers to calculate the statistics due to the huge amount of data to process. Not every team is able to buy and maintain them. Additionally, many systems need too much time to calculate the statistics, making it impossible to deliver them during the game. Many already existing sport analysis systems are using several cameras placed around the playing field. This increases the costs for buying and maintaining the system even further.

In this paper, we propose *Herakles*, a live sport analysis system based solely on conventional low-cost hardware and software we are currently working on. Instead of using costly cameras, Herakles uses a network of sensors attached to all game-relevant objects, e.g., the players and the ball. Since sensor systems are usually not allowed during games and opponents won't wear the sensors anyway, Herakles focuses on the use for tryouts. The active sensors regularly send their actual position, acceleration and other values as *data streams*. Therefore, Herakles uses a distributed *data stream management system* (DSMS). With such a DSMS, it is possible to compute relevant sport statistics in real-time. In our work, real-time means "near to the actual event"(rather than in timeouts or after a match). But using a single machine with a DSMS installed invokes the possibility of single-point-of-failures: (1) a single DSMS instance would run the risk of overload and (2) the live analysis stops completely if the single DSMS instance fails. Therefore, Herakles uses multiple machines to share the processing load and to increase the reliability of the entire system. Currently, many already existing distributed DSMSs are using computer clusters, grids, etc. (e.g., [vdGFW+11]).

Since Herakles focuses on using a collection of smaller low-cost private machines like notebooks, a *private Peer-to-Peer* (P2P) network is more feasible. In our work, each peer is considered to be heterogeneous and autonomous. This makes it possible to use these types of private machines for distributed data stream processing of sensor data for live sport analysis. Although the peers are low-cost private machines, the system costs depend on the used sensor network.

To show the calculated sport statistics in a convenient way for different users, Herakles focuses on mobile devices for the presentation (e.g., smartphones or tablets). With this, the statistics can be directly shown in real-time. Additionally, Herakles abstracts from specific sensors as well as from a specific sport. Statistics for different sports can be calculated using different sensors but based on the same system. Subsequently, these statistics have to be presented in a way that the technological background is encapsulated from the user, e.g., the user does not need to configure the P2P network.

The remainder of this paper is structured as follows: Section 2 gives an overview of related sport analysis systems. The concept of Herakles is explained in Section 3. In Section 4 we give an overview on the current state of our implementation. Section 5 gives a prospect of future work and Section 6 concludes this paper.

2 Related Work

There are many existing sport analysis systems developed and used. However, the majority of them is built on cameras. Many of them are commercial products like Synergy Sports Technology[1] or Keemotion[2]. Other projects are scientific. For instance, Baum [Bau00] provides a video-based system for performance analysis of players and objects. It uses high-speed video cameras and was tested in the context of baseball. As mentioned earlier, there are many problems related to video-based systems, especially the high acquisition and maintenance costs.

Leo et al. [LMS+08] refer to a video-based soccer analysis of players and objects. In their approach, the ball and players are tracked by six video cameras. Then, the video data is

[1] `http://corp.synergysportstech.com/`

[2] `http://www.keemotion.com/`

received by six machines and processed by a central supervisor. The arising problems of this approach are the same as for Baum [Bau00].

Smeaton et al. [SDK^{+}08] propose an analysis of the overall health of the sportsperson in the context of football games. Their sensor-based approach includes body sensors in combination with video recordings. The location is tracked by GPS, but the data can only be reviewed after the game and their approach currently handles only one person at a time.

Von der Grün [vdGFW^{+}11] proposes a sensor-based approach called RedFIR. It uses expensive hardware for the real-time analysis (e.g., SAP HANA for a german soccer team[3]), which leads to high acquisition and maintenance costs like in [Bau00]. Additionally, their data is processed in an in-memory database instead of a DSMS.

To our best knowledge, there is no sport analysis system that uses a network of conventional, low-cost machines for processing streaming sensor data and that shows the results on a mobile device in real-time.

3 Concept

In this section, we give an overview of Herakles. Its architecture is shown in figure 1 and is divided into three components: (1) sensor network, (2) P2P network of DSMSs and (3) mobile device.

