



WebSphere software

Bereitstellung eines Messaging- Backbone für SOA-Konnektivität

Inhalt	
2	<i>Kurzübersicht</i>
4	<i>Messaging-Backbone: ein Einstieg in SOA</i>
8	<i>Messaging-Konzepte</i>
16	<i>Erfüllung von Unternehmensanforderungen</i>
23	<i>Nahezu grenzenlose Konnektivität</i>
24	<i>Schnittstellen und Standards</i>
26	<i>Messaging für Web-Services</i>
27	<i>Zuverlässige Übertragung von Dateien</i>
29	<i>Basis für Ihren ESB</i>
31	<i>Zusammenfassung</i>
31	<i>Weitere Informationen</i>

Kurzübersicht

Derzeit besitzen Sie in Ihrem Unternehmen eventuell eine disparate, weit verteilte, zunehmend komplexe IT-Infrastruktur – eine Infrastruktur, die sich aus Systemen unterschiedlicher Art zusammensetzt, deren Standorte und Verwaltungen auf verschiedene Abteilungen und Orte verteilt sind. Ein flexibler, leistungsfähiger Messaging-Backbone versetzt Sie in die Lage, schnell neue Anwendungen einzubinden und vorhandene Anwendungen kosteneffizient zu nutzen, bei minimalem Risiko für die Geschäftsdaten. Diese Anwendungen können Sie in Ihr gesamtes Unternehmen und mit denen Ihrer wichtigsten Handelspartner, Lieferanten und Kunden integrieren.

Um wettbewerbsfähig zu bleiben, können Sie sich bei der Verwaltung von Informationen, die auf eine Vielzahl getrennter Systeme verteilt sind, nicht mehr auf manuelle Prozesse verlassen: Diese erfordern kostenintensive Wartung, sind anfälliger für Benutzerfehler und nicht für zukünftiges Wachstum gerüstet. Ein starker Messaging-Backbone kann Sie mit dem sicheren Fundament ausstatten, auf dem Sie Ihre Produkte und Services über das Internet bereitstellen, die Interaktionen effektiver gestalten, kritische Prozesse optimieren und die Produktivität in der gesamten Wertschöpfungskette steigern können. So kann sich der Fluss der Transaktionen, Informationen und Ideen unverzüglich in Ihrem Unternehmen ausbreiten – und darüber hinaus.

Viele Unternehmen denken an eine serviceorientierte Architektur (SOA), um die Flexibilität ihrer IT zu erhöhen. Daher wird es immer wichtiger, das Neue (serviceorientierte Anwendungen und Ressourcen) mit dem Alten (vorhandenen IT-Ressourcen, die das Lebenselixier Ihres Unternehmens darstellen) verknüpfen zu können. SOA erhöht die Interoperabilität der IT-Ressourcen und verbessert die Möglichkeit, diese dynamisch miteinander zu verbinden. Deshalb ist es wichtiger als je zuvor, einen leistungsfähigen, zuverlässigen Backbone für diese Interaktionen bereitzustellen, der bei jeder Interaktion von Services und nicht-serviceorientierten Ressourcen das geschäftsentscheidende Merkmal der Zuverlässigkeit bewahrt.

IBM WebSphere MQ ist ein flexibler, leistungsfähiger Messaging-Backbone, der für die Integration von SOA und Ihren vorhandenen IT-Ressourcen sorgt. WebSphere MQ ist der branchenführende Messaging-Backbone mit garantierter exakt einmaliger Auslieferung von Daten für ein breites Spektrum von Betriebssystemen. In allen Branchen, von Banken über Telekommunikationsunternehmen bis zu Regierungsbehörden, haben IT-Abteilungen davon profitiert, dass sie zur Verbindung unterschiedlicher Systeme einheitliche Technologien einsetzen.

Im letzten Jahrzehnt hat sich die Softwarebranche auf bestimmte Standards geeinigt: Web-Services für das Auffinden und den Aufruf von Services, Java™ 2 Platform Enterprise Edition (J2EE) als Programmiermodell für Unternehmen oder XML als kanonisches Datenformat. Messaging-Technologie – insbesondere WebSphere MQ – ergänzt diese Standards: Sie spielt eine wichtige Rolle beim Aufbau einer starken, offenen IT-Infrastruktur.

Das vorliegende White Paper untersucht den Nutzen der Implementierung eines Messaging-Backbone, verglichen mit dem Aufbau eines komplexen Netzes mittels eigener Programmentwicklungen zur Integration Ihrer Geschäftsprozesse. WebSphere MQ und die wesentlichen Konzepte, die dieses Produkt beinhaltet, werden ausführlicher behandelt. Das White Paper geht auch auf einige der Standards ein, die zurzeit die Branche prägen – und darauf, warum der Messaging-Backbone nach wie vor ein wesentliches Element des Ganzen darstellt, besonders in einer SOA. Schließlich ermöglicht es Einblicke in das Branchenumfeld insgesamt, um deutlicher zu machen, was WebSphere MQ ist, wie es in die IT-Branche von heute passt und welchen Nutzen Sie davon haben.

Messaging-Backbone: ein Einstieg in SOA

Die Geschäftsziele von heute bestimmen die IT-Prioritäten. Wettbewerbsdruck und Marktkräfte drängen Führungskräfte dazu, ihre Unternehmen radikal umzubauen. Das Ziel ist strategische Flexibilität durch Innovationen. Und eine SOA sorgt dafür, dass die IT schneller auf Änderungen reagieren kann. Üblicherweise wird der Weg zu einer flexiblen IT durch ein wahres Dickicht von Anwendungen versperrt, die in isolierten Silos entwickelt wurden und häufig auf einzelne Abteilungen ausgerichtet sind. Erhebliche Anteile der IT-Budgets werden lediglich für die Aufrechterhaltung der Konnektivität aufgewandt, die Kosten für deren Erweiterung steigen meist mit dem Wachstum.

Der Messaging-Backbone ermöglicht den Einstieg in SOA (Abbildung 1). Er überbrückt die Kluft zwischen neuen, serviceorientierten Ressourcen und den vorhandenen Kernressourcen und bildet die Transportschicht für einen Enterprise Service Bus (ESB). Außerdem entlastet er Anwendungen von der Konnektivitätslogik, mit der geregelt wird, wie jede Anwendung mit den anderen kommuniziert.

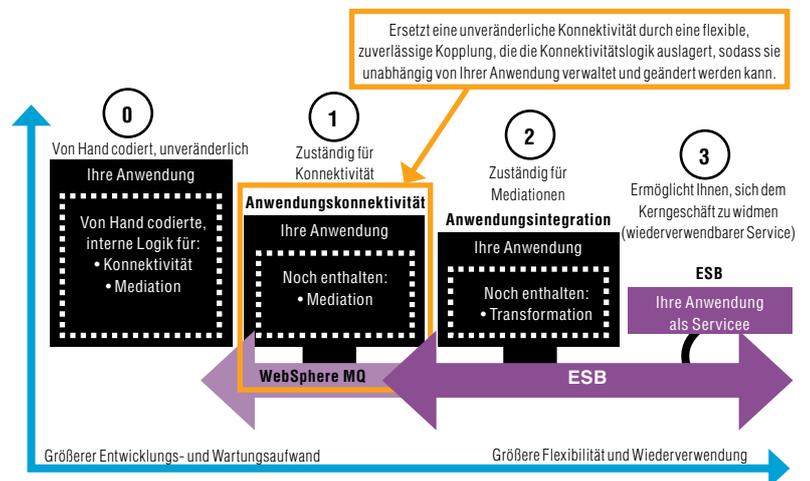


Abbildung 1. WebSphere MQ ermöglicht den Einstieg in SOA.

Ein Messaging-Backbone ermöglicht Ihnen die Wiederverwendung bereits vorhandener Ressourcen. Dadurch sind Sie nicht gezwungen, Anwendungen mit hohem Aufwand vollständig zu ersetzen, und können den Investitionsertrag der bestehenden Anwendungen steigern. Der Websphere MQ Messaging-Backbone unterstützt nahezu jedes kommerzielle IT-System; daher brauchen Sie auch Ihre Hardware und Ihr Betriebssystem nicht zu ersetzen. Das breite Angebot von Schnittstellen in zahlreichen gängigen Programmiersprachen ermöglicht es Ihnen, verfügbares Know-how weiterhin einsetzen zu können.

Geschäftsprozesse gewinnen an Zuverlässigkeit durch einen Messaging-Backbone, der sicherstellt, dass die Daten übermittelt werden und die Transaktionen erhalten bleiben, selbst wenn diese Prozesse mehrere IT-Systeme betreffen, die sich über Ihr gesamtes Unternehmen erstrecken. Wenn Sie das Risiko verringern, dass IT-Systeme ihre Integrität verlieren, sind Sie besser auf die Herausforderungen der Einhaltung gesetzlicher Bestimmungen vorbereitet, für welche die Finanzberichte auf nachweisbar korrekten Geschäftsdaten basieren müssen.

Die Vorteile von Messaging-Software werden deutlicher, wenn Sie die Alternativen bedenken. In den meisten Unternehmen stehen mehrere Systeme, Anwendungen oder Automatisierungsinself eigenständig nebeneinander – häufig auf verschiedenen Betriebssystemen. Die Daten befinden sich in der Regel an mehreren Stellen, was Duplizierungs- und Synchronisierungsprobleme mit sich bringt. Mitarbeiter geben Daten manuell in mehrere unterschiedliche Systeme ein. Wenn Sie eine neue Anwendung entwickeln oder kaufen – oder wenn ein Zusammenschluss oder eine Übernahme eintritt –, wird die Situation noch komplexer. Zur Lösung des Problems müssen Sie Ihre Anwendungen miteinander vernetzen, damit Informationen zwischen ihnen ausgetauscht und die über Ihr Unternehmen verteilten Daten überall verfügbar gemacht werden können.

Um dieses Konnektivitätsniveau zu erreichen, hätten Sie die Möglichkeit, Code zu schreiben und in Ihre Anwendung einzufügen, um mit anderen Systemen zu kommunizieren. Demzufolge müssten Ihre Entwickler eine Konnektivitätslogik schreiben – und sich mit den Grundbestandteilen eines außerordentlich schwierigen Gebietes der Softwareentwicklung auseinandersetzen. Hinzu kämen Problemstellungen wie die Behandlung von TCP/IP-Sockets, die stark von dem Betriebssystem und der verwendeten Programmiersprache abhängen. Außerdem würde ein breites Spektrum spezieller Qualifikationen in Ihrem Entwicklerteam erforderlich sein. Der Code muss mit Situationen umgehen können, in denen das Netz ausfällt oder die empfangenden Anwendungen nicht verfügbar sind. Jedes Element der Konnektivitätslogik ist nur auf die dadurch verbundenen Anwendungen bezogen. Dies setzt der Wiederverwendbarkeit Grenzen und erschwert das Hinzufügen von Anwendungen bei einer Änderung der Geschäftsanforderungen. Die wahrscheinliche Folge wäre, dass Ihre IT-Mitarbeiter in hohem Umfang komplexe, unhandliche Konnektivitätslogik schreiben, verwalten, erweitern und pflegen müssten.

