TABEX
JTC

TABEX4
JAVA TABLE CACHE
MACHEN SIE DAS BESTE AUS IHREN STAMMDATEN

BOI
BETTER
ORGANIZED
INFORMATION

www.boi.at

TABEX4 JAVA TABLE CACHE (JTC)

MACHEN SIE DAS BESTE AUS
IHREN STAMMDATEN

Überfordern Ihre Business-Anwendungen Ihre Server mit Stammdatenabfragen?

BOIs JAVA TABLE CACHE (**JTC**) ist die Lösung für Sie. Mit JTC...

- erzeugen Sie **sicher und angepasst an Ihre Bedürfnisse** JTC Snapshots Ihrer Stammdaten aus allen vorhandenen Stammdatenspeichern
- replizieren Sie diese JTC Sn apshots zwischen Ihren Rechenzentren oder weltweit in Ihrem Netzwerk unter Verwendung Ihrer bestehenden Infrastruktur auf eine **unlimitierte Anzahl an JTC Clients**
- profitieren Sie vom **komfortablen und ultra-schnellen in-memory-Zugriff** auf Ihre Stammdaten mittels Java in den JTC Clients
- genießen Sie die Sicherheit, dass Ihre IT Architektur nun die **automatisierte Verteilung** und den Zugriff auf Ihre Stammdaten **vollkommen revisionssicher** unterstützt

JTC im Überblick

JTC verteilt Ihre Stammdaten an Ihre Java Business-Anwendungen.

JTC, ein Produkt der **BOI Software Entwicklung und Vertrieb GmbH (BOI)**, verteilt Ihre Stammdaten an Ihre Java Business-Anwendungen. JTC ist linear skalierbar und stellt in sich konsistente, versionierte, revisionssichere und für den ultra-schnellen, lesenden Java-Zugriff optimierte Stammdaten zur Verfügung.

JTC besteht aus drei Komponenten: dem **JTC Publisher**, einem **Replikationssystem**, sowie den **JTC Clients**. Der JTC Publisher sammelt Stammdaten aus unterschiedlichen Quellen, bereitet diese nach Ihren Anforderungen in tabellarischer Form auf und veröffentlicht sie unter Verwendung eines Replikationssystems auf JTC Clients. Ein JTC Client ist eine kleine Java Bibliothek, die in Ihre Java Business-Anwendungen eingebunden wird. Der JTC Client kopiert die replizierten Stammdaten in den lokalen JVM-Speicher und stellt Java-Methoden für den performance-optimierten in-memory Zugriff zur Verfügung. Die Verteilung vom JTC Publisher zu den JTC Clients erfolgt durch ein Replikationssystem Ihrer Wahl.

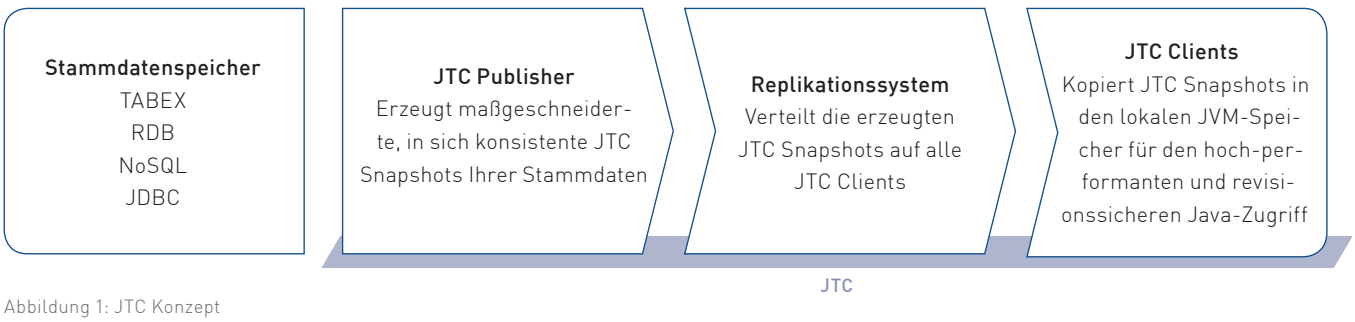

Abbildung 1: JTC Konzept

JTC ist in zwei Editionen erhältlich.

JTC ist in zwei Editionen erhältlich. Die **JTC Server Edition** wurde für Kunden entwickelt, die kein skalierbares Replikationssystem zur Verteilung ihrer Stammdaten zur Verfügung haben. Die JTC Server Edition stellt hierfür das BOI-eigene hierarchische Replikationssystem zur Verfügung. Die **JTC Enterprise Edition** ermöglicht Ihnen, Ihre Investition in ein Replikationssystem mehrfach zu nutzen: Sie können jede bestehende Enterprise-Caching-Lösung für die Verteilung verwenden, sofern diese PUT- und GET-Semantik unterstützt.

Einleitung: Stammdaten und Geschäftsprozesse

Stammdaten sind wichtige Geschäftsdaten, die zentral in einer Datenquelle gepflegt werden (sollten), aber verteilt in diversen Systemen, Anwendungen und Prozessen im Unternehmen benötigt werden. Da sich diese Stammdaten zwar typischerweise nur selten ändern, jedoch intensiv genutzt werden, müssen sie in der gesamten Organisation verteilt werden.

Stammdaten sind wichtige Geschäftsdaten, die zentral gepflegt werden sollten.

