

Event Driven Architectures

Ralf Truffer
Pirmin Büeler
HTW Chur

Ralf.truffer@tet.htwchur.ch
Pirmin.bueeler@tet.htwchur.ch

1. Zusammenfassung

Die stetige Miniaturisierung in der Computertechnik macht es möglich Sensornetzwerke mit Zehntausenden Sensorknoten statt mit lediglich 10 Knoten zu konzipieren. Aufgrund der besseren Auflösung lassen sich damit diverse Phänomene viel genauer beobachten.

Das Problem liegt jedoch in der Auswertung der dabei entstehenden Datenflut und infolgedessen in der Gefahr weniger relevante Informationen zu gewinnen, als mit einem kleinen Sensornetzwerk.

Um diese Herausforderung zu meistern, gibt es die Event Driven Architecture (EDA). EDA's bauen mit ESP und CEP auf zwei verschiedenen Ansätzen auf. Bei ESP (Event Stream Processing) geht es darum gewisse Ereignisse aus einer grossen Datenmenge herauszufiltern. Das kann z.B. bedeuten, dass aus den Kamerabildern von unzähligen Überwachungskameras, Bilder mit schnellen Bewegungen herausgefiltert werden.

Beim CEP (Complex Event Processing) werden nicht nur bestimmte Ereignisse herausgefiltert, sondern es werden diese Ereignisse zudem zueinander in zeitliche, räumliche oder kausale Relationen gestellt. So dürfen z.B. bei einem Überwachungssystem ein Ereignis, indem eine Person rennt und ein weiteres Ereignis, indem eine andere Person schreit, keine Alarme generieren. Geschehen jedoch diese beiden Ereignisse zur selben Zeit und am selben Ort, darf ein Alarm ausgelöst werden, da die Wahrscheinlichkeit eines Überfalls um einiges grösser ist.

2. Einleitung

Der stetige Fortschritt und die Miniaturisierung in der Computertechnik werden in der Zukunft den traditionellen Computer und deren Ein- und Ausgabegeräte immer weiter verdrängen. Die Verwendung von so genannten Sensornetzen stellt auf eine gewisse Weise eine Neuerung in der Datenerfassung dar. Im Gegensatz zur klassischen Datenverarbeitung wo die Dateneingabe manuell geschah, ist heute eine weitgehend automatische Datenerfassung möglich. Dies geschieht durch den Einsatz von Sensornetzen, die eine weitaus umfassendere Datenerfassung in Echtzeit ermöglichen und dadurch qualitativ hochwertigere Daten liefern[1]. Mit den heutigen Systemen ist es leider noch nicht

möglich diese riesigen Datenhaufen, welche ein solches Sensornetzwerk erfasst, auszuwerten.

3. Sensornetze

Wird das mooresche Gesetz, welches sagt, dass sich die Leistungsfähigkeit von Prozessoren innert 18 Monaten verdoppelt zutreffen [2]? Wahrscheinlich ja, denn bis jetzt hat es immer zutreffend und wir sind weiter voll im Zeitrahmen.

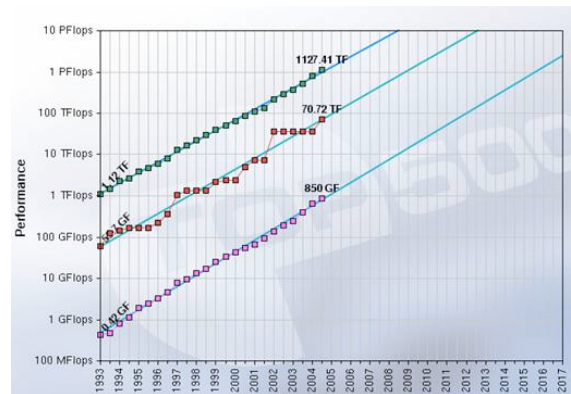


Abb.1: Moorsches Gesetz

Immer kleinere Prozessoren in Verbindung mit neuen Werkstoffen und fortgeschrittener Kommunikationstechnik bilden die Grundlage für Minisensoren, welche in grosser Zahl ein Netzwerk bilden.