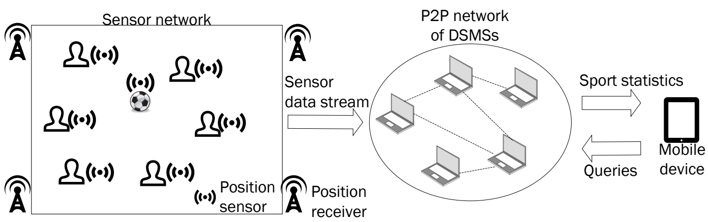

Figure 1: Architecture of Herakles with sensors, a P2P network and a mobile device.

We equip important game entities with sensors to form the sensor network. Each of these sensors sends the position and other information like speed and acceleration of the respective entity. This information is captured by receivers, which are placed around the playing field. These receivers are connected to the next component, the P2P network of conventional (private) machines with DSMSs installed.

The DSMSs receive the sensor data stream. To be independent from a specific sport and sensor technology, Herakles separates the incoming raw sensor data into multiple intermediate schemata at first (data abstraction). The DSMSs use continuous queries built on top of these intermediate schemata to continuously analyze the data and to calculate the sport statistics. With this, we can replace the sensors, but we do not need to change the continuous queries.

[3] http://tinyurl.com/o82mub5

The query results are streamed to the mobile device, the last component in the architecture of Herakles. The mobile device visualizes the statistics and the user can decide at any time, which statistics should be shown at the moment.

Inside Herakles, each statistic is mapped to a continuous query, which is distributed, installed and executed in the P2P network. The distribution is necessary to make sure that not a single peer processes a complete query (avoiding overloads).

In the following sections, we give a more detailed view about the components of Herakles. In section 3.1, we describe our data abstraction. The used P2P network and the distributed DSMS with it's features is explained in section 3.2. Finally, the presentation on mobile devices is explained in section 3.3.

3.1 Data Abstraction

To improve reusability and flexibility, Herakles separates the sensor data into different layers. In doing so, we can avoid direct dependencies between calculated statistics and a sensor's technology and data format. An overview of Herakles' data abstraction is shown in figure 2.

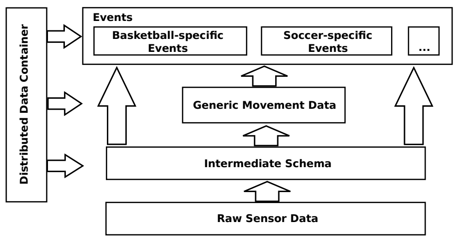

Figure 2: Data abstraction layers ranging from sensor schema up to sport specific schemata.

The sensor network sends its data as *raw sensor data*. Inside the P2P network, this data is converted into a common *intermediate schema* for standardization. This includes for example a unit conversion.

The intermediate schema improves the interchangeability in Herakles: While the sensor's data format can change, the intermediate schema stays unchanged. When different sensors are used, only the adapter between raw sensor data and the intermediate schema has to be rewritten. Other parts of Herakles including the continuous queries for the sport statistics stay unchanged.

In our opinion, generic movement information is interesting for more than one sport and should be reused. Therefore, on top of the intermediate schema, we differentiate between *generic movement data* (e.g., the player's position and running speed) and *sport-specific events* (e.g., shots on goal for soccer). Sport-specific events need to be created for each sport individually (e.g., basketball and soccer). By separating generic movement data from sport-specific events, we can reuse parts of continuous queries in different sports. In general, Herakles uses continuous queries built on each other to transform the data: one continuous query receives the raw sensor data, transforms it into the intermediate schema and provides the results as an artificial data source. This source can be reused from other continuous queries to determine the movement data, identifying sport-specific events etc.

Another aspect of the data abstraction is the storage and the access to static values. For instance, the positions of the soccer-goals are needed in many statistics (e.g., for counting shots on goal or identifying goals itself). Another example is the identification, which sensor is attached to which player or ball. Since Herakles uses a decentralized and dynamic P2P network, a central server is not applicable. Additionally, it is not feasible to define the static values on each peer individually. Therefore, Herakles needs a decentralized way to share these static information across the network.

