



Enterprise Job Scheduling beflügelt iSeries-Anwendungen

Anwendungsintegration ist eine der größten Herausforderungen in der modernen IT. Unternehmensweite Job Scheduling-Lösungen bieten mit Batch Application Integration einen effektiven Ansatz, um auch die iSeries nahtlos in heterogene Umgebungen zu integrieren.

(erschieden in Midrange Magazin 10/2006, S. 42-46; Autor: Kai Leonhardt)

Für Betreiber von Rechenzentren ist eines der vorrangigen Anliegen, möglichst unabhängig von Zufällen und menschlichen Fehlern zu operieren. Je weniger manuelle Eingriffe Abläufe benötigen, desto stabiler und damit auch fehlerärmer ist der tägliche Betrieb in der Praxis. Dies gilt insbesondere für die Steuerung unternehmensweiter Anwendungen, die in heterogenen Umgebungen laufen und Plattformen wie Mainframe, Midrange-Systeme und Client-/Server-Strukturen einbinden. Speziell im iSeries-Umfeld, traditionell stark im SAP-Betrieb, nimmt daher der Bedarf an Anwendungsintegration und Automatisierung im Rechenzentrum permanent zu. In praktisch jedem Rechenzentrum sind dazu Job Scheduler im Einsatz, um IT-Prozesse automatisiert zu steuern. Dazu müssen Jobs nicht nur nach vordefinierten Zeitplänen verarbeitet werden, sondern auch auf geplante und ungeplante Ereignisse im Unternehmen reagieren können. Moderne Scheduling-Lösungen sind in der Lage, auch Anwendungsintegration weitgehend über Batchverarbeitung zu realisieren.

Anwendungsintegration im Batchverfahren

Für die Integration von Anwendungen existieren prinzipiell zwei Wege: über zeitversetzte Batchjobs, also die traditionelle Stapelverarbeitung, wie sie seit der Zeit der Großrechner bekannt ist, oder synchron über EAI (Enterprise Application Integration)-Werkzeuge. Bei der Batch Application Integration (kurz BAI) erfolgt die Integration von Anwendungen meist über Enterprise Job Scheduling Software, mit der prozessorientiert und asynchron Batchjobs ausgeführt werden. Zeit- oder ereignisgesteuert werden dabei vorab definierte Prozesse aufgerufen und abgearbeitet. Im Gegensatz dazu konvertiert EAI Daten und Befehle aus dem Format einer Anwendung in das Format einer anderen, um den Datenaustausch zwischen inkompatiblen Anwendungen zu ermöglichen. EAI-Lösungen integrieren Daten online, also in Echtzeit. In den meisten Fällen ist diese Verarbeitungsgeschwindigkeit beim Austausch von Daten zwischen Anwendungen aber gar nicht notwendig. Oftmals reicht es aus, nur einen kleinen Teil der Anwendungen realtime auszuführen und den Großteil mittels Batch Application Integration zu einem späteren Zeitpunkt.

Ein Beispiel: Die Eingabe neuer Kundeninformationen im CRM-System eines Mobilfunkproviders löst ein Ereignis aus. Als Reaktion startet der Job Scheduler eine ganze Prozesskette, welche die Kundendaten in alle relevanten Systeme (Billing, Buchhaltung, etc.) importiert und dafür sorgt dass die SIM-Karte aktiviert wird. Das alles erledigt ein Job Scheduler wesentlich billiger als alle vergleichbaren Lösungen. Im Gegensatz zu EAI erfolgt der Datenaustausch bei BAI nicht synchron, also in Echtzeit, sondern asynchron, also zeitversetzt unter Berücksichtigung der verfügbaren Rechenressourcen und konfigurierbarer Einstellungen. Ein Zeitversatz von einigen Sekunden spielt in den meisten Fällen keine Rolle, für viele Systeme ist auch eine größere Verzögerung beim Datenabgleich unproblematisch. Durch die auf diese Weise mögliche Lastverteilung kann die Hardware-Infrastruktur auf eine Durchschnittslast ausgelegt werden. Die Hardwarekosten fallen also im Vergleich zu EAI, wo Systeme stets auf die mögliche Maximallast ausgelegt werden müssen, gering aus. Auch bei den Lizenzkosten sorgt BAI für Einsparungen. Die benötigte Job-Scheduling-Lösung ist schließlich meist schon im

Haus. Sogar bei einer Neuan-schaffung – nicht alle Job Scheduler erfüllen die Voraussetzungen für BAI – liegen die Lizenzkosten deutlich unter jenen für EAI-Lösungen.

