Anforderungen an die Anwendungsprotokollierung mittels XML

Daniel Schall
AG Datenbanken und Informationssysteme
Technische Universität Kaiserslautern
Email: d_schall@cs.uni-kl.de

Henrik Loeser
IBM Deutschland Research & Development
D-71032 Böblingen
Email: hloeser@de.ibm.com

21. April 2009

1 Abstract

Anwendungen setzen zunehmend auf Protokollierung ihrer Aktivitäten, sei es aus Gründen der Nachverfolgbarkeit, zu Abrechnungszwecken oder zur Fehleranalyse. Die entstehenden Dokumente sind geschäftskritisch und müssen daher sicher gespeichert werden. Aufgrund seiner Flexibilität hat sich XML in vielen Anwendungsbereichen als bevorzugtes Protokollformat durchgesetzt. In diesem Artikel werden wir zeigen, woher der wachsende Bedarf an Protokollierung stammt und welche Anforderungen an eine Infrastruktur zur Verarbeitung dieser Protokolle bestehen. Wir werden die Bedeutung von XML für diesen Kontext beschreiben und die Vorteile des Formats erläutern. Ein exemplarisches Szenario wird eine einfache Infrastruktur aufzeigen, die sich zur anschaulichen Erläuterung der prinzipiellen Anforderungen nutzen lässt. Abschließend werden einige neue Anforderungen angesprochen, die bislang von Standardsoftware noch nicht hinreichend unterstützt werden und bei denen noch Verbesserungsbedarf besteht.

2 Einleitung

Moderne IT-Infrastrukturen bewegen sich seit Jahren weg von einem zentralisierten Mainframe hin zu einer verteilten Client-Server-Architektur. Zunehmend werden Aufgaben und Prozesse auf mehrere Rechner verteilt. Serviceorientierte Architekturen ermöglichen rechnerübergreifende Prozesse, aber auch Abteilungs-, Standort- oder sogar Unternehmensgrenzen werden überschritten. Diese Trends fördern den Bedarf an Protokolldaten, um geschäftskritische Ereignisse verfolgen zu können.

2.1 Motivation

Viele Unternehmen verfügen über ein verteiltes Netz von Rechner, die Protokolldaten erzeugen. Ein neues und in letzter Zeit häufiger beobachtetes Problem besteht für viele Unternehmen darin, diese Protokolle zentral zu sammeln und auszuwerten (Application Logging). Außerdem müssen die Daten über mehrere Monate sicher gespeichert und jederzeit verfügbar sein. Da die Unternehmen bisher nicht über die notwendige Infrastruktur verfügen, um diese Aufgaben zu erfüllen, musse eine Lösung erarbeitet werden. Zunächst soll jedoch erläutert werden, woher der zunehmende Bedarf an Protokollen stammt.

2.2 Application Logging

Anwendungsprotokolle werden aus vielerlei Gründen erzeugt. Zum einen zur Nachvollziehbarkeit von Vorgängen sowie auf Grund von gesetzlicher Auflagen, zum anderen aus technischen Gründen um den Ablauf von Anwendungen verfolgen zu können.

2.3 Nachvollziehbarkeit

Die Abrechnung von Diensten erfolgt zunehmend nach On-Demand-Regeln und nach tatsächlicher Nutzung. Eine pauschale Abrechnung von Komplettleistung weicht dabei der feingranularen Bereitstellung und entsprechenden Abrechnung von kleinen Service-Einheiten. Ein Beispiel ist der Trend weg vom Komplettkauf eines Server zum flexiblen Einkauf von Rechenzeiten und Speicherplatz in Cloud-Computing-Szenarien. [1] Dadurch werden viele kleine Leistungseinheiten beansprucht, die exakt und nachvollziehbar abgerechnet werden müssen. Gesetzliche Regelungen wie der Sarbanes-Oxley-Act, die Unternehmen zu detaillierten Protokollierung aller finanziellen Aktivitäten zwingen, und die zunehmende beleglose Buchung von Transaktionen fördern ebenso den Bedarf an detaillierten Protokollen. [2] Allein die Finanzindustrie hat über ein Dutzend Standards geschaffen, um den Informationsaustausch zu vereinheitlichen und die vorgeschriebenen Daten zu erfassen. FIXML[1] (Financial Information eXchange Markup Language) oder FpML[2] (Financial Products Markup Language) sind nur zwei Beispiele für neu entstandene Protokollformate. Der Inhalt der Protokollen orientiert sich dabei am Prozess und den zur Durchführung der Aktivität benötigten Daten wie etwa Buchungsnummern, Beträgen und workflow-spezifische Angaben. [3]