The data abstraction of Herakles uses a so-called *distributed data container* (DDC). It reads information (e.g., from a file) and distributes them automatically to other peers in the network without further configuration. Then, each peer can use these values inside its own DSMS. Changes at one peer in the network are propagated to the other peers automatically. With this, the user has to define the static values only at one peer.

3.2 Data Stream Management System

Herakles uses a distributed DSMS in a P2P network for processing the sensor data. Since the peers are private machines (e.g., notebooks or mobile devices), the peers are considered to be autonomous and heterogeneous [VLO09].

To use a distributed DSMS in such a P2P network with enough performance, reliability, availability and scalability for live sports analysis, multiple mechanisms must be in place. *Query distribution* makes it possible to use more than one peer for query processing (to share system loads). With *fragmentation*, the data streams can be split and processed in parallel on multiple peers. *Replication* increases the reliability of the system by executing a query on multiple peers at the same time. Therefore, results are available, even if a peer failure occurs [Mic14]. *Recovery* also reacts to peer failures in order to restore the previous state of the distributed system at run-time. *Dynamic load balancing* monitors the resource usage of the peers in the network at run-time and shifts continuous queries from one peer to another.

In our opinion, a P2P network of DSMS instances with all features mentioned above can handle the challenges of processing and analyzing data from active sensors in the context of live sport analysis.

3.3 Presentation

The coach wants an agile and lightweight device to access the data. Therefore, we chose mobile devices like smartphones or tables as front-end. These devices are connected to the P2P network (e.g., via Wi-Fi) and receive the result streams of the continuous queries. The presentation of the calculated sport statistics should be fast and easy to support the coaches' decisions during the game. Therefore, the results have to be aggregated and semantically linked to provide a simple and understandable access to information.

We interviewed different coaches to identify the needed information. We classified those information in (1) player statistics, (2) team statistics and (3) global statistics. *Player statistics* provide information of a specific player (e.g., ball contacts or shots on goal). This view can be used to track the behavior of a single player in significant situations. *Team statistics* provide information about an entire team (e.g., ball possession or pass completion rate). An overview of the current game is provided by the *global statistics* view. All of these views are updated in real-time since the mobile device continuously receives the statistic values.

4 Implementation status

Currently, Herakles is under development and many functions and mechanics are already in place. Therefore, there are no evaluation results available, yet. However, we tested Herakles with the analysis of a self-organized basketball game.

In this section, we give a brief overview of the current state of our implementation. At first, we give a description of our sensor network and data abstraction (section 4.1). The used distributed DSMS and its relevant features are explained in section 4.2. Section 4.3 contains a description about the used sport statistics. Section 4.4 describes the presentation application, which is used for the coach to see the statistics in real-time.

4.1 Sensor networks and data abstraction

We already tested a GPS sensor network for outdoor and a Wi-Fi sensor network for indoor sports. In both sensor networks, position data is provided by mobile applications using sensors of the mobile device. That makes both sensor networks low priced. To test the approach, we equipped 12 players and the ball with a device running the sensor application and let them compete in a short basketball game while recording the resulting data.

The DDC and the different layers of the data abstraction component are implemented. The DDC can be filled in two ways: (1) by reading a file or (2) by messages from other peers. The DDC generates messages to be sent to other DDC instances on other peers resulting in a consistent state through the P2P network. The intermediate schema for the sensor data is shown in listing 1. x, y and z have to be sent by the sensors (but can be converted if they are in a different unit, e.g., in centimeters), whereas ts can be calculated by the DSMS as well as v and a (using two subsequent elements sent by the same sensor). In our opinion, measuring the positions in millimeters and the time in microseconds are sufficient for most sport statistics. Currently, we have implemented adapters to wrap the data streams from the testing source networks mentioned above into our intermediate schema.