Realtime bringt nicht immer Vorteile

Ein Webshop ist ein klassisches Beispiel für Online-Verarbeitung in Echtzeit sollte man meinen. Tatsächlich aber agieren nur der Webserver und die Webdatenbank online. Besucht man den Webshop, so reagiert die Seite selbstverständlich auf jeden Klick. Der Besucher füllt seinen Warenkorb mit den gewünschten Utensilien. Mit dem Klick auf den „Bestellen“-Knopf startet nun die Hintergrundverarbeitung. Die Abfrage des Lagerbestands, Erstellung von Packliste und Lieferschein werden vom Job Scheduler gestartet und über die gesamte Laufzeit überwacht. Auch die Bestätigungsmails für Bestellung und Versand werden vom Job Scheduler ausgelöst. Diese asynchrone Verarbeitung der Bestellung reduziert die Hardwareanforderungen beträchtlich und reduziert daher die Kosten des Website-Betreibers. Denn bei synchroner Verarbeitung müssten auch die Backend-Systeme auf die Höchstlast ausgelegt werden. Für die Kunden hat das keinerlei Nachteil. Ob die Bestätigungsmail unmittelbar oder erst 30 oder 90 Sekunden nach der Bestellung verschickt wird, merkt kein Kunde. Wird die Packliste erst zehn Minuten nach der Bestellung erstellt, hat das keine Auswirkungen auf den Versand, der ein oder zwei Mal täglich erfolgt.

Die Implementierung von Batch Application Integration ist recht einfach. Moderne Job Scheduler bieten die Möglich-

keit, Prozesse anwendungs-übergreifend grafisch zu modellieren und zu überwachen. Dennoch sind nicht alle Job Scheduler auf dem Markt für BAI geeignet. Durch den Einsatz des Job Schedulers für BAI vervielfacht sich die Anzahl gesteuerter Prozesse. Skalierbarkeit ist daher eine grundlegende Voraussetzung, schließlich müssen in komplexeren Systemumgebungen in Spitzenzeiten mehr als 100.000 Arbeitsschritte pro Stunde abgewickelt werden. Hier hat zum Beispiel UC4 Software mit UC4:global eine Lösung auf dem Markt, die über beliebig viele CPUs und Server verteilt läuft, um maximale Performance zu liefern.

Scheduling für die iSeries

Wie jedes professionelle System verfügen auch die Server der iSeries über einen eigenen Job-Scheduler. Allerdings ist die Funktionalität des Standard-OS/400-Schedulers relativ begrenzt. Unternehmen, die Wert auf fortschrittliche Funktionen, etwa die Kontrolle von Abhängigkeiten zwischen einzelnen Jobs, legen, können aber glücklicherweise auf ein reiches Angebot an Lösungen von Fremd-anbietern zurückgreifen. Der Markt der Scheduler kann dazu in drei Kategorien eingeteilt werden: Nischen-, Abteilungs- und Enterprise Scheduler. Bei Nischen-Schedulern handelt es sich um hoch spezialisierte Werkzeuge für eine einzelne Plattform oder Anwendung. Abteilungs-Scheduler unterstützen auch andere Betriebssysteme, ihre Architektur spiegelt meist jedoch die Besonderheiten der ursprünglichen Plattform wieder, für die sie entwickelt wurden. Aufgrund ihrer Einschränkungen sind sie nur für kleine, eng abgegrenzte Projekte geeignet.

Enterprise-Scheduler wie beispielsweise UC4:global decken hingegen das gesamte Unternehmen, also alle Anwendungen und Plattformen, ab. Sie sind in der Lage, Jobs vom Mainframe über die iSeries bis auf die Ebene von PC-Servern zu steuern und dabei alle notwendigen Anwendungen, Daten oder Datenbanken zu integrieren. Da alle Systeme in ein einheitliches Ablauf- und Steuerungskonzept eingebunden werden, lassen sich OS/400-Jobs problemlos mit anderen Plattformen verbinden und von einem einzigen Monitor aus steuern. Für den Filetransfer von und zur iSeries wird ein integriertes Produkt genutzt, das für alle unterstützten Plattformen zur Verfügung steht und die Dateiübertragung verschlüsselt. Damit ist es nicht mehr notwendig, parallele Systeme für Job Scheduling und Filetransfer zu warten – die IT-Kosten sinken. Für eine optimale Interaktion zwischen OS/400 und UC4:global sorgen diverse Mechanismen. Einerseits wird die OS/400-Systemkonsole überwacht: In Abhängigkeit von auftauchenden Schlüsselwörtern werden Aktionen gestartet, viele Probleme werden durch die kontinuierliche Überwachung verschiedener Parameter bereits im Vorfeld erkannt und behoben. Auf der anderen Seite wird eine API für OS/400 angeboten, mit der die Integration von Anwendungen und Eigenentwicklungen realisiert wird. Über Standardprogrammiersprachen können diese Anwendungen direkt auf die Funktionen von UC4:global zugreifen und lassen sich so schnell und einfach einbinden.