Sensornetze bestehen aus einer Vielzahl von Sensorknoten gleicher Bauart.

3.1 Einsatzgebiete

Eingesetzt werden können solche Sensornetzwerke für einfache Überwachungsaufgaben in der Umwelt- und Gebäudetechnik, aber auch für komplexe Beobachtungsaufgaben. Konkret beispielsweise für die Überwachung von Truppen während eines Krieges, Überwachung eines Berghanges, der abrutschgefährdet ist, Beobachtung einer Stadt mittels Kameras oder eine flächendeckende Verkehrsüberwachung zwecks Staufrüherkennung.

Sensornetzwerke werden für diese Aufgaben aufgrund verschiedener Motiven eingesetzt. Zum Beispiel weil ein Gebiet für einen Menschen zu schwer erreichbar ist, zu gefährlich ist oder wenn es um grössere Daten-

ansammlungen geht. Je mehr Sensoren in einem Netzwerk vorhanden sind, desto genauer lässt sich logischerweise ein Phänomen beobachten[3].

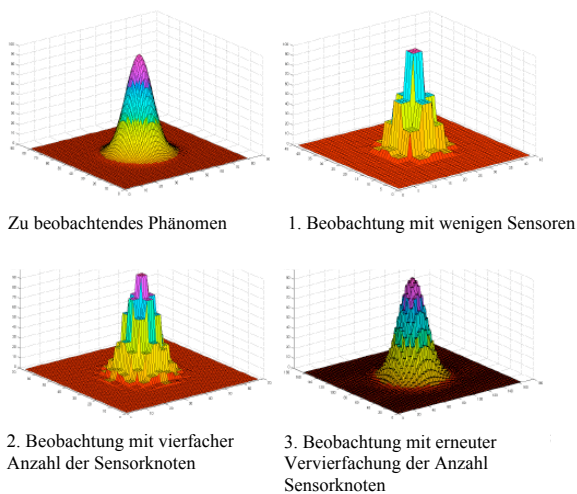


Abb.2: Sensormessungen

3.2 Bestandteile eines Sensornetzwerkes

Sensornetze bestehen aus einer Vielzahl von Sensorknoten gleicher Bauart. Ein Sensorknoten besteht in der Regel aus einem Prozessor, einem Speicher sowie aus einem Funkmodul. Weiterhin besteht er aus einem oder mehreren Sensoren und einer Batterie. Mittlerweile gibt es all diese Bestandteile integriert auf einem Chip, was die Grösse des Knotens entscheidend bestimmt. Der kleinste existierende Knoten weist eine Grösse von nur einem Millimeter aus. Es gibt auch Knoten, die lediglich der Kommunikation und Verwaltung dienen und keinen Sensor beinhalten[1].

3.3 Anforderungen

An die Knoten solcher Sensornetze werden verschiedene Anforderungen gestellt. Der Energieverbrauch der einzelnen Netzwerkkomponenten muss sehr gering sein, damit die Batteriebensdauer mehrere Monate oder sogar Jahre reicht. Es ist bei weit ausgedehnten Sensornetzwerken praktisch nicht möglich eine Batterie auszuwechseln. Die Energie könnte in Form von mechanischer oder thermischer Energie alternativ aus dem Umfeld stammen. In der Regel wird jedoch die Lebensdauer von Knoten zu Ende sein, wenn die Batterie leer ist.