2.3.1 Technische Motive

Auf der anderen Seite erfordern verteilte Anwendungen eine Verfolgung ihrer Aktivitäten zur Ablaufanalyse und Fehlerbehebung. Hier sind technische Daten, wie etwa Bearbeitungszeit, Serverauslastung oder beanspruchter Speicherplatz, von hohem Interesse. Für diese Art der Protokolldaten gibt es keine gesetzlich vorgegebenen Aufbewahrungsfristen. Sicherheitskritische Daten wie etwa Zugriffsprotokolle oder Fehleranalysen können jedoch ebenfalls über einen längeren Zeitraum von Interesse sein. Ein Standardformat für technische Protokolle hat sich noch nicht durchgesetzt.

3 Eigenschaften der Protokolle

Als Formatsprache der Protokolldateien hat sich XML durchgesetzt, da es flexible Dateiformate mit optionalen und wiederholenden Elementen erlaubt und zahlreiche Erweiterungen zur Auswertung und Transformation der Daten bietet. Zu nennen sind hier XML Schema zur Definition von XML-Datenstrukturen, die XQuery-Abfragesprache zur Auswertung von XML-Dokumenten, sowie XSLT zur Umformung von XML-Dokumenten in andere Formate. Trotz verbreiteter XML-Standards wie das erwähnte FIXML, sind die entstehenden Protokolldateien sehr heterogen. So umfasst die Version 4.4 von FIXML bereits 1310 verschiedene XML Typen und 41 XSD Dateien.
Das Headerformat der Nachrichten ist dabei meist identisch, der Inhalt variiert.
Anwendungen generierten meist hybride Protokolldateien, die sowohl geschäftskritische, als auch technische Daten in einer Protokolldatei vereinen.
Spitzenlasten von bis zu 500 Nachrichten pro Sekunde sind ebenso möglich wie ein Gesamtaufkommen von bis zu 20 Millionen Nachrichten pro Geschäftstag.
Bei einer durchschnittlichen Größe von 4 - 20 kByte pro Nachricht können also monatlich ca. 7

[1] `http://www.fixprotocol.org/`
[2] `http://www.fpml.org/`

Terabyte an Protokolldateien anfallen. [4]
Die Lebensdauer der Protokolle ist je nach Typ sehr unterschiedlich. Teils ist eine Mindestvorhaltedauer gesetzlich geregelt, teils bestimmt die Verwendung der Protokolle deren Lebensdauer. So sind technische Protokolldateien im Normalfall eher kurzlebig im Bereich von Stunden bis einigen Tagen, geschäftliche Protokolldaten erfordern teilweise eine Lebensdauer von Jahren oder sogar Jahrzehnten. In Deutschland sind beispielsweise alle Rechnungen, gestellt oder erhalten, zehn Jahre lang aufzubewahren. Darüber hinaus müssen diese den Finanzbehörden auf Verlangen unverzüglich zur Verfügung gestellt werden. [5]

3.1 Log Shipping

Da die Nachrichten zum Großteil von Anwendungen generiert werden, die auf speziell angepasster Hardware ausgeführt werden, wie etwa Geldautomaten oder integrierte Schaltungen, können diese weder ausreichend Speicherplatz noch Verarbeitungskapazitäten bereitstellen, um die Nachrichten direkt am Ort ihrer Entstehung verarbeiten zu können. Die Nachrichten müssen daher über ein Netzwerk an einen (zentralisierten) Speicherort weitergeleitet werden, bevor die Auswertung und Archivierung der Daten erfolgen kann. Da sich die Netzwerklandschaft der Kunden über ein Netz aus Filialen, Tochterfirmen und ausgelagerten Betriebsstätten erstreckt, besteht der Wunsch, alle anfallenden Protokolldaten vorab zu filtern und die geschäftskritischen Anteile in die Hauptverwaltung zur Auswertung und Sicherung zu leiten. Der Verlust von Nachrichten muss dabei ausgeschlossen werden.

4 Infrastruktur

Eine Infrastruktur, die Funktionen für das Sammeln der Protokolle bereitstellt und in einen Backend-Speicher leitet, wird benötigt. Die Anforderungen an diese und der Entwurf einer Infrastruktur basierend auf IBM Technologien wird im Folgenden vorgestellt.