Softwarearchitektur als entscheidender Faktor

Neben der Funktionalität ist die Architektur eines Job Sche-

dulern von großer Bedeutung, denn sie bestimmt Zuverlässigkeit, Skalierbarkeit und ist für den Datenfluss zwischen den verschiedenen Anwendungen und Plattformen verantwortlich. UC4:global ist offen aufgebaut und eignet sich mit seinen Schnittstellen optimal für die Anforderungen heterogener Systemlandschaften. Der Scheduler läuft auf einer eigenen Maschine und baut von dieser aus über seine Schnittstellen Verbindungen zu allen Systemen auf. Dazu werden auf den zu steuernden Plattformen wie der iSeries oder dem Webserver Agenten installiert, die die Kommunikation ermöglichen. Diese schlanken Programme benötigen nur minimale Systemressourcen und beeinträchtigen somit die Performance nicht. Die Steuerung der einzelnen Jobs und die Überwachung der Prozesskette gelingt über eine grafische Oberfläche. Wer will, kann natürlich die iSeries-Prozesse auch mittels Kommandozeilen von der OS/400 Konsole aus steuern.

Mittels einfachem „drag & drop“ können Objekte zu Prozessketten zusammengestellt werden. Wegen des durchgehend objektorientierten Aufbaus können einzelne Jobs problemlos bearbeitet, kopiert und auch verschoben werden. So lassen sich nach Muster der klassischen IT eine Test-, eine Entwicklungs- und eine Produktivumgebung betreiben. Aufgrund der Mandantenfähigkeit der Lösung kann ein Prozess problemlos zwischen den Mandanten ausgetauscht werden. In der Praxis heißt das, dass ein Job in der Entwicklungsumgebung erstellt und in der Testumgebung ausprobiert werden kann, bevor er in die Produktivumgebung integriert wird, wo er dann sofort einsatzfähig ist. Dazu muss das ent-

sprechende Objekt lediglich in einen virtuellen „Transportkoffer“ geschoben und dann einem Mandanten über die grafische Oberfläche zugeordnet werden. Einzelne Schritte in komplexen Jobplänen entsprechen dabei auch einzelnen Objekten. Das hat zur Folge, dass eine Änderung an einem Objekt ausreicht, um eine Änderung in allen betroffenen Jobs durchzuführen – eine enorme Arbeitserleichterung für das IT-Team.

Integration des Workload-Managements

Job Scheduling ist ein wichtiger Schlüssel für einen reibungslosen IT-Betrieb und damit zur Wettbewerbsfähigkeit eines Unternehmens. Auch wenn es eine etablierte Technologie ist, ergeben sich wegen ihrer universellen Einsatzfähigkeit immer wieder neue Wirkungsfelder. Nicht jeder Prozess bringt den Nutzern einen Mehrwert, wenn er „realtime“ erfolgt. Einige Prozesse müssen aber in Echtzeit abgearbeitet werden und benötigen dazu maximale Performance und entsprechende Systemressourcen. Vor diesem Hintergrund hat Gartner im vergangenen Jahr das Konzept des „IT Workload Automation Brokers (ITWAB)“ entwickelt. Die Vorteile einer Verbindung von Job Scheduling auf der einen und Workload-Management auf der anderen Seite liegen auf der Hand: die Scheduling-Lösung „weiß“ bereits im Vorfeld, wann welche Jobs laufen, und kann diese Informationen an die Lösung zur Workload-Steuerung weiterleiten, die dann entsprechende Ressourcen be-

reitstellt. Derzeit ist zwar noch kein einzelnes Unternehmen in der Lage, alle Elemente eines IT Workload Automation Broker aus eigener Hand anzubieten, aber ein Trend in diese Richtung ist bereits klar erkennbar. Entwickeln werden sich diese Werkzeuge nach Einschätzung von Gartner in der Hauptsache aus Werkzeugen für Job Scheduling, Anwendungsintegration sowie Prozessautomation. Aktuell lässt sich bereits die Entwicklung von Funktionen beobachten, mit denen sich etwa WebSphere MQ, Java Messaging Service oder Web Services handhaben lassen. Darüber hinaus gehende Integration, etwa mit Workload-Management, Prozessmodellierung, Grid-Computing und dem Gebrauch von Web-Services zwischen kundenspezifischen Anwendungen und solchen „von der Stange“ in einer heterogenen Umgebung, müssen jedoch erst noch geschaffen werden. An dieser Stelle werden vor allem die Anbieter im Vorteil sein, deren Lösungen objektorientiert aufgebaut sind.