Sie können dieser Situation entgehen, indem Sie ein Softwareprodukt einsetzen, das Ihnen alle diese Konnektivitätsprobleme abnimmt – einen Messaging-Backbone. Statt Ihre Anwendungen mit Konnektivitätslogik zu überfrachten, lassen Sie diese – über eine einfache, einheitliche Anwendungsprogrammierschnittstelle (API) – mit dem Backbone kommunizieren, um die Daten an andere Anwendungen zu übermitteln.

Wenn Sie einen Messaging-Backbone implementieren, können Sie standardisierte Programmiermodelle übernehmen und diese auf einer Auswahl von Betriebssystemen verfügbar machen. Ein effektiver Messaging-Backbone sollte universell sein, damit das Know-how und der Code in Ihrem Unternehmen weitestgehend wiederverwendet werden können. Ihre Anwendungsentwickler brauchen sich nur dem Schreiben der Geschäftslogik zu widmen, ohne in großem Umfang Konnektivitätscode verwalten zu müssen.

Ein entscheidender Aspekt des Messaging-Backbone ist die Zuverlässigkeit der Nachrichtenübermittlung. Sie müssen in der Lage sein, bei der Datenbereitstellung die erforderliche Servicequalität sicherzustellen. Bei nicht kritischen Daten ist es möglicherweise nicht notwendig, diese mit Empfangsbestätigung zu senden, auch wenn Ihnen bewusst ist, dass die Daten angesichts bestimmter Störungsszenarien verloren gehen könnten. Wenn es jedoch um kritische Geschäftsinformationen wie eine Banktransaktion geht, möchten Sie sich auf die exakt einmalige Übermittlung verlassen können. Wenn die Anwendung diese kritischen Daten an die Messaging-Ebene sendet, sollte die Verarbeitung fortgesetzt werden. Wenn Messaging-Ereignisse – etwa das Senden und Empfangen von Daten – auch als Bestandteile einer Transaktion fungieren können, kann sichergestellt werden, dass Aktionen wie Datenbankaktualisierungen in derselben Unit of Work wie Messaging-Operationen auftreten können, mit koordiniertem Commit oder Rollback.

Ein weiteres zentrales Konzept des Messaging-Backbone ist die zeitunabhängige – oder asynchrone – Verarbeitung. Dies bedeutet, dass eine Anwendung nicht auf die Verfügbarkeit der jeweils anderen Anwendung oder auf die Verfügbarkeit des Netzes angewiesen ist, um Daten zu senden. In einem rein synchronen Modell würden Ihre Anwendungen bei einem Netzausfall eine ausgefeilte Wiederholungslogik benötigen und wären eventuell blockiert, während sie auf die Wiederherstellung des Netzes warteten. Asynchrones Messaging lässt sich am besten als Auslieferungsmodell verstehen, nicht als das Gegensatz zu synchronem Messaging. Asynchrones Messaging entkoppelt einfach die Anwendungen voneinander und vom Netz. Es arbeitet nach dem schnellstmöglichen Übermittlungsmodell. Wenn Sie Daten von Anwendung A an Anwendung B senden möchten und das Netz verfügbar ist, werden die Daten fast unverzüglich übermittelt. Wenn jedoch das Netz oder die empfangende Anwendung nicht verfügbar ist, ist die sendende Anwendung davon nicht notwendigerweise betroffen. Der Messaging-Backbone nimmt bei Bedarf eine Zwischenspeicherung der Daten vor.

Da Ihr Messaging-Backbone Ihre kritischen Geschäftsdaten verarbeitet, sollten Sie sicherstellen, dass die von Ihnen eingesetzte Infrastruktur zuverlässig ist. Sie müssen sicherstellen, dass die Daten nicht unwiederbringlich verloren gehen können. Ihre Messaging-Software muss nach einem Failover-Modell arbeiten. Sie benötigt eine integrierte Funktionalität, die hohe Verfügbarkeit sicherstellt. Eventuell sind Sie auch an Funktionen interessiert, die den Lastausgleich unterstützen, damit Sie die Gewissheit haben, dass Ihr Messaging-Backbone wachsen und sich anpassen kann, wenn sich Ihre Geschäftsanforderungen ändern.

Messaging-Konzepte

Die wesentlichen Komponenten eines Messaging-Backbone sind Nachrichten und Warteschlangen. Der folgende Abschnitt geht kurz auf diese und einige weitere Schlüsselkomponenten ein. Detailliertere Informationen erhalten Sie in den WebSphere MQ-Produkt-handbüchern.

Nachrichten

WebSphere MQ versetzt Anwendungen und Services in die Lage, über das Senden von Nachrichten zu kommunizieren, statt sich direkt gegenseitig aufzurufen. Nachrichten sind nichts anderes als Folgen von Bytes, die die Daten enthalten, die Sie von einer Anwendung zur anderen übertragen möchten. Eine Nachricht besteht aus zwei Teilen: dem Header, der die Nachricht beschreibt, und den Daten selbst, bei denen es sich um Daten beliebigen Typs handeln kann, etwa XML-Daten, Binärdaten oder einen Bitstrom. Der Nachrichten-Header identifiziert die Nachricht durch eine eindeutige Nachrichten-ID und enthält weitere Felder, etwa den Nachrichtentyp, Angaben zu ihrem Ursprung, der Priorität, der Verfallszeit, der für Antworten vorgesehenen Warteschlange und so weiter.

Jede WebSphere MQ-Nachricht kann bis zu 100 MB groß sein. Größere Nachrichten und Dateien können transportiert werden, indem sie in kleinere Blöcke aufgeteilt werden. WebSphere MQ kann große Nachrichten automatisch aufteilen; die empfangenden Anwendungen haben die Wahl, die größeren Nachrichten abzurufen, nachdem diese durch WebSphere MQ wieder kombiniert worden sind, oder jedes Segment einzeln zu empfangen. Alternativ kann die Programmsteuerung dafür sorgen, dass Nachrichten in Segmente aufgeteilt werden, abhängig von logischen Grenzen oder der Größe des für die empfangende Anwendung verfügbaren Puffers. WebSphere MQ stellt sicher, dass die Reihenfolge der Segmente erhalten bleibt.

In ähnlicher Weise lässt sich der Nachrichtenverkehr verringern, wenn viele sehr kleine Nachrichten an denselben Empfänger gesendet werden, indem diese Daten zu größeren WebSphere MQ-Nachrichten zusammengefasst werden. Wenn die zusammengesetzte Nachricht beim Empfänger ankommt, zerlegt WebSphere MQ die Nachricht und kann auch hier die Reihenfolge beibehalten, in der die enthaltenen Nachrichten abgesendet wurden.

Verteilerlisten sind eine weitere Möglichkeit, den Nachrichtenverkehr zu reduzieren. Wenn dieselbe Nachricht an mehrere Warteschlangen, die von demselben Warteschlangenmanager verwaltet werden, gesendet wird, genügt es, eine einzige Kopie dieser Nachricht an den Warteschlangenmanager zu senden. WebSphere MQ verwendet eine Verteilerliste, um zu ermitteln, welche der eigenen Warteschlangen eine Kopie dieser Nachricht bekommen soll, und trifft die entsprechenden Maßnahmen. Diese Verteilerlisten können bei Bedarf jederzeit aktualisiert werden.

Nachrichtenpersistenz

WebSphere MQ kann Nachrichten sowohl auf persistente als auch auf nicht persistente Weise verarbeiten. Alle persistenten Nachrichten werden durch WebSphere MQ protokolliert. Dabei werden diese Nachrichten synchron, zum Zeitpunkt des Versendens, auf die Platte oder einen anderen nicht flüchtigen Speicher geschrieben. Diese Funktionalität ermöglicht die Übermittlung und Wiederherstellung von Nachrichten selbst dann, wenn die Anwendungen, die Netze oder der WebSphere MQ-Server in irgendeiner Übertragungsphase ausfallen.

Nicht persistente Nachrichten werden nicht auf diese Weise protokolliert; allerdings werden auch diese Nachrichten nur einmal übermittelt, um Problemen mit Nachrichten aus dem Weg zu gehen, die doppelt ankommen und so dazu führen, dass Transaktionen teilweise mehrmals ausgeführt werden.

Auch semipersistentes Messaging ist möglich, falls ein differenzierterer Kompromiss zwischen der Zuverlässigkeit der Übermittlung und dem Durchsatz des Messaging-Systems erforderlich ist. Der Durchsatz von semipersistentem Messaging ist in der Regel höher als bei persistenten Nachrichten, weil die Nachricht nicht an allen Stellen während der Übermittlung, an denen eventuell Störungen auftreten könnten, protokolliert wird. Im Fall semipersistenter Übertragung erfolgt die Protokollierung asynchron – meist nachdem die Nachricht tatsächlich versendet wurde. Diese Funktionalität kann die Durchsatzgeschwindigkeit erhöhen, denn sie muss nicht vor dem Senden auf den Abschluss von Ein-/Ausgabeoperationen der Platte warten. Auf der anderen Seite besteht ein gewisses Risiko, dass die Nachrichten und der Warteschlangenmanager nicht vollständig wiederhergestellt werden können, falls in dem kurzen Zeitraum vor der Aktualisierung des Protokolls eine Störung auftritt.



Typen von Warteschlangen

Die geläufigsten Warteschlangentypen:

- *Lokale Warteschlangen*, die konkret auf einem bestimmten System vorhandenen Warteschlangen
- *Aliaswarteschlangen*, Alternativname zur Bezeichnung lokaler Warteschlangen
- *Ferne Warteschlangen*, Verweise oder Kennungen für Warteschlangen auf anderen Systemen
- *Übertragungwarteschlangen*, spezielle Warteschlangen, die WebSphere MQ intern verwendet, um Nachrichten in einem Netz zu bewegen (dort befinden sich die Nachrichten, falls das Netz nicht verfügbar ist)
- *Dynamische Warteschlangen*, lokale Warteschlangen, die on demand erstellt werden
- *Warteschlangen für nicht zustellbare Nachrichten*, Warteschlangen zur Speicherung von Nachrichten, die nicht zugestellt werden können (etwa wenn eine Zielwarteschlange voll ist und nicht vergrößert werden kann)
- *Repository-Warteschlangen*, Warteschlangen für Clusterangaben

Es ist wichtig, die Persistenzstufe zu kennen, die von jeder Anwendung und jedem Service benötigt wird, und den entsprechenden Persistenzservice auszuwählen, damit der Messaging-Backbone optimal arbeitet und genau die im Unternehmen erforderliche Wiederherstellbarkeit gewährleistet. Standardmäßig verwendet WebSphere MQ persistentes Messaging. Um darüber hinaus präzise Leistungsvergleiche von Messaging-Backbones vornehmen zu können, sollten Sie funktional entsprechende Persistenzstufen einsetzen.

