

INSTITUT FÜR INFORMATIK
Lehr- und Forschungseinheit für
Programmier- und Modellierungssprachen
Oettingenstraße 67, D-80538 München

————— **LMU**
Ludwig ———
Maximilians—
Universität —
München ———

Datenströme

François Bry, Tim Furche, Dan Olteanu

Technical Report, Institute for Computer Science, Munich, Germany
<http://www.pms.informatik.uni-muenchen.de/publikationen>
Forschungsbericht/Research Report PMS-FB-2004-2, 2004

Datenströme

François Bry, Tim Furche und Dan Olteanu
Institut für Informatik, Universität München, Oettingenstraße 67,
80538 München, <http://www.pms.informatik.uni-muenchen.de>

Unter einem „Datenstrom“ versteht man kontinuierlich übersandte Datensätze, deren Größe, Menge sowie schnelles Aufkommen verbieten, sie vor der Verarbeitung zu speichern. Die bisherige Forschung hat in erster Linie zum Ziel, Verfahren zu entwickeln, die es erlauben, ohne Verzögerung des Datenflusses (1) einen Strom auf das Vorkommen von bestimmten Daten zu überwachen und (2) die Daten aus einem Strom zu analysieren.

Anwendungsgebiete. Datenströme finden sich in vielen Gebieten wie der Nachrichten- (Presse-, Börsen- und Meteorologienachrichten) oder Systemüberwachung und -steuerung (Verkehrs- und Produktionssteuerung, Logistik, Netzwerkverwaltung) und der Analyse von wissenschaftlichen Messdaten (Astronomie, Medizin, Meteorologie, u.a. zur Früherkennung von Tornados). Neue Techniken zur Vermittlung von Datenströmen werden beispielsweise zur Verfolgung von Gepäckstücken oder Paketen entwickelt, die mit (kleinen und billigen) Sendern ausgestattet sind. Presse- oder sonstige Nachrichten werden an Abonnenten als Ströme gesandt, die nach bestimmten Inhalten oder Mustern gefiltert und verteilt werden. Aus Anwendungssicht wird oft zwischen Transaktions- und Messdatenströmen unterschieden. Transaktionsdatenströme sind kontinuierlich ausgesandte Aufzeichnungen (so genannte Log-Daten) über Transaktionen wie Kreditkartennutzung, Telefonanrufe oder Zugriffe auf Webressourcen. Messdatenströme bestehen aus Daten, die Sensoren, Rechnernetzwerke und wissenschaftliche (astronomische, meteorologische oder medizinische) Messstationen liefern. Transaktionsdatenströme und Messdatenströme unterscheiden sich im Hinblick auf ihre Verarbeitung nicht wesentlich.

Push- contra Pull-Kommunikation. Datenströme sind zentraler Bestandteil einer neuartigen Kommunikationsform, der sogenannten „Push-Kommunikation“: Nur die für einen Interessenten relevanten Daten werden automatisch an diesen übersandt, sobald sie zur Verfügung stehen. Für manche Anwendungen ist dieser Ansatz der dem Web zu Grunde liegenden „Pull-Kommunikation“ überlegen, bei der Datenverbraucher die von ihnen gewünschten Daten von den Datenquellen anfordern. Insbesondere angesichts der zunehmenden Informationsfülle wird vermutlich in Zukunft das Informationsbedürfnis von Nutzern durch eine Mischung von Push- und Pull-Kommunikation befriedigt werden.

Punkt-, Tupel- und XML-Ströme. In der Forschung über Datenströme werden derzeit drei komplementäre Arten von Datensätzen betrachtet: Ein Datenstrom wird als eine sehr lange oder ununterbrochene Folge entweder von Punkten (also skalaren Werten wie Zahlen oder Zeichen) oder von Tupeln oder von XML-Dokumenten (oder so genannten XML-Elementen, d.h. wohlgeformten Fragmenten von XML-Dokumenten) angesehen.

Punkt- und Tupelströme bestehen aus Datensätzen, die (1) alle dieselbe Länge haben und (2) „flach“ sind, d.h. keine Schachtelung zulassen. Diese Eigenschaften vereinfachen die Verarbeitung von Strömen. Punktströme können als Sonderfall von Tupelströmen angesehen werden.