Eine weitere wichtige Anforderung bei Ausdehnungen von 100 000 statt 10 Knoten pro Sensornetz ist der Preis. Dieser müsste sich bestimmt unter einem Franken befinden. Dabei ist zu beachten, dass der grösste Anteil der Kosten von Netzwerken durch die Administration und den Unterhalt verursacht werden. Für ein low-cost System ist also ein selbstkonfigurierendes und selbstunterhaltendes AD-Hoc Netz von Nöten. Selbstkonfigurierend ist definiert durch die Fähigkeit, dass Netzwerkknoten die Präsenz anderer Knoten erkennen und sich selbst organisieren in einem strukturierten und

funktionierenden Netz ohne menschliche Intervention. Selbstunterhaltend steht für die Fähigkeit, dass das Netzwerk Fehler in einem Knoten oder einer Netzwerkverbindung erkennt und behebt ohne menschliches eingreifen [4].

Eine wohl eher futuristische Anforderung an Sensorknoten ist, dass sie biologisch abbaubar wären, damit man sie nach ihrer Lebensdauer nicht zusammensuchen muss.

Die grösste Herausforderung liegt jedoch nicht bei den genannten Kriterien wie Grösse, Preis, Energieverbrauch usw., sondern bei der Auswertung der Daten. Die Schwierigkeit liegt darin in der riesigen Datenflut, die flächendeckende Sensornetzwerke mit sich bringen, die relevanten Informationen herauszudestillieren.

3.4 Netzprotokolle

Aufgrund der genannten Anforderungen, im Speziellen der Wunsch nach sehr energieeffizienten Lösungen, sind für Sensornetze eigene Netzprotokolle nötig, die die Kommunikation zwischen den Knoten regeln.

Ein vollständiges Netzprotokoll beinhaltet folgende Punkte:

Initialisierung: Damit ist das gegenseitige Ausfindigmachen der Sensorknoten gemeint und der Aufbau der Netztopologie. Es handelt sich dabei um ein Ad Hoc - Netz, also um ein Funknetz, dass mehrere Knoten zu einem vermaschten Netz verbindet.

Tagesablauf: Dieser Punkt betrifft den Wechsel zwischen Wach- und Schlafzeiten des Knotens. Durch die Sleep Time kann zwar Energie gespart werden, jedoch sind die Knoten während dieser Zeit für andere Knoten unerreichbar.

Kommunikationsschema: Das Kommunikationsschema regelt den Datenaustausch zwischen den einzelnen Knoten. Dabei ist ein schneller, fehlerfreier und störungsfreier Datentransfer gefordert.

Routing: Durch das Routing wird bestimmt auf welchem Weg die Nachrichten durch das Sensornetz geleitet werden. Dabei ist der schnellste Weg nicht immer der geeignetste. Es ist vielmehr darauf zu achten, dass das komplette Netz gleichmässig belastet wird, da ansonsten ein überbelasteter Knoten aufgrund des begrenzten Energievorrates frühzeitig ausfallen könnte[5].

4. Praktisches Beispiel

Eine besonders typische Anwendung, welche die Problematik von Sensornetzwerken aufzeigt, sind die Überwachungskameras in Grossbritannien. Über das ganze Land sind 4,5 Millionen Kameras verteilt, die Strassen, Bahnhöfe und Einkaufszentren bewachen. Dass diese vielen Kameras eine Unmenge von Bildern liefern, die nur etwas nützen, wenn man sie auch ansieht, versteht sich von selbst. Eine Überwachung von 4.5 Millionen Kameras ist jedoch selbst mit Milliardenbeträgen an finanziellen Mitteln nicht zu bewältigen.

gen. Die Polizei ertrinkt also in der Bilderflut und man spricht längst über die toten Augen von London oder über „Big Brother is not watching“. Scotland Yard spricht sogar von einem vollständigen Fiasko. Die Aufklärungsquote von Überfällen auf offenen Strassen liegt nur bei 3% [6].

Die Schwierigkeit liegt also darin, die wenigen relevanten Bilder aus der Bilderflut herauszukristallisieren. Dies kann schon aus Kostengründen nicht durch das durchstöbern der Bilder mit dem menschlichen Augen geschehen. Es müssen also andere Lösungen gefunden werden.