Warteschlangen

Mithilfe von WebSphere MQ können Anwendungen und Services asynchron miteinander kommunizieren, ohne dass alle zur selben Zeit verfügbar sein müssen. Diese Funktionalität wird ermöglicht durch Warteschlangen, also durch Datenstrukturen zur Speicherung von Nachrichten. Wenn sich Nachrichten durch den WebSphere MQ-Backbone bewegen, werden (selbst für einen kurzen Moment) Warteschlangen zu deren Zwischenspeicherung eingesetzt. Auf diese Weise können die Nachrichten wieder abgerufen werden, sobald der Empfänger verfügbar ist. Im Regelfall, wenn alle sendenden und empfangenden Anwendungen verfügbar sind, kommen die Nachrichten, die über diese Warteschlangen versendet werden, nahezu in Echtzeit beim Empfänger an. Falls jedoch Anwendungen oder das Netz ausgelastet oder nicht verfügbar sind, ist die Warteschlange in der Lage, die Nachrichten solange aufzubewahren, bis sie empfangen und verarbeitet werden können. Dieses asynchrone Messaging-Modell ist ein leistungsfähiges Verfahren, Anwendungen und Services flexibel zu koppeln, sodass ihre Kommunikation zeitunabhängig erfolgt.

Infolgedessen kann der Messaging-Backbone Anwendungen und Services gegen Unterbrechungen und Störungen abschirmen und sorgt dafür, dass sie sinnvoll weiterarbeiten, statt durch den Datenaustausch blockiert zu sein. Darüber hinaus können Sie WebSphere MQ zum Start von Anwendungen – über den Weg der Anwendungssteuerung – veranlassen, wenn genügend Nachrichten zur Verarbeitung vorliegen. Dabei behält WebSphere MQ die Reihenfolge der Nachrichten bei; sie werden in derselben Reihenfolge, in der sie versendet wurden, an die Anwendungen übermittelt. Standardmäßig wird die Strategie First In/First Out (FIFO) befolgt.

Warteschlangenmanager

Den Kern eines Messaging-Backbone bilden seine Warteschlangenmanager, welcher den Messaging-Service bereitstellt und Objekte wie Warteschlangen und Kanäle verwalten. Warteschlangenmanager setzen Übertragungswarteschlangen ein, um Nachrichten an ferne Warteschlangen zu übergeben, die von anderen Warteschlangenmanagern verwaltet werden. Sie stellen Auslöseservices bereit, damit Anwendungen gestartet werden können, wenn genügend Nachrichten zur Verarbeitung angekommen sind. Außerdem übernehmen sie die Konvertierung der in den Nachrichten enthaltenen Zeichensätze zwischen verschiedenen Plattformen. Auf verteilten Systemen können die Warteschlangenmanager von WebSphere MQ als Transaktionskoordinatoren fungieren, die mit zweiphasigem Commit arbeiten, um die Transaktionalität der Operationen in Datenbanken und Warteschlangen zu bewahren.

Warteschlangenmanager kümmern sich um die Wiederherstellung, die Persistenz und die zuverlässige Übermittlung von Nachrichten. Bei persistentem oder semipersistentem Messaging protokolliert der Warteschlangenmanager die Nachrichtendaten auf die Platte. Die Warteschlangenmanager von WebSphere MQ werden in Hochverfügbarkeitsumgebungen häufig durch Backups gesichert.

Kanäle

WebSphere MQ verwendet Kanäle, um seine Warteschlangenmanager anzubinden und um die WebSphere MQ-Clients mit diesen zu verbinden. Kanäle sind logische Kommunikationsverbindungen. Ein Nachrichtenkanal ist als Verbindung eines Warteschlangenmanagers mit einem anderen definiert; dies wird als Kommunikation zwischen Servern bezeichnet. Diese Kanäle sind unidirektional, sie werden häufig paarweise definiert. An beiden Enden der Nachrichtenkanäle koordinieren Sendee- und Empfangsagenten – Listener genannt – die Kommunikationsverbindung. Auch WebSphere MQ-Clients verwenden Kanäle, um die Verbindung zu den Warteschlangenmanagern der WebSphere MQ-Server herzustellen; allerdings handelt es sich in diesem Fall um einen anderen Kanaltyp, denn Clients haben keine Warteschlangenmanager. Clientkanäle sind bidirektional. Einige Kanäle können von WebSphere MQ automatisch definiert werden. Warteschlangenmanager enthalten einen Nachrichtenkanalagenten (Message Channel Agent, MCA), der für Kanäle zuständig ist.

Clients

WebSphere MQ unterstützt eine Vielzahl von Clients, sodass sich Anwendungen und Services mit seinem Messaging-Backbone verbinden können. WebSphere MQ-Clients können sich mit fernen Warteschlangenmanagern verbinden. WebSphere MQ-Clients sind für zahlreiche Plattformen im Produktumfang enthalten und können zudem gebührenfrei heruntergeladen werden. Die meisten Clients werden vollständig unterstützt.

Der Unterschied zwischen Clients und Servern liegt hauptsächlich darin, dass Clients im Gegensatz zu Servern keine Warteschlangenmanager besitzen. Dies bedeutet, dass Clients keine lokale Warteschlange für die Speicherung von Nachrichten bereitstellen und daher kein asynchrones Messaging zwischen Client und den damit verbundenen Servern unterstützen. Clients stellen ihre Verbindungen über dedizierte, bidirektionale Kanäle her und können sich nur mit WebSphere MQ-Servern verbinden, wenn das Netz verfügbar ist. In diesem Fall kann die Anwendung oder der Service den Verbindungsaufbau wiederholen oder eine andere Route wählen, um die Messaging-Server zu erreichen.

Ob ein WebSphere MQ-Server mit seinem Warteschlangenmanager oder eher ein Client vorzuziehen ist, hängt von der Übermittlungsqualität ab, die von den lokalen Anwendungen und Services auf einem bestimmten System benötigt wird. Falls für dieses System eine zuverlässige Nachrichtenübermittlung erforderlich ist, ist ein Client vermutlich für Ihre Anforderungen ungeeignet.

WebSphere MQ unterstützt einen speziellen Client, erweiterter Transaktionsclient genannt. Wie andere Clients auch stellt er keinen Warteschlangenmanager bereit. Erweiterte Transaktionsclients ermöglichen den Anwendungen die Ausführung mehrerer Tasks innerhalb einer Transaktion, die aus einer einzigen Arbeitseinheit (Unit of Work, UOW) besteht. Dazu gehört die Übertragung von Nachrichten an einen Server oder die Aktualisierung der Ressourcen eines anderen Ressourcenmanagers, etwa eines Transaktionskoordinators, eines Datenbankservers oder eines Anwendungsservers. Diese Funktionalität kann die Transaktionalität einer Verbindung erweitern und den Einsatz eines lokalen Warteschlangenmanagers überflüssig machen. Erweiterte Transaktionsclients sind ein Bestandteil der Lizenz für WebSphere MQ. Bei der Installation von WebSphere MQ können Sie wählen, ob Sie einen Server oder einen erweiterten Transaktionsclient einsetzen möchten.

Weitere Informationen zu Clients finden Sie unter ibm.com/webspheremq/clients

Messaging-Topologien

Topologien beziehen sich auf die Form des Messaging-Backbone, also darauf, wie seine Übertragungsknoten miteinander verbunden sind. Verschiedene Übertragungskonzepte und -topologien haben ihre jeweils eigenen Vorzüge und können innerhalb Ihres Messaging-Backbone kombiniert werden. Abbildung 2 zeigt Beispiele verschiedener Messaging-Topologien.

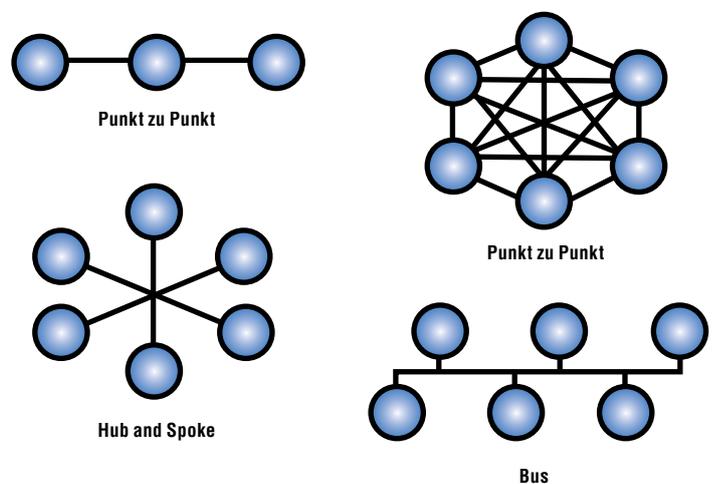


Abbildung 2. Messaging-Backbone-Topologien.

Die Punkt-zu-Punkt-Topologie ist eine einfache Verbindung zwischen zwei Übertragungsknoten oder Anwendungen. Die Logik, die entscheidet, wohin die Nachrichten gesendet werden, befindet sich entweder in der Ursprungsanwendung selbst oder in den Messaging-Komponenten, die lokal für diese Anwendung definiert wurden. Falls die Zielanwendung an eine neue Position verlagert wird, muss auch die Logik der Quellenanwendung aktualisiert werden, damit die Nachrichten sie weiterhin erreichen. Diese Funktionalität ist in einfachen Szenarien von Nutzen, bei denen nur wenige Anwendungen verbunden sind. Um allerdings n Anwendungen miteinander zu verbinden, müssen Sie $n(n-1)/2$ Verbindungen in Ihrer Messaging-Middleware definieren, damit jeder Punkt mit jedem anderen Punkt verbunden ist. Um einige Beispiele zu geben: Bei 2 Anwendungen muss nur eine Verbindung hergestellt werden, bei 5 Anwendungen sind es 10 Verbindungen und bei 10 Anwendungen werden 45 Verbindungen benötigt. Das Hinzufügen einer weiteren Anwendung erfordert nun die Definition 10 zusätzlicher Verbindungen und zieht wahrscheinlich Änderungen bei den meisten der vorhandenen Verbindungen nach sich. Nimmt man die explosionsartige Zunahme der Konfigurationsdaten und die Notwendigkeit zahlreicher Aktualisierungen bei jeder sich ändernden Anwendung zusammen, so ist diese Topologie bei größeren Implementierungen keine ideale Lösung.