4.1 Anforderungen an die Infrastruktur

Basierend auf den Eigenschaften der Protkolle und den Bedürfnissen der Unternehmen, ergeben sich für die Infrastruktur zur Anwendungsprotokollierung folgende Anforderungen: Die Infrastruktur muss eine **Schnittstelle für die Annahme von Protokolldateien** bieten, sodass Anwendungen Daten übergeben können. Alle Arbeitsschritte sollen wenn möglich **Unterstützung für verteilte Transaktionen** bieten um einen Verlust von Nachrichten ausschließen zu können. Ein **Caching** von Nachrichten auf Clientseite ist hilfreich um Netzwerke zu entlasten und Clients von der Infrastruktur zu entkoppeln. Ebenso kann **Routing** helfen, die Protokolle zu verschiedenen Back-End-Speichern zu leiten. Die Infrastruktur kann durch **Transformation** die Nachrichten in ein einheitliches Format bringen, bevor diese gespeichert werden. Insbesondere können dadurch beliebige Protokollformate in XML überführt werden um eine einheitliche Verarbeitung zu ermöglichen. Gleichzeitig muss die Infrastruktur eine **hohe Durchsatzrate** erreichen, um alle anfallenden Protokolle verarbeiten zu können. Eine **effiziente Auswertung** und **persistente Speicherung** der Protokolle muss trotz der großen Datenmenge gewährleistet werden, die Speicherung auf Bandlaufwerken oder in Dateiordnern scheidet daher für aktuelle Protokolldateien aus. Weiterhin soll die Infrastruktur eine gute **Verteilbarkeit** aufweisen und mit wachsender Anzahl der Clients **Skalierbarkeit** bringen. Da die Protokolldaten im XML-Format vorliegen, soll die Infrastruktur eine gute Unterstützung dieses Formats bieten. Daher sind XQuery-Unterstützung, XSLT und die Unterstützung von XML-Schemata wünschenswert.

4.2 Entwurf einer Infrastruktur

Die vorgestellten Anforderungen an die Infrastruktur sind bisher von keiner eigenständigen Softwarelösung zu erbringen. Durch eine Kombination mehrerer bekannter Technologien lässt sich ein grundlegendes Szenario für das Application Logging jedoch einfach entwerfen. Abbildung 1 zeigt den Systementwurf, der im Folgenden vorgestellt wird.

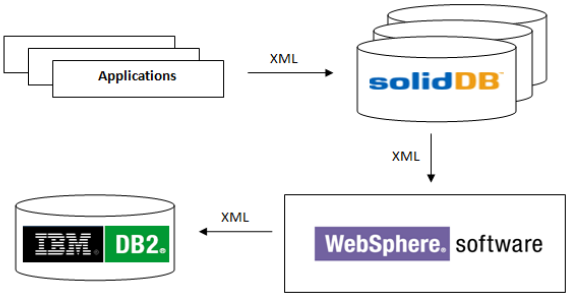

Abbildung 1: Szenario

Als Cache und zur Entkopplung der Anwendungen vom Protokollversand wird IBMs solidDB Datenbank eingesetzt. Diese In-Memory-Datenbanken dienen als Zwischenspeicher zur Vermeidung von Lastspitzen. WebSphere Message Queue dient als persistente Warteschlange, die Nachrichten speichert und über große Distanzen leiten kann. WebSphere Message Broker (WMB) wird eingesetzt, um die Protokolle zu unterschiedlichen Zielen routen, den Inhalt analysieren und Nachrichten in andere Formate überführen. WMB unterstützt XML-Daten und bietet dneben dem Import von XML-Schemata, XSLT-Umformungen und XQuery außerdem eine native Unterstützung für XML-Daten. WebSphere Message Broker wird auch eingesetzt, um die Nachrichten in die Back-End-Datenbank zu schreiben. Da der Message Broker verteilte Transaktionen (XA) unterstützt, ist ein Verlust von Nachrichten ausgeschlossen. Als persistenter Back-End-Datenspeicher wird IBMs DB2 for Linux, Unix and Windows eingesetzt. Dieses Datenbanksystem ist einerseits durch die pureXML-Technologie in der Lage, XML-Nachrichten nativ zu speichern und andererseits die aufkommende Datenlast durch Partitionierung der Datenbank aufzuteilen. Die Partitionierung von XML-Daten ist in der neuesten Version der DB2 hinzugekommen und erlaubt es nun, auch Tabellen, die XML-Spalten enthalten, zu partitionieren. Da die DB2 auch das Erstellen von XML-Indices erlaubt, ist eine effiziente Analyse der Datenmengen möglich.