5. Event Driven Architecture

Die Lösung dieses Problems bieten Event Driven Architectures, kurz EDA's. Wie der Name schon aussagt, handelt es sich dabei um ereignisgetriebene Architekturen, welche je nach Situation auf ein oder unzählige Ereignisse reagieren und dies quasi in Echtzeit. Es werden drei Systeme unterschieden, welche unabhängig voneinander arbeiten können:

- Systeme, die Ereignisse erkennen
- Systeme, die Ereignisse auswerten, aggregieren und falls notwendig neue Ereignisse daraus erzeugen
- Systeme, welche die Verarbeitung eines Ereignisses übernehmen [3]

Für das ereigniserzeugende System ist es somit irrelevant, welche Systeme darauf reagieren und umgekehrt. EDA's sind prinzipiell mit den verschiedensten Technologien umsetzbar. Um der Datenflut von tausenden Kameras Herr zu werden ist ein System einzusetzen, welches Auffälligkeiten in den riesigen Datenströmen ausfindig macht und einen Kontext zwischen ihnen herstellt. Es muss also die Möglichkeit bestehen Event Stream Processing und Complex Event Processing zu verbinden. So kann z.B. aus den Informationen, dass unnatürlich viele und schnelle Bewegungen stattfinden ein neues Ereignis „Schlägerei, Überfall oder Flucht“ erzeugt werden.

5.1 Event Stream Processing (ESP)

Der eigentliche Nutzen eines solchen Überwachungssystems, wie es in den Städten in Gross Britannien eingesetzt wird, geht dadurch verloren, dass die Echtzeitinformationen nicht genutzt werden können. In der neuen Generation des Stream Computing, welche eine neue Form der Datenverarbeitung mit sich bringt, kann dieses Problem gelöst werden.

In der traditionellen Datenverarbeitung werden statische Daten verwendet, welche analysiert werden um eine zeitliche Momentaufnahme zu machen und Fragen zu beantworten. Im erwähnten Beispiel wäre dies vielleicht „wer hat wen Überfallen“.

Beim Stream Computing werden Ereignisse verarbeitet, welche dem Strom von Bildern und Tönen in einem Film ähneln. Ein solcher Strom kann einen Menschen

zum Lachen oder zum Weinen bringen. In ähnlicher Weise kann das ESP Aktivitäten „fühlen“. Es beruht auf völlig anderen physischen Charakteristika als die statische Datenverarbeitung.

Das ESP beschäftigt sich mit dem Filtern unzähliger Ereignisse aus Datenquellen (z.B. Sensoren, Kameras usw.). Es ist nicht jeder Event von Bedeutung sondern der Fluss als solches [7,8].

5.2 Complex Event Processing (CEP)

Beim CEP verhält es sich ein wenig anders. Hier werden bestimmte Muster innerhalb von Ereignisströmen erkannt. Diese Muster werden durch die temporalen, kausalen oder Räumlichen Beziehungen zwischen Ereignissen innerhalb eines Stroms bestimmt. Wird ein Muster erkannt, kann ein neuer Event generiert werden[9].

5.3 Aufbau

Dieses Konzept ist nicht neu. Schon 1985 haben Datenbankforscher damit begonnen rationale Datenbanken durch zeitbezogenen Ergänzungen zu erweitern. Dies scheiterte jedoch auf fatale Weise, da Datenbanken nur dazu dienen Fragen aus der Vergangenheit zu beantworten und nicht Entscheidungen in der Gegenwart zu treffen.

Um solche EDA's realisieren zu können, müssen sieben Elemente vorhanden sein:

Event Processing Engine – Die Event Processing Engine stellt das Hirn dar. Sie überwacht mehrere Ereignisdatenströme und erkennt durch die Anwendung von Regeln Beziehungen und Muster zwischen Ereignissen. Sie kann dabei auf einen Event-Daten-Speicher oder eine rationale Datenbank zugreifen. Dabei werden Events generiert, welche an andere Applikationen gesendet werden können.