In einer Hub-and-Spoke-Topologie wird jede Anwendung mit einem zentralen Knoten im Messaging-Backbone verbunden. Dies reduziert die Anzahl der benötigten Verbindungen für eine bestimmte Zahl von Anwendungen, denn die Anzahl der Verbindungen ist nur so groß wie die Zahl der zu verbindenden Anwendungen. Somit brauchen in dem Beispiel mit 10 Anwendungen nur 10 Verbindungen mit dem Messaging-Backbone definiert zu werden. Das Hinzufügen einer weiteren Anwendung bedeutet lediglich die Definition einer zusätzlichen Verbindung – nur ein Zehntel des Aufwandes bei einer Punkt-zu-Punkt-Topologie. Die Logik, die entscheidet, wohin die Nachrichten gesendet werden, kann zentral im Netzknoten angesiedelt werden. Auch wenn bei dieser Topologie die Einrichtung zu Beginn etwas aufwendiger sein mag, kommt sie mit geringerer Rekonfiguration aus, wenn Anwendungen hinzugefügt oder geändert werden. Single Points of Failure können durch Clustering mehrerer Warteschlangenmanager beseitigt werden.

Auch in einer Bustopologie, ebenso wie in einer Hub-and-Spoke-Topologie, ist die Anzahl der Verbindungen, die zwischen einer bestimmten Zahl von Anwendungen benötigt werden, nur so groß wie die Anzahl der zu verbindenden Anwendungen. Im Wesentlichen entspricht diese Topologie dem Hub-and-Spoke-Typ; allerdings legt die Bustopologie größeres Gewicht darauf, die Konnektivitätslogik auf den gesamten Backbone zu verteilen, um den Single Point of Failure zu mindern, der häufig der Hub-and-Spoke-Topologie zugeschrieben wird.

Oft werden Hub-and-Spoke-Topologien mit „traditionellem“ Messaging assoziiert und Busse als der neue Ansatz propagiert. Dieser Vergleich hängt weitgehend davon ab, wie diese Topologien definiert oder implementiert werden. Im Idealfall sollte ein Messaging-Backbone eine Kombination von Topologien unterstützen, damit der optimale Ansatz verwendet werden kann.

Übertragungskonzepte

Sie können verschiedene Konzepte einsetzen, um einen Messaging-Backbone einzurichten, der den Durchsatz optimieren oder die laufende Konfiguration vereinfachen kann (Abbildung 3). Das „Fire-and-Forget“-Prinzip ist ein Übertragungskonzept, bei dem die sendende Anwendung nicht auf eine Antwort oder Bestätigung von der oder den empfangenden Anwendung(en) angewiesen ist. Derartige Nachrichten werden auch als Datagramme bezeichnet. Das „Request-and-Response“-Prinzip ist ein Übertragungskonzept, bei dem die Anwendungen Nachrichten anfordern, in denen angegeben ist, wohin Antworten gesendet werden sollen. Anwendungen, die solche Anforderungen senden, können entweder auf Antworten warten oder die Verarbeitung fortsetzen.



Beispiele für „Publish-and-Subscribe“-Themen

Die Themen können einfache Zeichenfolgen sein:
 „Neuer Abschluss“, „Neuer Kunde“,
 „Kredit abgelehnt“, ...

Die Themen können hierarchisch organisiert sein
 (etwa die Organisation der Vertriebsaktivitäten nach
 Standort und Branche für jeden Kunden):
 „Westen/Köln/Einzelhandel/Allesverkauf/
 Neuer Abschluss“

Größere Abonnements können mit „*“-Platzhaltern
 beschrieben werden (z. B.: immer Nachrichten dann
 abrufen, wenn der Kredit für Einzelhändler abgelehnt
 wird):
 „*/*/Einzelhandel*/Kredit abgelehnt“

(oder: immer Nachrichten dann abrufen, wenn es in
 Köln neue Kunden gibt):
 „Westen/Köln*/Neuer Kunde“

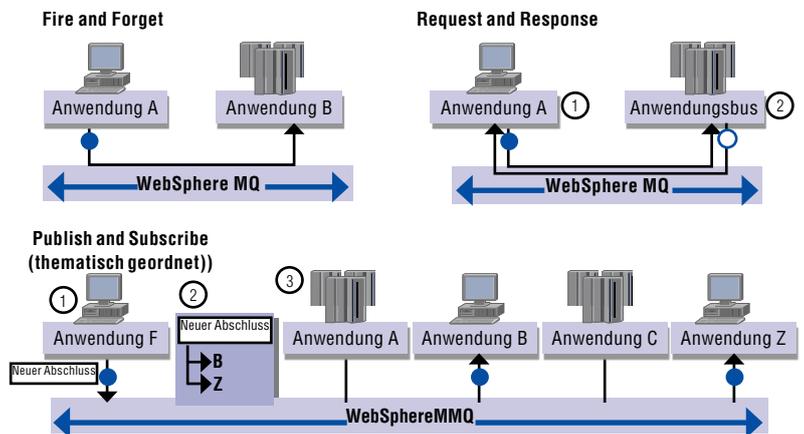


Abbildung 3. Übertragungskonzepte.

Das „Publish-and-Subscribe“-Prinzip ist ein ereignisgesteuertes Übertragungskonzept, bei dem der Messaging-Backbone dynamisch ermitteln kann, wohin die Nachrichten übermittelt werden sollen. Dieser Ansatz befreit Anwendungen und Services von dem Aufwand, stets aktuelle Informationen darüber zu verwalten, welche Anwendungen und Services Nachrichten erhalten müssen und an welchem Ort sie sich gerade befinden. Er kann beim Übergang zu einer SOA von Nutzen sein und ist besonders wertvoll, wenn Sie vorhaben, Anwendungen und Services zu verlagern oder zu ersetzen.

Im „Publish-and-Subscribe“-Modell werden Nachrichten durch Schlüsselwörter oder Stichwörter gekennzeichnet – Zeichenfolgen, die ein Thema für die Nachricht darstellen. Diese Stichwörter können hierarchisch organisiert werden, um eine komplexere Klassifikation zu ermöglichen. Anwendungen und Services definieren die Nachrichten, die für sie bestimmt sind, indem sie mit dem Messaging-Backbone eine Subskription vereinbaren. Dabei kommen Stichwörter, die den Informationsraum beschreiben, sowie optionale Platzhalter zum Einsatz, um umfangreichere Subskriptionen zu definieren.

WebSphere Message Broker, der professionelle ESB von IBM, geht bei dem thematisch geordneten „Publish-and-Subscribe“-Messaging noch einen Schritt weiter: Die Nachrichten können anhand des Nachrichteninhalts und ausgefeilterer Definitionen zum Nachrichtenrouting weitergeleitet werden. Sie können Werkzeuge für die statistische und die kausale Analyse auf Nachrichtenströme anwenden, um Muster zu entdecken, die noch komplexere Ereignisse und Situationen kennzeichnen.

Erfüllung von Unternehmensanforderungen

Der Messaging-Backbone spielt eine entscheidende Rolle in einer IT-Infrastruktur: Er stellt das Bindeglied dar, das Ressourcen und Services miteinander verbindet. Der Backbone muss den hohen Anforderungen der Unternehmen von heute gerecht werden, um mit der Aufgabe betraut zu werden, eine ihrer wertvollsten Ressourcen zu befördern – ihre Geschäftsdaten. WebSphere MQ verfügt über die Erfolgsgeschichte und die Referenzen, welche diese Lösung zu einer hervorragenden Wahl als Messaging-Backbone für Unternehmen aller Größen und Branchen machen.

Zuverlässigkeit

Die Geschäftskosten, Konsequenzen und Geldeinbußen sind enorm, wenn kritische Geschäftsdaten verloren gehen oder wenn die Integrität von Daten und Anwendungen beeinträchtigt wird. Verlorene oder duplizierte Geschäftstransaktionen können die Geschäftsprozesse massiv unterbrechen und eine Kettenreaktion kostspieliger Folgeerscheinungen auslösen, die die Interaktionen der Lieferkette durcheinanderbringen und die Kundenzufriedenheit untergraben könnten. Initiativen zur Einhaltung gesetzlicher Bestimmungen wie das amerikanische Sarbanes-Oxley-Gesetz (SOX) sehen schwere Strafzahlungen für Unternehmen vor, die die Korrektheit ihrer Finanzberichte nicht nachweisen können – ein unerreichbares Ziel, wenn die Gefahr besteht, dass Geschäftstransaktionen verloren gegangen oder dupliziert worden sind. Der Messaging-Backbone eines Unternehmens muss stabil sein und sicherstellen können, dass Geschäftstransaktionen auch bei Störungen der Hardware, der Software und des Netzes erhalten bleiben.

WebSphere MQ garantiert die Übermittlung der über seinen Backbone transportierten Nachrichten und stellt sicher, dass diese Nachrichten während der Übertragung nicht dupliziert werden. Die Lösung organisiert ihre zuverlässige Nachrichtenübermittlung automatisch mithilfe von Quittungen, die den Empfang bestätigen, und sendet Nachrichten bei Bedarf erneut; alle diese Vorgänge sind für Anwendungsprogrammierer unsichtbar. Warteschlangen, die die Nachrichten speichern, gewährleisten die zuverlässige Übermittlung auch dann, wenn das Netz, die Hardware und die empfangenden Anwendungen nicht verfügbar sind.

WebSphere MQ setzt das Zwei-Phasen-Commit-Protokoll als Mechanismus ein, der die Integrität der IT-Systeme bewahrt, wenn im Rahmen von Transaktionen über seinen Backbone Nachrichten übertragen werden. Dabei muss der gesamte Datenaustausch zum Abschluss kommen; eine partielle Ausführung oder eine Störung hinterlässt die Systeme in einem inkonsistenten Zustand. Diese Lösung kann Transaktionen koordinieren, bei denen Nachrichten über ihren ganzen Backbone (nicht nur von einem Server zum anderen) bewegt werden, und sorgt dafür, dass die Transaktionalität durchgängig gewahrt bleibt.