5 Fazit

Es wurden die Anforderungen an eine Application Logging Infrastruktur vorgestellt und es konnte gezeigt werden, dass der Aufbau einer Infrastruktur möglich ist. Die vorgestellte Infrastruktur erfüllt die prinzipiellen Anforderungen an die Anwendungsprotokollierung, allerdings bleiben noch viele Möglichkeiten für Verbesserungen. Eine detaillierte Ausarbeitung und Anpassung auf kundenspezifische Gegebenheiten steht bei diesem generischen Ansatz noch aus, ebenso die Auswertung der XML-Daten im Backend. Auch wurden Sicherheitsaspekte in diesem Entwurf nicht betrachtet.

6 Ausblick

Die professionelle Anwendungsprotokollierung ist ein bisher kaum beachtetes Arbeitsfeld. Zwar bieten einzelne Produkte (SAP, DB2, Microsoft Windows, etc.) Tools zur Auswertung anwendungseigener Protokolle, eine allgemeine Lösung für die Auswertung von beliebigen Protokol-

len ist allerdings noch nicht verfügbar. Auch beim Log Shipping setzen Hersteller auf eigene Lösungen. Eine integrierte Infrastruktur, die definierte Schnittstellen für Clientanwendungen bietet ist derzeit ebenfalls noch nicht absehbar. Wünschenswert wäre eine Standardisierung der Middleware für die Anwendungsprotokollierung, wie schon in einigen Arbeiten angedacht. [6] Einheitliche Benchmarks wie TPoX, ein Benchmark für XML-Datenbanken, die speziell auf die Probleme und Anforderungen der Protokollierung verteilter Anwendungen abzielen, sind ebenso wie ein einheitliches, domänenübergreifendes Protokolldateiformat, das geschäftlichen und technische Informationsbedürfnisse vereinen kann, noch zu entwickeln. [7] Die Speicherung großer relationaler Datenmengen ist zwar von DBM-Systemen unterstützt und praxiserprobt, zur Speicherung großer Mengen von XML-Daten gibt es jedoch weit weniger Erfahrung. Speziell XML-Warehousing und die Echtzeitanalyse von XML-Daten sind weit weniger ausgereift als ihre relationalen Entsprechungen. [8]

7 Zusammenfassung

Es wurde gezeigt, woher der wachsende Bedarf an Anwendungsprotokollen stammt, welche Anforderungen an die Anwendungsprotokollierung bestehen und wie eine unterstützende Infrastruktur basierend auf Standardsoftware aufgebaut werden kann. Die Bedeutung von XML in diesem Kontext wurde erläutert und gezeigt, wie die Infrastruktur die Vorteile von XML nutzen kann.

Deutlich wurde, dass das Application Logging noch Potential bietet, insbesondere die Vereinheitlichung der Infrastruktur, sowie verbesserte Auswertungsmöglichkeiten sind wünschenswert.

Literatur

[1] Hayes and Brian. Cloud computing. *Commun. ACM*, 51(7):9–11, 2008.

[2] Rakesh Agrawal, Christopher Johnson, Jerry Kiernan, and Frank Leymann. Taming Compliance with Sarbanes-Oxley Internal Controls Using Database Technology. In *ICDE '06: Proceedings of the 22nd International Conference on Data Engineering*, page 92, Washington, DC, USA, 2006. IEEE Computer Society.

[3] A. Dan, D. M. Dias, R. Kearney, T. C. Lau, T. N. Nguyen, F. N. Parr, M. W. Sachs, and H. H. Shaikh. Business-to-business integration with tpaml and a business-to-business protocol framework. *IBM Syst. J.*, 40(1):68–90, 2001.

[4] Henrik Loeser, Matthias Nicola, and Jana Fitzgerald. Index Challenges in Native XML Database Systems. *BTW 2009*, März 2009.

[5] Deutsches Umsatzsteuergesetz, §14b, „Aufbewahrung von Rechnungen“.

[6] Marcelo Pitanga Alves, Paulo F. Pires, Flávia Coimbra Delicato, and Maria Luiza Machado Campos. Middlog: A web service approach for application logging. In Kurt Bauknecht, Birgit Pröll, and Hannes Werthner, editors, *EC-Web*, volume 3590 of *Lecture Notes in Computer Science*, pages 337–347. Springer, 2005.

[7] Matthias Nicola, Irina Kogan, and Berni Schiefer. An xml transaction processing benchmark. In *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 937–948, New York, NY, USA, 2007. ACM.

[8] Byung-Kwon Park, Hyoil Han, and Il-Yeol Song. *XML-OLAP: A Multidimensional Analysis Framework for XML Warehouses*, volume 3589. 2005.