Source + Sink Stream Adapters – Die generierten Events werden über einen ESB versandt. Die Adapter werden zum „Abhören“ dieser Ströme genutzt.

Event-Entwicklungs-Tools – Entwickler können mit einer EPL so genannte ESP- und CEP-Regeln (Complex Event Processing) erstellen. Dabei handelt es sich um eine Java-basierte Sprache.

Event-Visualisierung – Benutzer werden mit Hilfe von Alarmmeldungen, die meist in Dashboards dargestellt werden, über Events informiert.

Event-Daten-Speicher – Es müssen alle Events aufgezeichnet werden. Sie sollten nicht in Tabellen oder relationalen Datenbanken, die sich auf statische Attribute konzentrieren, gespeichert werden, da die Zeit als wichtigster Faktor für die Organisation dient.

Cash-Pufferung – Durch die Pufferung kann verhindert werden, dass das ESP System vermehrt Abfragen auf einen relationalen Server senden muss.

Management – Das Management ist ein grundlegender Bestandteil jedes Softwarepakets. Im Bereich von ESP ist das Management mit einigen Problemen ver-

bunden. Da sich z.B. das Wiederherstellen und Verwalten von Eventzuständen als schwierig herausstellt [6].

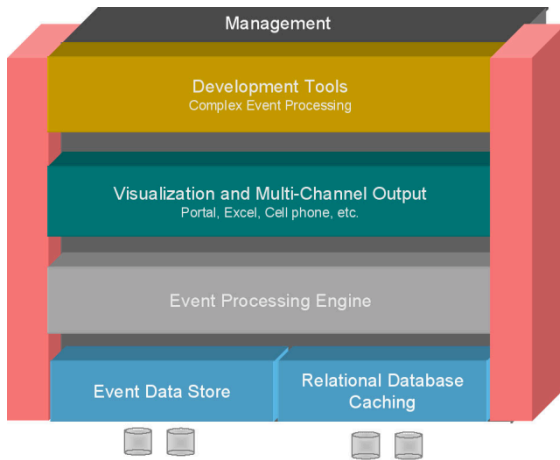


Abb.3: Aufbau

5.4 Beispiel

Was bedeutet all dies, wenn man es auf das konkrete Problem der Überwachung anwendet.

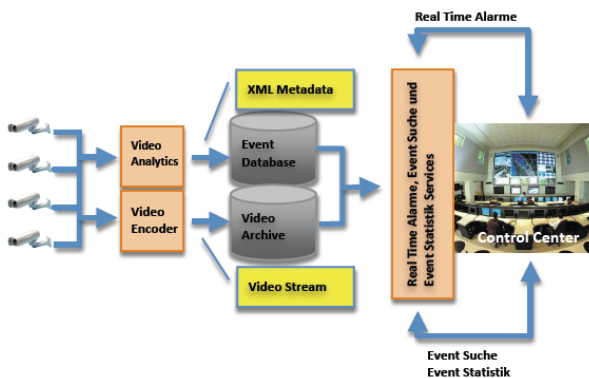


Abb.4: Kamerasensornetz mit ESP/CEP

Die Überwachungsdaten der Kameras werden von einem Video Analyse Server analysiert und über einen Adapter zu XML adaptiert. In diesem Adapter können die Daten „abgehört“ werden. Dieses „Abhören“ übernimmt die Event Processing Engine (Event Database). Sie vergleicht den Datenstrom und die Events mit den in der Event Processing Language definierten Mustern und Regeln. Wird ein Muster oder Zusammenhang erkannt, wird ein Alarm generiert. Weiter soll es auch die Möglichkeit geben statistische Auswertungen der Events zu machen oder nach Events zu suchen. Nebenbei werden auch noch die normalen Video-Streams zur späteren Auswertung aufgezeichnet.