WebSphere MQ: bewährt und zuverlässig

- Mehr als 10.000 Kunden weltweit nutzen WebSphere MQ.
- WebSphere MQ bewegt über 10 Milliarden Nachrichten pro Tag, die Geschäftstransaktionen von mehr als 1 Billionen (1.000.000.000.000.000) US-Dollar entsprechen.
- WebSphere MQ wurde mehrfach ausgezeichnet, wie etwa im Jahr 2004 mit dem renommierten MacRobert Award der Royal Academy of Engineering.
- WebSphere MQ ist der zugrunde liegende Backbone für den professionellen ESB von IBM, WebSphere Message Broker, und wird von allen IBM ESB-Angeboten vollständig unterstützt.
- Mehr als 800 IBM Business Partner weltweit unterstützen WebSphere MQ durch Software, Lösungen und Services.

Hochverfügbarkeit

Heutzutage ist es wichtig, rund um die Uhr erreichbar zu sein. Darum können Ausfälle von Anwendungen, Netzwerken und Systemen schwerwiegende und tiefgreifende Auswirkungen auf die Geschäftsabläufe haben. So muss auch der Messaging-Backbone, der diese Anwendungen und Services miteinander verbindet, den Unternehmensanforderungen im Hinblick auf Hochverfügbarkeit gerecht werden. Hochverfügbarkeit ist der grundlegendste Teil einer Strategie, die Ausfallsicherheit einer IT-Umgebung zu maximieren. Sie setzt voraus, dass diese Umgebung in der Lage ist, den Systembetrieb nach einer Störung schnell und vollständig wiederherzustellen – insbesondere bei unvorhergesehenen Ausfällen.

Hochverfügbarkeit wird dadurch erreicht, dass Single Points of Failure im Backbone beseitigt werden; zu diesem Zweck werden Sicherungs- oder Redundanzsysteme bereitgestellt, die im Fall von Störungen den Betrieb übernehmen können. Sie wird auch dadurch erreicht, dass sämtliche Daten, die transportiert werden, sowie der Status des Messaging-Systems selbst auf dem aktuellen Stand protokolliert werden und im Anschluss an eine Störung zur Wiederherstellung der Daten und des Messaging-Systems verwendet werden können.

Häufig werden Replikationsverfahren als einfacher, direkter Ansatz zur Erhöhung der Verfügbarkeit von Messaging-Systemen gepriesen, der nur Software und keine spezielle Hardware voraussetzt. Diese Ansätze sind jedoch nicht zu empfehlen. Bei asynchroner Replikation kann es vorkommen, dass Nachrichten dupliziert werden oder verloren gehen; synchrone Replikation setzt die Echtzeitreplikation aller Nachrichten voraus, um diese Schwäche zu umgehen, und kann zu erheblichen Leistungseinbußen führen.

Zur Vermeidung dieser Probleme setzt WebSphere MQ auf ein persistentes Übertragungskonzept, das die Nachrichten zusammen mit dem Status der Warteschlangenmanager synchron zur Nachrichtenübertragung auf der Festplatte protokolliert, um nach einer Störung die Nachrichten wiederherstellen und wieder für einen konsistenten Zustand sorgen zu können. Es ist äußerst wichtig, dass die Nachrichten genau zu der Zeit, zu der sie versendet werden, synchron protokolliert werden; andernfalls besteht die Gefahr, dass Sie nach einer Störung nicht mehr in der Lage sind, einen kohärenten Zustand herzustellen, bevor oder nachdem die Nachricht gesendet wird. WebSphere MQ kann Maßnahmen ergreifen, um für Nachrichten, die gerade unterwegs waren, ein Rollback, und für abgeschlossene Übertragungen ein Commit durchzuführen. So bleibt die Integrität der Nachrichten und der betreffenden Anwendungen gewahrt.

WebSphere MQ verwendet plattformspezifische Hochverfügbarkeitsfunktionen, etwa High Availability Cluster Multi-Processing (IBM HACMP) auf der Plattform IBM AIX und Automatic Restart Manager (ARM) auf der Plattform IBM z/OS. Auf IBM System z, das unter dem Betriebssystem z/OS ausgeführt wird, stellt WebSphere MQ gemeinsam genutzte Warteschlangen bereit, auf die von verschiedenen Warteschlangenmanagern zugegriffen werden kann; sollte einer von diesen ausfallen, kann ein anderer Warteschlangenmanager automatisch auf die gemeinsam genutzten Nachrichten zugreifen.

Clustering

Die Clustering-Funktionen von WebSphere MQ sorgen dafür, dass Nachrichten um Teile des Messaging-Backbone, die durch einen Netzausfall oder aus einem anderen Grund ohne die Anwendungen nicht verfügbar sind, herumgeleitet werden können.

Clustering kann den Aufwand für die Administration des Messaging-Backbone verringern. Viele Warteschlangenmanager von WebSphere MQ können gemeinsam in einer Gruppe verwaltet werden. Wenn Warteschlangen in einem Cluster mit demselben Namen konfiguriert werden, kann WebSphere MQ feststellen, auf welche Weise die Messaging-Workload zwischen ihnen aufgeteilt werden sollte. Dieses Clustering kann auf dem integrierten Lastausgleichsalgorithmus oder auf Ihrem eigenen Algorithmus basieren. Es ist auch möglich, einzelne Warteschlangenmanager innerhalb eines Clusters anzusprechen, indem diese eindeutige Namen erhalten, und sie zu referenzieren, wenn Nachrichten zu ihrem Cluster übertragen werden. Mittels des Lastausgleichs versetzt das Clustering Sie in die Lage, den Durchsatz eines Messaging-Backbone zu verbessern und bei Bedarf weitere Messaging-Engines hinzuzufügen.

Sicherheit

Im heutigen Wirtschaftsklima ist die Sicherheit der wertvollen und vertraulichen Informationen und IT-Systeme ein zentrales Anliegen. Die IT-Systeme so sicher wie möglich zu machen, ist eine ganzheitliche Herausforderung, die Bereichen wie Verschlüsselung, Authentifizierung, Autorisierung, Unbestreitbarkeit und Vertraulichkeit Rechnung tragen muss.

WebSphere MQ verfügt über integrierte Sicherheitsfunktionen zum Schutz der Daten, die über seinen Messaging-Backbone bewegt werden. Die Lösung unterstützt den Standard Secure Sockets Layer (SSL) zur strikten Authentifizierung von Nachrichtenkanälen vor dem Austausch von Nachrichten, um böswillige Angriffe auf den Backbone zu verhindern. SSL ermöglicht auch die Verschlüsselung von Massendaten, vergleichbar mit virtuellen privaten Netzen (VPNs).

IBM WebSphere MQ Extended Security Edition erweitert den Messaging-Backbone um zusätzliche Sicherheitsfunktionen, die auf IBM Tivoli-Technologien zurückgreifen, welche eine individuelle, einzelne Nachrichten betreffende Sicherheitskapselung ermöglichen. So wird der Schutz über die Nachrichtenkanäle hinaus auf die Anwendungsebene ausgedehnt. Die Lösung stellt differenzierte Protokolleinträge bis zur Ebene der einzelnen Nachrichten bereit und kann an ein vorhandenes WebSphere MQ-Netz angepasst werden, ohne dieses modifizieren zu müssen. Die Sicherheit wird über Richtlinien konfiguriert, die remote über eine Web-Browser-Schnittstelle verwaltet werden können; Sie brauchen also nicht für jede Anwendung sicherheitsspezifischen Code zu schreiben.

Skalierbarkeit und Leistung

Da die Geschäftsanforderungen immer weiter zunehmen und immer mehr Anwendungen und Services zur Kommunikation Messaging einsetzen, muss der Messaging-Backbone entsprechend skaliert werden können, damit er das steigende Volumen des Datenverkehrs unterstützt und Sie das Potenzial Ihrer IT-Infrastruktur optimal nutzen können.

WebSphere MQ nutzt die Vorteile von Multiprozessor- und Mehrkernmaschinen. Demzufolge können Sie den Messaging-Backbone mithilfe von Parallelverarbeitung skalieren, um die Nachrichtenübertragung zu beschleunigen. Dieser Ansatz steigert erheblich den Durchsatz und die Skalierbarkeit des Punkt-zu-Punkt-Messaging gegenüber Einzelthread-Alternativen. Eine einzige Instanz von WebSphere MQ kann auf mehreren Prozessoren und Kernen ausgeführt werden. Es ist also nicht erforderlich, das Routing oder den Lastausgleich zwischen mehreren Instanzen eines Servers zu konfigurieren. Die Lösung benötigt für die Ausführung auf mehreren Prozessoren oder Kernen praktisch keine Optimierung oder Konfiguration und ist nicht auf mehrfache Nachrichtenprotokolle angewiesen. Wenn Sie sicherstellen möchten, dass Ihr Messaging-Backbone mit Ihrem Unternehmen wachsen kann, sollten Sie auf Leistungsvergleichen mit mehreren Prozessoren bestehen. Ein weiteres nützliches Verfahren zur Skalierung eines Messaging-Backbone ist das Clustering, das weiter oben in diesem White Paper beschrieben wurde.

Bei einem Vergleich des Durchsatzes von Messaging-Servern muss die Servicequalität bei der Nachrichtenübermittlung übereinstimmen – etwa das Vorliegen oder Fehlen von Persistenz. Leistungsberichte für WebSphere MQ liefern speziell für jede Plattform Informationen zur Durchsatzanalyse, der Kapazitätsplanung und der Optimierung. Sie werden als IBM SupportPacs veröffentlicht unter ibm.com/webspheremq/support

Konfiguration

Die IT-Infrastruktur Ihres Unternehmens ist so einzigartig wie ein Fingerabdruck. Dabei ist sie ständigen Änderungen unterworfen, um von neuen Technologien und IT-Ressourcen zu profitieren und auf geschäftliche Notwendigkeiten und Chancen zu reagieren. Es ist von entscheidender Bedeutung, dass der Messaging-Backbone diese Änderungen unterstützt, statt sie zu blockieren. Um die Zeit, das Risiko und die Kosten der Konfiguration zu verringern, ist es wichtig, dass der gesamte Messaging-Backbone zentral konfiguriert und verwaltet werden kann, obgleich es sich um eine ausgedehnte, verteilte Infrastruktur handelt.