Die XML-Dokumente, wie sie in vielen XML-Strömen, insbesondere bei der Überwachung von Messdaten, anzutreffen sind, haben einen geringen Textanteil. XML wird hier also als Formalismus verwendet, um verbundartige Datensätze zu spezifizieren, deren Größe und/oder Schachtelungstiefe möglicherweise unbegrenzt ist und denen eine rekursiv definierte Struktur zu Grunde liegen kann. Wenn diese Datensätze unterschiedlich groß sind, in unterschiedlicher Weise geschachtelt sein oder eine unbestimmte Schachtelungstiefe haben können, dann stellt die Verarbeitung von XML-Strömen eine besondere Herausforderung dar.

Überwachung von Datenströmen oder Anfrageauswertung gegen Datenströme. Die Überwachung eines Datenstroms ist die Suche nach bestimmten Mustern in den übersandten Daten wie etwa Nachrichten über ein bestimmtes Land in einem Strom von Pressemeldungen, große Kurschwankungen in einem Strom von Börsendaten oder bestimmte lebensbedrohliche Wertkombinationen in einem Strom von medizinischen Messdaten. Neben „Überwachung“ werden auch die Begriffe „Filtern“ und „Anfragen“ verwendet. In der Tat handelt es sich dabei um mehr oder weniger komplexe Anfragen, wie sie in Datenbankanwendungen zu finden sind. Es bietet sich daher an, zur Überwachung von Datenströmen die selben Anfragesprachen wie für Datenbankanfragen zu verwenden. In der Datenstromforschung werden derzeit in erster Linie SQL für Anfragen gegen Punkt- und Tupelströme [2] und XPath für Anfragen gegen XML-Ströme [5], sowie mehr oder weniger weitgehende Vereinfachungen dieser Anfragesprachen betrachtet. Für besondere Anwendungsgebiete werden spezielle Anfragesprachen definiert. So wurde z.B. am AT&T eine Anfragesprache namens Hancock mit dem Ziel entwickelt, in Tupelströmen über angerufene Telefonnummern Veränderungen im Benutzerverhalten zu erkennen, die auf Missbrauch hinweisen.

Datenströme stellen neue Herausforderungen für die Anfrageauswertung dar. Neue Verfahren zur Anfrageauswertung werden benötigt, die eine zeitnahe Auswertung von möglicherweise komplexen Anfragen ermöglichen und die Speicherung von Daten und/oder Zwischenergebnissen so weit wie möglich einschränken. Anfragen gegen Datenströme werden manchmal kontinuierliche Anfragen genannt, da die Anfragen ununterbrochen oder kontinuierlich gegen eintreffende Daten ausgewertet werden. Davon unabhängig ist die Frage, zu welchem Zeitpunkt der Nutzer über Antworten informiert wird. Dies kann ebenfalls

kontinuierlich, in vorgegebenen Zeitintervallen, beim Eintreten bestimmter Ereignisse oder sogar nur auf explizite Anforderung des Nutzers geschehen.

Analyse von Datenströmen. Die Analyse von Datenströmen besteht darin, aus Datenströmen aggregierte Werte zu ermitteln. Anwendungen der Datenstromanalyse sind z.B. die Trendanalyse, die Abrechnung über die Nutzung von Rechnern und Rechnernetzen sowie Verkehrswegen, die Systemüberwachung und etwa in der Astronomie oder Meteorologie die Früherkennung von Ereignissen. Wie die Datenstromanfrage verlangt die Datenstromanalyse neue platzsparenden Verfahren, die eine zeitnahe Verarbeitung ermöglichen. Die Datenstromanalyse beruht aber auf herkömmlichen Aggregationsformen.

Sichten. Die Indizierung der Daten in herkömmlichen Datenbanksystemen zur Beschleunigung des Zugriffs auf relevante Daten ist auf einem Datenstrom mit sequenziellem Zugriff nur sehr eingeschränkt möglich. Stattdessen werden manchmal Sichten oder Indizes über die Datensätze eines Datenstroms mit dem Ziel berechnet, jeden Datensatz im Strom mit geeigneten Metadaten anzureichern. Die Verwendung dieser Metadaten kann die Anfrageauswertung oder Datenanalyse beschleunigen, beispielsweise indem irrelevante Daten übersprungen werden.