Listing 1: Intermediate schema.

```
sid - unique ID
ts  - timestamp [microseconds]
x   - x-position [mm]
y   - y-position [mm]
z   - z-position [mm]
v   - current absolute velocity [mm/s]
a   - current absolute acceleration [mm/s^2]
```

4.2 Distributed Data Stream Management System

For our distributed DSMS, we use *Odysseus*, a highly customizable framework for creating DSMSs. Its architecture consists of easily extensible *bundles*, each of them encapsulating specific functions [AGG+12]. Odysseus provides all basic functions for data stream processing and is already extended for the distributed execution in a P2P network of heterogeneous and autonomous peers. Furthermore, mechanisms for query distribution, fragmentation and replication are already available [Mic14]. Because of its current functionality in addition with the extensibility, we decided to use Odysseus for Herakles.

However, Odysseus had not all features we need to implement a reliable real-time sport analytic system. It did not support dynamic load balancing (for shifting continuous queries at run-time) and recovery. Therefore, we extended Odysseus with these mechanisms.

For the dynamic load balancing, we implemented a communication protocol and a simple load balancing strategy. For recovery, we implemented a combination of active standby and upstream-backup. *Active standby* means to have a continuous query executed multiple times on different peers (similar to replication). However, only the streaming results of one peer are used. But, if that peer fails (or leaves the network on purpose), another peer with a copy of the query replaces it. With *upstream-backup*, a peer saves the stream elements which had been sent to its subsequent peer, until that peer indicates that it has processed said stream elements. If the subsequent peer fails, another peer can install the lost continuous query again and the processing can be redone with the previously saved stream elements.

4.3 Live Statistics

Analyzing the data and calculating useful statistics in terms of continuous queries is an essential part of Herakles. Typically, queries are described by a declarative query language. To avoid complex query declarations on the mobile devices, we implemented an own query language called *SportsQL*. It is a compact language especially designed for sport statistic queries in Herakles. If the user selects a statistic to show, the mobile application of Herakles generates a corresponding SportsQL query, which is sent to an Odysseus-instance. The transition of the SportsQL query to an executable continuous query is done inside of Odysseus. With this decision, the continuous queries and the mobile application are disconnected from each other: we can create and improve the complex continuous queries in Odysseus without changing the mobile application. Furthermore, any other device can also use SportsQL without changing Odysseus. An example of SportsQL is shown in listing 2.

Listing 2: Example of SportsQL for the shots on goal of a specific player with the id 8.

```
{
    "statisticType": "player",
    "gameType": "soccer",
    "entityId": 8,
    "name": "shotsongoal"
}
```

In this query, a mobile device wants a statistic about the current amount of shots on goal for a specific player. The attribute `name` identifies the statistic to generate and the attribute `gameType` identifies the analyzed game (in this case soccer). `StatisticType` differentiates between team statistic, player statistic and global statistic mentioned in section 3.1. In this example, a player statistic is specified. Therefore, an `entityId` refers to a player for which this statistic should be created. It is possible to send further parameters within a SportsQL query such as time and space parameters. This can be used to limit the query results, e.g., for a specific range of time or for a specific part of the game field.

4.4 Presentation

We implemented an extensible Android application intended for tablets and smartphones. Android has been chosen because it is common and allows the application to be run on different kinds of devices. Figure 3 shows a screenshot of our current application.

5 Future Work

In our current implementation of Herakles, we focus on soccer-specific statistics to show the proof of concept. But there are a lot of possibilities for more advanced statistics and views. Depending on the sport, we want to support additional statistics. Consequently, we plan to enhance our mobile application, the SportsQL query language and the corresponding continuous queries in the P2P network.

Herakles works with radio-based sensors reducing the costs compared to video-based systems. Therefore, it focuses on training sessions or friendly games. Nevertheless, the probability to get a license to use such a system within an official, professional match is low

Figure 3: Mobile application for the coach with statistics and a game topview.

whereas video-based systems are more accepted. An extension could be to use a video-based system to get the player positions.

Currently, we do not consider inaccuracies in sensor data streams. For example, some sensors send inaccurate data or no data at all for a certain time. Therefore, there can be anomalies in sensor data [ACFM14] and sport statistics, which should be considered in Herakles in the future.