Bei WebSphere MQ kann der gesamte Messaging-Backbone über eine einzige Konsole, den WebSphere MQ Explorer (Abbildung 4), entfernt konfiguriert werden. Dieses neue Konfigurationswerkzeug, das in Version 6.0 von WebSphere MQ eingeführt wurde, basiert auf der Open-Source-Technologie der Eclipse Workbench. Das Eclipse-Framework ist für alle IBM Softwareprodukte einheitlich. Somit kann WebSphere MQ Explorer mit den Werkzeugen anderer Produkte, etwa WebSphere Message Broker, zu einer einzigen, integrierten Konsole kombiniert werden. Das grafisch orientierte Werkzeug dient der Untersuchung und Konfiguration aller Objekte und Ressourcen von WebSphere MQ, darunter Java Message Service (JMS) sowie „Publish and Subscribe“. WebSphere MQ Explorer basiert auf Eclipse-Technologie und ist daher hochgradig anpassbar und vollständig erweiterbar. Sie können neue Werkzeuge als Plug-ins in WebSphere MQ Explorer aufnehmen und so neue Funktionen hinzufügen, die in die Konsole integriert sind. Die mit WebSphere MQ Explorer gelieferte Dokumentation erläutert die Schnittstellen für Plug-ins und enthält Beispiele für deren Entwicklung. IBM Business Partner und Benutzer können so gemeinsam mit IBM an der Erweiterung des Leistungsspektrums arbeiten.

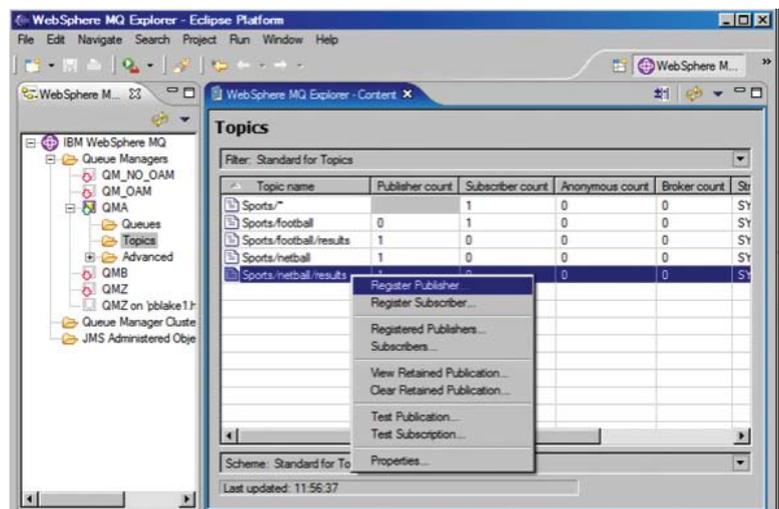


Abbildung 4. Mit WebSphere MQ Explorer können Sie den gesamten Messaging-Backbone remote untersuchen, testen und konfigurieren.

WebSphere MQ Explorer kann auf x86-Systemen mit Microsoft® Windows® und mit Linux® ausgeführt werden. Das Werkzeug braucht nicht zusammen mit einem WebSphere MQ-Client oder -Server implementiert zu werden; Sie können so viele Kopien installieren wie erforderlich. Außerdem sind Sie für die Untersuchung oder die Konfiguration nicht auf die direkte Verbindung mit einem Warteschlangenmanager angewiesen. Sie können die Warteschlangenmanager entfernt über zwischenliegende Warteschlangenmanager konfigurieren. WebSphere MQ Explorer kann remote mit Warteschlangenmanagern auf jeder unterstützten Plattform Verbindung aufnehmen; so kann Ihr gesamter Messaging-Backbone von der Konsole aus angezeigt, beobachtet und geändert werden. Sie können sogar WebSphere MQ for z/OS auf der Plattform System z über Fernzugriff konfigurieren. Voraussetzung dafür ist, dass Version 6.0 (oder neuer) ausgeführt wird, weil diese nun die administrativen Nachrichten im Programmable Command Format (PCF) unterstützt, die von WebSphere MQ Explorer verwendet werden.

WebSphere MQ unterstützt mehr als 80 Plattform-konfigurationen und deckt nahezu jedes kommerzielle IT-System ab:

- AIX
- IBM System i (IBM eServer iSeries und IBM OS/400)
- IBM z/OS (IBM eServer zSeries und IBM OS/390)
- HP-UX
- HP NonStop Server und OpenVMS
- Linux on Intel®
- Linux on System z
- Betriebsumgebung Sun Solaris
- Microsoft Windows XP und 2000

Aktuelle Details zur Unterstützung finden Sie unter:
ibm.com/webspheremq/requirements

Sie können die Ansichten des Messaging-Backbone individuell anpassen, beispielsweise indem Sie für die Anzeige von Warteschlangen oder anderen Ressourcen Filter einsetzen, die bestimmte Kriterien wie die Anzahl der Nachrichten in einer Warteschlange oder deren Namen zugrunde legen. Sie können die Aktualisierungsraten dieser gefilterten Ansichten anpassen, um sie auf der Ebene des Systems oder des Warteschlangenmanagers zu aktualisieren. Sie können auch Attribute vergleichen, etwa um festzustellen, ob zwei Warteschlangen die gleichen Merkmale aufweisen.

Um unautorisierten Änderungen vorzubeugen, setzt WebSphere MQ Explorer Sicherheitsfunktionen des SSL-Standards ein. Grafisch orientierte Werkzeuge dienen der Verwaltung von Berechtigung und Zugriff auf der Basis des Objektberechtigungsmanagers (Object Authority Manager, OAM), um die Corporate-Governance-Kontrolle zu vereinfachen. Beispielsweise können Sie mit WebSphere MQ Explorer auf einen Blick alle Benutzer oder Gruppen anzeigen, die über Berechtigungen für bestimmte Warteschlangen und Objekte verfügen. Abbildung 5 illustriert diese Funktion.

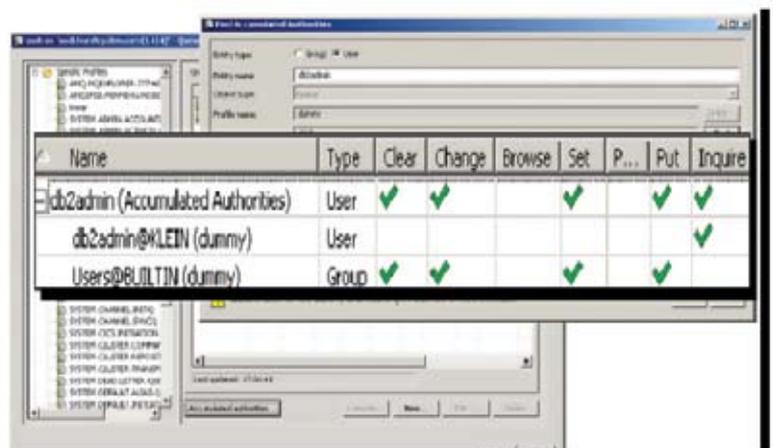


Abbildung 5. Mit WebSphere MQ Explorer können Sie den Zugriff auf Ihren Messaging-Backbone kontrollieren.

Neue Werkzeuge zur Problemdiagnose sind vor kurzem hinzugefügt worden, mit denen Sie Ihren Messaging-Backbone testen können, um Fehler und potenzielle Probleme in der Konfiguration aufzuspüren. Auf einen Mausklick sucht das Werkzeug nach Problemen und gibt Ratschläge in Bezug auf Lösungen und Verbesserungen. Die Testfolge kann um benutzerdefinierte Überprüfungen erweitert werden.



Eine einfache Schnittstelle

MQI stellt konsistent für alle unterstützten Plattformen 13 einfache Befehle bereit:

- MQCONN verbindet mit dem als Parameter genannten Warteschlangenmanager.
- MQCONNX verbindet über Direktaufverbindungen für vertrauenswürdige Anwendungen.
- MQDISC trennt die Verbindung zum Warteschlangenmanager.
- MQOPEN öffnet eine Nachrichtenwarteschlange im Warteschlangenmanager.
- MQCLOSE schließt die Nachrichtenwarteschlange.
- MQPUT reiht eine Nachricht in die Warteschlange ein.
- MQPUT1 ist äquivalent zu der Folge MQOPEN, MQPUT, MQCLOSE.
- MQGET ruft eine Nachricht aus der Warteschlange ab.
- MQINQ fragt die Eigenschaften einer Warteschlange ab.
- MQSET legt die Eigenschaften einer Warteschlange fest.
- MQBEGIN beginnt eine Transaktion oder UOW.
- MQBACK macht eine noch nicht beendete Transaktion rückgängig.
- MQCMIT schreibt eine Transaktion fest und beendet die UOW.

Nahezu grenzenlose Konnektivität

Ein Messaging-Backbone ist nur so wertvoll wie das Spektrum der Anwendungen und Services, mit denen er sich verbinden kann. Selbst wenn bestimmte Plattformen und Umgebungen zurzeit nicht Ihrer IT-Domäne angehören, können Sie sich das Risiko nicht leisten, dass einzelne Anwendungen infolge einer abteilungsübergreifenden Initiative, einer Fusion oder einer Übernahme in Isolation geraten oder von der Nutzung einer neuen Technologie oder Plattform abgeschnitten werden. WebSphere MQ ist für die Verbindung mit nahezu jedem kommerziellen IT-System konzipiert, von den neuesten Technologien bis zu denjenigen Kernsystemen, von denen Ihr Unternehmen abhängt – selbst wenn es sich um weniger gebräuchliche Plattformen handelt.

WebSphere MQ stellt eine konsistente Funktionalität für das gesamte Spektrum der unterstützten Plattformen bereit. WebSphere MQ for z/OS wurde nativ für z/OS erstellt. Das Design ist dennoch mit der verteilten Version konsistent. Daher kann die Lösung die speziellen Möglichkeiten der z/OS-Umgebung für einzigartige Funktionen nutzen, die das Potenzial des Messaging-Backbone bedeutend erweitern. WebSphere MQ for z/OS ist nicht einfach eine Anpassung des verteilten Produkts, sondern eine spezielle Codebasis, die von Grund auf dafür konzipiert wurde, die Plattform z/OS optimal zu nutzen und sich nahtlos in die Großrechnerumgebung einzufügen.

WebSphere MQ for z/OS profitiert von der IBM Technologie Parallel Sysplex, die die Verfügbarkeit, die Kapazität und den Durchsatz für persistente und nicht persistente Nachrichten erhöht, da mehrere Warteschlangenmanager auf dieselbe Warteschlange zugreifen können. Darüber hinaus bietet WebSphere MQ for z/OS einen Lastausgleich auf der Basis eines Pull-Modells, um auf der Plattform z/OS Messaging mit außerordentlich hoher Verfügbarkeit zu ermöglichen.

Außerdem bietet WebSphere MQ for z/OS nahtlose Integration mit IBM CICS und IBM IMS mithilfe der IBM MQSeries-CICS Dynamic Program Link (DPL) Bridge, der IBM MQSeries-CICS 3270 Bridge und der IBM MQSeries-IMS Bridge.