Merkmale der Datenströmenanfrage und -analyse.

- Einpass-Algorithmen, die mit nur einen einzigen Durchlauf über die Daten auskommen, werden angestrebt, weil sie keine oder nur eine beschränkte Speicherung der Eingabedaten benötigen.
- Änderungen an bereits ermittelten Ergebnissen (wie Antworten zu Anfragen und aggregierten Werten) sind meist ausgeschlossen. Die einzige Datenänderungsart, die meist berücksichtigt wird, ist das Hinzufügen von weiteren Datensätzen.
- So genannte „Fensterverfahren“, die Auszüge mit einer bestimmten Höchstlänge (das „Fenster“) aus einem Datenstrom verarbeiten, werden oft eingesetzt. Dabei wird in Kauf genommen, dass statt exakten Ergebnissen nur angenährte ermittelt werden.
- Verfahren werden oft mittels Automaten, wie endlichen oder Kellerautomaten, formalisiert, die über einfache und eingeschränkte Speichermöglichkeiten verfügen.

Datenstrom- contra Datenbanksysteme. Die Anfrageauswertung und Analyse von Datenströmen stellt ein komplementäres Forschungsgebiet zu der Anfrageauswertung und Datenanalyse (siehe u.a. data warehouses) in Datenbanken dar. Aus praktischer Sicht ergänzt die Anfrageauswertung gegen Datenströme die Anfrageauswertung in Datenbanken, weil Datenströme gefiltert, d.h. angefragt, werden können, um eine Datenbank zu bevölkern. Der Vergleich zwischen der herkömmlichen Anfrageauswertung in Datenbanken und derjenigen gegen

Datenströme ist interessant (siehe Abbildung 1). Während in Datenbanken die Daten in der Regel dauerhaft und zahlreich und die Anfragen flüchtig und gering an Zahl sind, sind bei Datenströmen die Anfragen dauerhaft und zahlreich, die Daten aber flüchtig. Publish-Subscribe-Systeme, die aus Datenströmen (z.B. von Pressenachrichten) Daten gemäß Anfragen von Abonnenten filtern, speichern und aktualisieren, beruhen auf Datenbanken von Anfragen. Traditionelle Datenbanktechniken wie Indizierung können für Anfragen oder Teile von Anfragen angewandt werden. So indiziert das System XTriE [4] XPath-Anfragen. Mit solchen Indizes beschleunigt XTriE die Auswertung großer Mengen von Anfragen gegen den selben XML-Strom.

	Datenstromsysteme	Datenbanksysteme
Daten	flüchtig	dauerhaft
Anfragen	dauerhaft	flüchtig
Änderungen	(meist) auf Hinzufügen beschränkt	beliebig
Ergebnisse	eventuell angenähert	exakt
Datenzugriff	möglichst Einpass-Verfahren	beliebig
Indizierung	der Anfragen	der Daten

Abbildung 1: Vergleich Datenstrom- und Datenbanksysteme

Einige Stromsysteme. Aurora [1] und STREAM [2] werten Anfragen gegen Tupelströme aus. Sie bieten Anfragealgebren mit ähnlichen Operationen wie die relationale Algebra und Fenster-basierte Aggregation an. Aurora kennt Qualitätsmerkmale (u.a. hinsichtlich der Datenrate, Exaktheit der Antwort, Verzögerung), die dazu dienen, eintreffende Tupel zu ignorieren, falls die Daten schneller eintreffen, als die Anfragen ausgewertet werden können. Aurora sammelt Statistiken, um spätere Anfrageauswertungen durch Veränderungen des Auswertungsnetzwerks zu beschleunigen. Abbildung 2 zeigt den gemeinsamen Anfrageplan zur Auswertung zweier Anfragen Q_1 und Q_2 im STREAM-System, wobei Q_1 eine Selektion über einen Verbund der Ströme R und S und Q_2 einen Verbund über die Ströme R , S und T darstellt.

Weitere Tupelstromsysteme sind Gigascope, Hancock, NetFlow, OpenCQ, SEQ, SWAT, Telegraph und Tribeca.