Additionally, we do not face the problems of security, which rise with the use of private P2P networks. Currently, we expect that each peer is cooperative and does not want to damage the system on purpose. But this assumption must be weakened in the future: sensor data streams have to be encrypted, peers need to be checked (e.g., web of trust [GS00]) and a distributed user management should be in place.

An open issue is the lack of evaluation. Currently, we are implementing the last steps. We made a few tests with a GPS sensor network and are about to begin with the evaluation. We plan to measure how much data Herakles can process in real-time, how fast it can react to sport-events (like interruptions), how accurate the sport statistics are and how much reliability and availability the P2P network really provides.

6 Summary

In professional sports, complex and expensive computer-based analysis systems are used to collect data, to setup statistics, and to compare players. Most of them are video-based and not every team has the opportunity to buy and maintain them. With Herakles, we proposed an alternative sport analysis system, which uses a decentralized and dynamic P2P network of conventional private computers. A sensor network placed on the playing field (e.g., sensors attached to the players and balls) is continuously sending position data to this P2P network. On each peer, a DSMS is installed. This collection of DSMSs is used to cooperatively process the streaming sensor data in real-time. To be independent from specific sensors, we designed a data abstraction layer, which separates the sensors from our continuous queries generating the sport statistics. Herakles presents the statistics on a mobile device, where the user immediately sees those statistics during the game. Users can choose other statistics at any time and the P2P network adapts to these changes automatically.

In the P2P network, we use Odysseus, a component-based framework for developing DSMSs. Despite the data stream processing, it supports continuous query distribution, replication and fragmentation. We added dynamic load balancing and recovery mechanisms to fulfill our requirements. Finally, Herakles uses Android-based mobile devices to show the calculated statistics in real-time.

There are still many tasks to do: primarily, extensions to other sports and statistics. We are about to begin with the evaluation to measure performance of efficiency of Herakles. But we are confident: With Herakles, we show that it is possible to analyze sport events in real-time with commodity hardware like notebooks.

References

[ACFM14] Annalisa Appice, Anna Ciampi, Fabio Fumarola und Donato Malerba. *Data Mining Techniques in Sensor Networks - Summarization, Interpolation and Surveillance*. Springer Briefs in Computer Science. Springer, 2014.

[AGG+12] H.-Jürgen Appelrath, Dennis Geesen, Marco Grawunder, Timo Michelsen und Daniela Nicklas. Odysseus: a highly customizable framework for creating efficient event stream management systems. DEBS '12, Seiten 367–368. ACM, 2012.

[Bau00] C.S. Baum. Sports analysis and testing system, 2000. US Patent 6,042,492.

[GS00] T. Grandison und M. Sloman. A survey of trust in internet applications. *Communications Surveys Tutorials, IEEE*, 3(4):2–16, Fourth 2000.

[LMS+08] Marco Leo, Nicola Mosca, Paolo Spagnolo, Pier Luigi Mazzeo, Tiziana D'Orazio und Arcangelo Distante. Real-time multiview analysis of soccer matches for understanding interactions between ball and players. In *Proceedings of the 2008 international conference on Content-based image and video retrieval*, Seiten 525–534. ACM, 2008.

[Mic14] Timo Michelsen. Data stream processing in dynamic and decentralized peer-to-peer networks. In *Proceedings of the 2014 SIGMOD PhD symposium*, Seiten 1–5. ACM, 2014.

[SDK+08] Alan F. Smeaton, Dermot Diamond, Philip Kelly, Kieran Moran, King-Tong Lau, Deirdre Morris, Niall Moyna, Noel E O'Connor und Ke Zhang. Aggregating multiple body sensors for analysis in sports. 2008.

[vdGFW+11] Thomas von der Grün, Norbert Franke, Daniel Wolf, Nicolas Witt und Andreas Eidloth. A real-time tracking system for football match and training analysis. In *Microelectronic Systems*, Seiten 199–212. Springer, 2011.

[VLO09] Quang Hieu Vu, Mihai Lupu und Beng Chin Ooi. *Peer-to-Peer Computing: Principles and Applications*. Springer, 2009.