Annahme: Überfall mit Gewaltanwendung

Eine Kamera zeichnet einen Überfall auf. Bei welchem eine Frau überfallen wird. Es kommt zu einem kleinen Kampf, da sich die Frau wehrt. Das ESP registriert jetzt ungewöhnlich viele und schnelle Bewegungen. Dies könnte etwa folgendermassen aussehen: Die unge-

wöhnlichen Bewegungen werden am Bildrand unten-links registriert.



Abb.5: Auswertung

Die Frau liegt nun am Boden und der Räuber flüchtet. Mit dem CEP können nun komplexere Zusammenhänge erkannt werden. Das führt zum Schluss, dass das System während dem Überfall erkennt, dass ein Überfall stattfindet. Und nicht wie bis anhin wo ein Überfall erst registriert wird, wenn das Opfer Anzeige erstattet und die aufgezeichneten Kamerabilder ausgewertet werden. Wenn der Vorfall nicht von einer einzelnen Kamera, sondern von einer Kamera, welche im Verbund mit vielen weiteren ist, registriert wurde, kann der Täter z.B. automatisch über die weiteren Kameras verfolgt werden.

6. Ausblick

Leider wird den Verarbeitungsmodellen ESP und CEP noch zu wenig Aufmerksamkeit geschenkt. Es sind jedoch genau diese Modelle, welche Probleme im Bereich der Real-Time Verarbeitung entscheidend beeinflussen können. Simple Event Processing (Verarbeiten von einfachen Events) ist in einigen Infrastrukturen schon zu finden. Die stromartige Ereigniserkennung und die Mustererkennung sind leider noch nicht so viel vertreten. Vielfach hängt das damit zusammen, dass diese Modelle noch nicht richtig standardisiert sind und die Implementierung sehr kostspielig ist.

Es darf jedoch angenommen werden, dass diese Modelle in der Zukunft eine wichtige Rolle im Bereich der IT und Sensornetze einnehmen.

7. Quellen

- [1] Michal Strucker und Dennis Wagner; „Tiny& Cougar – Datenverarbeitung in Sensornetzen“; Seminararbeit fortgeschrittene Themen der Datenbanken; Dr Thomas Bode, Uni Bonn, Institut für Informatik III.
- [2] Wikipedia; „Moorsches Gesetz“;

- http://de.wikipedia.org/wiki/Mooresches_Gesetz;
zuletzt besucht: 15.05.2008
- [3] Alexander Coers; „Verfahren zur redundanten und distributiven Datenverarbeitung in drahtlosen Sensornetzen“; 12/2005; http://deposit.ddb.de/cgi-bin/dokserv?idn=979602637&dok_var=d1&dok_ext=pdf&filename=979602637.pdf; zuletzt besucht: 31.05.2008
- [4] Edgar H. Callaway, Jr.; „Wireless Sensor Networks“; 2003; ISBN 0-8493-1823-8
- [5] Wikipedia; „Sensornetz“;
<http://de.wikipedia.org/wiki/Sensornetz>
- [6] Süddeutsche Zeitung; „Die toten Augen Londons“; 5/2008
<http://www.sueddeutsche.de/computer/artikel/973/173458/>; zuletzt besucht 31.5.2008
- [7] Roman Roth und Stefan Tilkov; „Ereignis Getriebene Architekturen: Ein Überblick“; 2/2006;
http://www.sigs.de/publications/os/2006/02/roth_tilkov_OS_02_06.pdf; zuletzt besucht: 31.05.2008
- [8] Mark Palmer; Event Stream Processing Tools für eine Ereignis gesteuerte Service-Orientierte Architektur“; 3/2005;
http://www.sigs.de/publications/os/2005/SOA/palmer_SOA_03_05.pdf; zuletzt besucht: 15.05.2008
- [9] Papick G. Taboada und ChristioanDedek; „Esper als EDA-Plattform“; javamagazin 5/2008