SPEX [5] wertet XPath-Anfragen gegen XML-Ströme aus. XPath-Anfragen werden zuerst in so genannte ForwardXPath-Anfragen übersetzt, die keine Rückwärtsachsen (wie `parent` oder `ancestor`) enthalten. Aus solchen Anfragen werden Netzwerke von Kellerautomaten generiert, wie in Abbildung 3 gezeigt, die einen XML-Strom wie folgt verarbeiten. Jeder Automat leitet den XML-Strom unverändert oder annotiert (d.h. mit Zusatzzeichen ergänzt) an die Automaten weiter, die ihn folgen. Der erste Automat in annotiert den Anfang des ersten Elements; der letzte Automat `out` fügt die berechneten Antwortteilen zusammen. Die Automaten nutzen ihre Keller, um die Tiefe und/oder relative Position der Elemente im XML-Strom zu ermitteln und durch Annotationen zu markieren. So können u.a. Geschwisterbeziehungen (z.B. das nächste „child“) erkannt und weitergegeben werden. Zu einem `or`- oder `and`-Automat gehört ein `cd-or`-

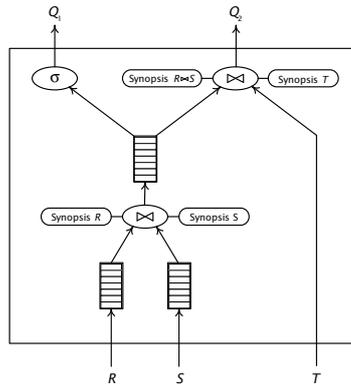


Abbildung 2: Anfrageplan von STREAM zur Auswertung der Anfragen Q_1 und Q_2 gegen die Datenströme R , S und T

bzw. *cd-and*-Automat, welches das Ergebnis entsprechend verknüpft. Die als Ergebnis zu liefernden Elemente werden vom als Box dargestellten Automat ermittelt. SPEX kann XPath-Anfragen mit Prädikaten gegen XML-Ströme von unbegrenzter Tiefe und Größe auswerten. Die kombinierte Zeit- und Speicherkomplexität der SPEX-Anfrageauswertung wächst polynomiell in Länge und Anzahl der XPath-Anfragen. SPEX wird auch zur Auswertung von angenäherten Antworten unter Verwendung von Fensterverfahren eingesetzt.

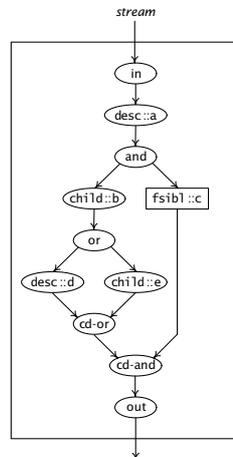


Abbildung 3: Kellerautomatennetzwerk von SPEX zur Auswertung der XPath-Anfrage `/descendant::a[child::b[descendant::d or child::e]]/following-sibling::c`

XMLTK [3] übersetzt mehrere XPath-Anfragen ohne Prädikate in einen einzigen deterministischen endlichen Automaten, der alle Anfragen gegen einen XML-Strom auswertet. Da die Anzahl der Zustände des erzeugten Automaten exponentiell in der Größe der Anfragen wächst, wurde eine „träge“ (lazy) Automatenenerzeugung entwickelt.

Weitere XML-Stromsysteme sind MatchMaker, NiagaraCQ, XPush, XSM, XTrie und YFilter. Alle ermöglichen eine gemeinsame Auswertung von gemein-

samen Teilanfragen.

Literatur

- [1] *Aurora*. <http://www.cs.brown.edu/research/aurora/>.
- [2] *STREAM*. <http://www-db.stanford.edu/stream/>.
- [3] *XMLTK*. <http://xmltk.sourceforge.net/>.
- [4] Chee-Yong Chan, Pascal Felber, Minos Garofalakis, and Rajeev Rastogi. Efficient filtering of XML documents with XPath expressions. In *Proc. of Int. Conf. on Data Engineering (ICDE)*, 2002.
- [5] Dan Olteanu, Tim Furche, and François Bry. An Efficient Single-Pass Query Evaluator for XML Data Streams. In *Proc. of ACM Symposium on Applied Computing (SAC)*, 2004.