Schnittstellen und Standards

Ein Messaging-Backbone muss die in Ihrem Unternehmen vorhandenen Ressourcen anbinden, Ihnen die optimale Nutzung des verfügbaren Know-how ermöglichen und Ihre zukünftigen Entscheidungen unterstützen. WebSphere MQ ist mit dem Ziel entwickelt worden, eine Vielzahl von Wegen für die Verbindung mit seinem Messaging-Backbone bereitzustellen.

MQI

Das Message Queue Interface (MQI) ist die ursprüngliche Programmierschnittstelle für WebSphere MQ. Es verfügt über einen einfachen, überschaubaren Bestand von Verben, die für alle Plattformen konsistent sind. Je nach der Betriebssystemumgebung können Sie für die MQI-Programmierung C, C++, Java, PL/I, COBOL, Visual Basic, ActiveX/COM, Assembler, Report Program Generator (RPG) oder Typed Assembly Language (TAL) einsetzen. Unterstützung für weitere Sprachen, beispielsweise Perl, ist als Download verfügbar.

MQI enthält Strukturen (Gruppen von Feldern), mit denen Sie Eingaben für Aufrufe und Ausgaben von Aufrufen bereitstellen können. Ferner enthält die Schnittstelle einen umfangreichen Satz benannter Konstanten, die die Angabe von Optionen in den Parametern der Funktionsaufrufe vereinfachen. Datendefinitionsdateien enthalten die Definitionen der Aufrufe, Strukturen und benannten Konstanten für jede unterstützte Programmiersprache. Die MQI-Aufrufe sind mit Standardwerten vorbelegt.

JMS

Der Java Message Service (JMS) ist eine standardisierte Programmierschnittstelle für Messaging auf der Basis von J2EE. Er hat dieselbe Funktion wie das MQI, ist jedoch standardisiert und wird bei Java-Entwicklern immer häufiger eingesetzt. Mit der J2EE-Technologie kompatible Anwendungsserver, zum Beispiel IBM WebSphere Application Server, besitzen eine JMS-Schnittstelle und können ebenfalls eine Implementierung von Messaging-Services bieten. JMS wird auch von vielen eigenständigen Messaging-Lösungen unterstützt, darunter WebSphere MQ. J2EE-Anwendungsserver können WebSphere MQ als JMS-Provider einsetzen, auch wenn sie bereits einen solchen enthalten. Die Annahme, JMS-kompatible Produkte müssten vollständig in Java entwickelt werden, trifft nicht zu.

WebSphere MQ unterstützt die neueste Version des JMS-Standards, Version 1.1. So wie das oben beschriebene MQI ist auch JMS eine Schnittstelle für die Programmierung. Da es sich bei JMS um einen Industriestandard handelt, können entsprechend programmierte Anwendungen zwischen Messaging-Produkten mit JMS-Unterstützung ohne Modifikationen portiert werden. Entwickler brauchen daher nur einen Befehlssatz zu erlernen, um eine Vielzahl externer Produkte verwenden zu können. In der Praxis stellen viele Hersteller Erweiterungen der JMS-Anwendungsprogrammierschnittstelle bereit. JMS standardisiert nur die Schnittstelle, nicht die für die zugrunde liegende Datenbereitstellung verwendete Technologie. Darum ist das Verbindungsprotokoll JMS-kompatibler Produkte proprietär und herstellerspezifisch. JMS-kompatible Produkte können also nicht miteinander interagieren.

Zwar sorgt der JMS-Standard nicht für Interoperabilität zwischen Messaging-Providern, doch IBM stellt die Funktionalität bereit, die für die Zusammenführung von JMS-Implementierungen und die Schaffung eines kombinierten Messaging-Backbone benötigt wird.

WebSphere MQ und WebSphere Application Server sind so konzipiert, dass ihre JMS-Implementierungen interagieren. WebSphere MQ und WebSphere Application Server können Nachrichten austauschen, um einen kombinierten Messaging-Backbone zu bilden, bei dem die Transaktionalität zwischen diesen erhalten bleibt und „Publish-and-Subscribe“-Definitionen gemeinsam genutzt werden können. Somit kann eine Java-Anwendung, die in WebSphere Application Server gehostet wird, über JMS mit anderen Anwendungen kommunizieren, die mit WebSphere MQ verbunden sind, entweder über JMS oder über dessen MQI. Auch IBM Produkte wie IBM WebSphere ESB oder WebSphere Process Server, die auf WebSphere Application Server basieren, können sich auf diese Weise mit WebSphere MQ verbinden. Sie können aber auch native MQI-Aufrufe zur Verbindung mit WebSphere MQ verwenden.

Zusätzlich stellt WebSphere Message Broker, der professionelle Enterprise Service Bus, einzigartige Mediationen bereit, die alle Messaging-Lösungen, die mit Version 1.1 der JMS-Technologie kompatibel sind, zu einem zentralen Messaging-Backbone konsolidieren können.

Mehrsprachiger Nachrichtenservice

Die im vorherigen Abschnitt beschriebene standardisierte Schnittstelle JMS setzt voraus, dass die Programmierer über Java-Kenntnisse verfügen. Auch wenn Java heutzutage weit verbreitet ist, so ist sie nicht immer die bevorzugte Programmiersprache. Sie eignet sich auch nicht optimal für die Verbindung vieler Nicht-Java-Umgebungen. IBM hat eine Anwendungsprogrammierschnittstelle entwickelt, die mit JMS konsistent ist, aber in weiteren Sprachen implementiert worden ist. Während JMS speziell auf Java ausgerichtet ist, wird das IBM API in mehreren Sprachen bereitgestellt und daher als mehrsprachiger Nachrichtenservice bezeichnet. Häufig wird es als XMS abgekürzt, wobei X als Platzhalter für einen beliebigen Nachrichtenservice steht.

WebSphere MQ-Clients verfügen über die Schnittstelle XMS. Zu den unterstützten Sprachen gehören C und C++ sowie einen vollständig verwalteten Client für Microsoft .NET-Umgebungen, der mit jeder .NET-Sprache wie etwa C# verwendet werden kann.

Messaging für Web-Services

Web-Service-Standards definieren Mechanismen für die Klassifizierung, die externe Bereitstellung, das Auffinden und das Aufrufen von Services. Diese Services reichen von Abläufen innerhalb Ihrer vorhandenen Anwendungen, die für Ihr gesamtes Unternehmen verfügbar gemacht werden können, bis zu neuen J2EE-Komponenten innerhalb eines Anwendungsservers. Mit der Weiterentwicklung der Web-Service-Standards kommen Konzepte wie Web-Services-Reliable Messaging (WSRM) und Web-Services-Notification (WSN) auf.

Web Services Description Language (WSDL) definiert die Standards für Web-Services. Universal Description, Discovery and Integration (UDDI) bietet eine Verzeichnis- und Suchfunktionalität. SOAP ist das Datenformat, das bei der Kommunikation mit einem Web-Service zum Einsatz kommt. Einen vorgegebenen Standard für die Datenübertragung gibt es jedoch nicht. Sie können Ihre SOAP-Nachrichten mit beliebigen Transportmethoden senden. Ein verbreitetes Verfahren besteht darin, SOAP-Nachrichten über HTTP oder HTTP Secure (HTTPS) zu senden.

Wenn Ihre SOAP-formatierten Daten mit der Servicequalität und den Vorteilen eines Messaging-Backbone übermittelt werden sollen, benötigen Sie für die zugrunde liegende Datenübertragung eine Messaging-Lösung. WebSphere MQ kann SOAP-Nachrichten innerhalb einer Web-Service-Implementierung – in der Regel innerhalb der J2EE-Umgebung – senden und empfangen, was häufig als „SOAP over JMS“ bezeichnet wird. Durch diesen Ansatz können Web-Services von den Vorteilen eines Messaging-Backbone profitieren.

Zuverlässige Übertragung von Dateien

Nach Schätzung vieler Analysten werden nicht weniger als 80 Prozent der Geschäftsinformationen in – und zwischen – Unternehmen über das File Transfer Protocol (FTP) weitergegeben. Sie werden vielleicht erstaunt sein, in welchem Umfang Ihre wertvollen Geschäftsinformationen Risiken ausgesetzt sind und verloren gehen oder unbemerkt beschädigt werden können. Anwendungen und Daten können verfälscht werden, Geschäftsprozesse können ins Stocken geraten oder ihre Effektivität verlieren, Finanzberichte können falsch sein. Und vielleicht das Schlimmste ist, dass Sie es möglicherweise nicht einmal merken, wenn dies geschehen ist, weil die Übertragungen alles andere als transparent sind.

FTP-Technologien breiten sich wegen ihrer verlockenden Einfachheit rasch aus. Es ist eine Fülle kostenloser FTP-Pakete verfügbar, das Konzept der Dateiübertragung ist sehr intuitiv und der Einstieg setzt normalerweise nur grundlegende Kenntnisse voraus. Möglicherweise sind weder die Einstiegskosten noch die laufenden Kosten für die Implementierung und Wartung dieser Lösung jemals geprüft oder analysiert worden. Je mehr allerdings die Abhängigkeit von diesen Lösungen zunimmt, desto mehr Zeit und Know-how investieren die IT-Abteilungen in die Entwicklung zusätzlicher Funktionen, um die Zuverlässigkeit und Sicherheit dieser Lösungen zu testen und deren Mängel zu beseitigen. Wenn immer mehr Sender und Empfänger von Dateien hinzukommen, nimmt die Komplexität der Umgebung rasch zu. Das Unternehmen stellt fest, dass es nun vollauf mit der Wartung und Korrektur dieser Lösungen beschäftigt ist und diese Ressourcen nicht in andere IT-Projekte investieren kann, etwa um die Flexibilität oder die Serviceorientierung zu verbessern.

Eine solche IT-Abteilung ist unwissentlich in den Middlewaredbereich eingestiegen. Selbst Unternehmen, die große Anstrengungen zur Einführung von Integrationsmiddleware unternommen haben, sind möglicherweise immer noch stark auf Dateiübertragungen angewiesen, auch wenn ihnen dies nicht bewusst ist. Nicht nur, dass diese Abhängigkeit die Flexibilität Ihres Unternehmens verringert und dessen Reaktionsfähigkeit beeinträchtigt, sie kann auch zahllose Arbeitsstunden für die Fehlerdiagnose oder die Überarbeitung vermeintlich einfacher Lösungen beanspruchen. Werden Ihre wertvollen IT-Ressourcen auf diese Weise wirklich optimal genutzt? Benötigt wird ein Weg, Dateien in Ihren Messaging-Backbone einzubeziehen.

IBM vertreibt PM4Data von dem IBM Business Partner MetaStorm Incorporated. PM4Data setzt WebSphere MQ als zugrunde liegende Transportmethode ein und kann auf den meisten Plattformen ausgeführt werden, die von WebSphere MQ unterstützt werden, einschließlich einer speziellen Edition von PM4Data für z/OS. Die Kombination von PM4Data und WebSphere MQ bewirkt, dass Dateiübertragungen durchgängig, sogar über Fernzugriff, verwaltet und mit durchgängiger Transparenz überprüft werden können. Die Übertragungen werden durch WebSphere MQ garantiert und nutzen die von WebSphere MQ bereitgestellten, auf Unternehmen abgestimmten Sicherheitsfunktionen, darunter SSL. PM4Data kann umfangreiche Dateien komprimieren, um die Netzübertragung zu optimieren. Die Größe der Dateien, die über WebSphere MQ bewegt werden können, ist theoretisch grenzenlos. Unterschiede zwischen Plattformen und Dateitypen werden automatisch während der Übertragungen berücksichtigt.

Mit MetaStorm PM4Data können Sie alle Aspekte Ihrer Dateiübertragungen verwalten, etwa die Anzeige und Protokollierung der Startzeit und Endzeit von Dateiübertragungen oder die Anzeige der gerade laufenden Übertragungen. Mithilfe von PM4Data können Sie ermitteln, ob Übertragungen aus irgendeinem Grund fehlgeschlagen sind, und die Geschwindigkeit der Dateiübertragungen oder den zugrunde liegenden Durchsatz von WebSphere MQ überwachen. Mit PM4Data können Sie Dateiübertragungen während des Betriebs erneut starten oder umleiten; daher ist es in der Regel nicht notwendig, die Datei noch einmal auf den gesamten Weg zu schicken. Die Datei wird lediglich von dem Ort umgeleitet, an dem sie hängen geblieben ist.

Falls erforderlich, kann PM4Data mehrere Dateien automatisch innerhalb einer einzigen WebSphere MQ-Nachricht übertragen, was die Auslastung Ihres Netzes optimiert. Die anpassbaren Exitpunkte und eine leistungsfähige Scripting-Sprache bieten vielfältige Erweiterungsmöglichkeiten.

Doch es geht nicht nur um die verwaltete, zuverlässige Übertragung von Dateien. Unternehmen erkennen zunehmend den Nutzen der Verarbeitung, Umwandlung oder Aufbereitung ihrer Dateien. Mit WebSphere Message Broker File Extender können Sie Dateien direkt an WebSphere Message Broker übergeben, wo sie mittels leistungsfähiger Vermittlungsfunktionen konvertiert, neu formatiert und aufbereitet werden können. Die Lösung ist vollständig in die Palette des WebSphere Message Broker Toolkit integriert. Diese neue Erweiterung steigert somit die Produktivität neuer und bisheriger Benutzer von WebSphere Message Broker.

Mit WebSphere Message Broker File Extender können Sie WebSphere Message Broker sowohl zu einem Dateibroker als auch zu einem Nachrichtenbroker erweitern. Mithilfe dieser Funktionalität können Sie die in Dateien enthaltenen Daten in demselben Maß nutzen und wiederverwenden, wie dies zurzeit bei denjenigen Daten der Fall ist, die in Nachrichtenform in Ihrem Unternehmen versendet werden. Sie können Broker-Services wie Konvertierung, Aufbereitung, Protokollierung und Weiterleitung auf Dateiinhalte anwenden und Nachrichten nahtlos in Dateien und Dateisätze (und umgekehrt) konvertieren. Dazu brauchen Sie nur die entsprechenden E/A-Knoten im jeweiligen Nachrichtenfluss zu verbinden und zu konfigurieren.

Mobile Einheiten und Endgeräte

IBM WebSphere MQ Everyplace umfasst die zentralen Funktionen eines Messaging-Backbone für Geräte mit begrenzten Ressourcen und ist für instabile Netze optimiert. Das Produkt unterstützt den Standard JMS und arbeitet über eine integrierte Brückenfunktion mit WebSphere MQ zusammen.

Sensoren und Aktuatoren

WebSphere Message Broker, der professionelle Enterprise Service Bus von IBM, stellt eine Transportmethode bereit, die den Messaging-Backbone in Hinsicht auf sehr kleine (und eingebettete) Einheiten wie Sensoren, Ventile oder Messgeräte erweitern kann. Diese Transportmethode heißt MQ Telemetry Transport (MQTT).

Basis für Ihren ESB

Ein Messaging-Backbone kann Ihnen – selbst wenn er nur wenige Anwendungen verbindet – erhebliche Vorteile verschaffen, wie etwa zur Senkung der IT-Kosten beitragen oder die Flexibilität der Infrastruktur und des Unternehmens steigern. Ein Messaging-Backbone dient jedoch auch als Basis für weitere Integrationsfunktionen, er bildet die Grundlage für durchgängige Gesamtintegration und Serviceorientierung. Ein Messaging-Backbone fungiert als Sprungbrett für den Start von IT-Projekten, die auf komplexere Integrationsaufgaben und die Erzielung noch wichtigerer Resultate ausgerichtet sind.

Ein ESB bildet eine Abstraktionsebene oberhalb des Messaging-Backbone, wodurch dieser ohne Programmieraufwand um umfassendere Integrationsfunktionen erweitert wird. Während Messaging-Backbones direkt mit Anwendungen verbunden werden, dient ein ESB in erster Linie zur flexiblen Verbindung von Services. Services sind interoperable Codeeinheiten, die von der zugrunde liegenden Plattform und Programmiersprache unabhängig sind. Ein ESB erweitert einen Messaging-Backbone um folgende Funktionen:

- *Dynamische Abstimmung und Weiterleitung von Nachrichten zwischen Services*
- *Konvertierung von Transportprotokollen oder Anpassung von Transport-Service-Levels zwischen Services*
- *Umwandlung von Nachrichtenformaten, Aufbereitung oder Änderung von Nachrichteninhalten auf dem Weg zwischen Services*
- *Verteilung von Geschäftsereignissen zwischen Services*

WebSphere MQ ist ein Messaging-Backbone, der von allen IBM ESB-Angeboten verwendet werden kann – IBM WebSphere ESB, IBM WebSphere Message Broker und IBM WebSphere DataPower XI50 Integration Appliance. WebSphere ESB ist auf die Verwendung von Web-Service-Standards ausgerichtet und stellt Mediationsservices für XML-formatierte Daten bereit. WebSphere Message Broker ist ein professioneller ESB, dessen erweiterte Funktionalität ein breites Spektrum standardisierter und vom Standard abweichender Situationen bewältigt und der Daten in beliebigen Nachrichtenformaten vermittelt.

Auf der anderen Seite bildet ein ESB die Grundlage für das Geschäftsprozessmanagement, indem er die Automatisierung und effektivere Verwaltung von Geschäftsprozessen ermöglicht. Auf seiner Basis können Sie fundierter beurteilen, wie die Ausführung der Geschäftsaktivitäten weiterentwickelt und geändert werden soll, und zwar anhand einer quantitativen Analyse auf Geschäftsebene sowie anhand vorhandener Prozesse, die mithilfe von Key Performance Indicators (KPIs) und weiteren Geschäfts- oder Finanzkennzahlen gemessen und optimiert werden.

Zusammenfassung

WebSphere MQ ist ein leistungsfähiger, innovativer Messaging-Backbone, der einen Einstieg in SOA ermöglicht. Auch wenn WebSphere MQ ein ausgereiftes Produkt mit hoher Reputation und Marktpräsenz darstellt, wird es kontinuierlich weiterentwickelt. In der heutigen Zeit, in der Web-Services eine zunehmende Dynamik entwickeln, kommt IBM bei der Definition einschlägiger Standards eine Vorreiterrolle zu. WebSphere MQ nimmt bei der Entwicklung dieser aufregenden Technologien eine Schlüsselrolle ein.

Auf welche Weise Sie SOA auch einsetzen möchten, um die Flexibilität Ihrer IT zu erhöhen – ein Messaging-Backbone bietet Ihrem Unternehmen, unabhängig von seiner Größe, attraktive Vorteile. Falls Ihr Unternehmen einen Enterprise Service Bus implementieren möchte, stellt WebSphere MQ eine Konnektivitätsebene als Grundlage dafür bereit. Auch kleine oder mittlere Unternehmen, die ihre Anwendungen unverzüglich vernetzen müssen, können durch die Implementierung einer Messaging-Lösung erhebliche Vorteile erzielen. Welcher Branche Sie auch angehören, Sie können sich darauf verlassen, dass IBM auch in Zukunft in WebSphere MQ investieren wird. IBM wird weiterhin neue Funktionen hinzufügen und neue Technologien unterstützen und zugleich die vorhandene Funktionalität verbessern.

Weitere Informationen

IBM WebSphere MQ ist mehr als ein Softwareprodukt mit langer Erfolgsgeschichte. Diese Lösung wird auch zukünftig die grundlegenden Konnektivitätsprobleme lösen, mit denen die IT-Branche heute konfrontiert ist.

Weitere Informationen zum Thema Messaging-Backbone und zu IBM WebSphere MQ erhalten Sie bei Ihrem IBM Ansprechpartner oder IBM Business Partner oder auf folgender Website:

ibm.com/webspheremq

Wenn Sie an der Global WebSphere Community teilnehmen möchten, besuchen Sie:

www.websphere.org



IBM Deutschland GmbH
Pascalstrasse 100
70569 Stuttgart
ibm.com/de

IBM Österreich
Obere Donaustrasse 95
1020 Wien
ibm.com/at

IBM Schweiz
Vulkanstrasse 106
8010 Zürich
ibm.com/ch

Die IBM Homepage finden Sie unter:

ibm.com

IBM, das IBM Logo, ibm.com, AIX, CICS, DataPower, eServer, Everyplace, HACMP, IMS, iSeries, MQSeries, OS/390, OS/400, Parallel Sysplex, SupportPac, System i, System z, Tivoli, WebSphere, z/OS und zSeries sind Marken der IBM Corporation in den USA und/oder anderen Ländern.

Intel, das Intel-Logo, Intel Inside, das Intel Inside-Logo, Intel Centrino, das Intel Centrino-Logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium und Pentium sind Marken oder eingetragene Marken der Intel Corporation oder deren Tochtergesellschaften in den USA oder anderen Ländern.

Java und alle auf Java basierenden Marken sind Marken von Sun Microsystems, Inc. in den USA und/oder anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Microsoft und Windows sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und/oder anderen Ländern.

Weitere Unternehmens-, Produkt- oder Servicenamen können Marken anderer Hersteller sein.

Die oben genannten Erklärungen bezüglich der Produktstrategien und Absichtserklärungen von IBM stellen die gegenwärtige Absicht von IBM dar, unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Hergestellt in den USA
03-07

© Copyright IBM Corporation 2009
Alle Rechte vorbehalten.