Die Bandbreite von Stammdaten in einer Organisation ist enorm: sie reicht von einfachen Listen, wie z.B. Postleitzahlen, bis hin zu komplexen Parameterdaten für die Kalkulation von Lebensversicherungspolizzen, angepasst an die regulatorischen und geschäftlichen Anforderungen in verschiedenen Ländern.

Stammdaten	Abteilung/Branche
Tabellen für Lebensversicherungsprämien	Versicherung
Zins- und Umrechnungskurstabellen	Finanz
Tabellen für Versand- und Postgebühren	Logistik
Transportfahrpläne	Logistik
Stücklisten von Maschinenersatzteilen	Produktion/Betrieb
Tabellen für Mietpreise von Leihwägen	Autoverleih
Gehaltstabellen	HR/Management
Steuertabellen für Programme (Geschäftslogik)	Management
Data Warehouses, Business Intelligence	Management
Andere Formen von Referenzdaten	

Tabelle 1: Beispiele für Stammdaten

Eine große Vielfalt an Softwarelösungen ist heutzutage für das Management von Stammdaten erhältlich. Es fehlen jedoch Produkte, die sich explizit auf die Auswahl und Aufbereitung von Stammdaten, deren Verteilung und den ultra-schnellen, lesenden Zugriff auf diese ausgewählten Stammdaten konzentrieren. JTC ist ein hoch fokussiertes Produkt, das diese existierende Lücke schließt.

JTC: Verteilung und ultra-schneller Java Zugriff auf Stammdaten.

Die drei JTC Komponenten

JTC besteht aus drei Komponenten, die zusammen den konsistenten Zugriff auf die Stammdaten in Ihrer gesamten Organisation sicherstellen:

- dem JTC Publisher
- einem Replikationssystem Ihrer Wahl
- den JTC Clients

Abbildung 2 zeigt beispielhaft ein JTC Deployment mit seinen drei Komponenten.

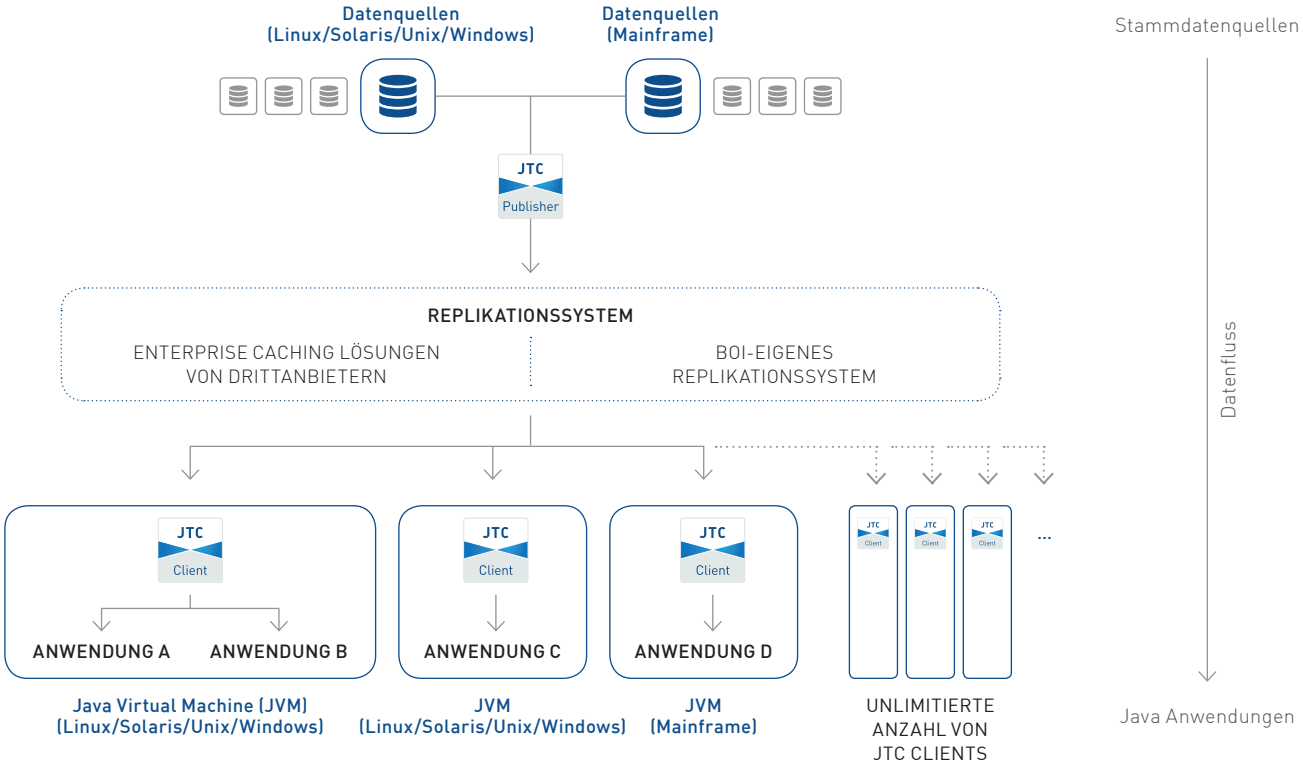

Abbildung 2: Schematisierte Darstellung eines JTC Deployments

JTC Publisher

JTC Publisher konsolidiert Ihre Stammdaten nach Ihren Vorgaben in einen JTC Snapshot.

Der JTC Publisher ist die serverseitige Komponente, die Stammdaten aus unterschiedlichen Datenquellen Ihrer Organisation sammelt. Der **JTC Publisher konsolidiert Ihre Stammdaten nach Ihren Vorgaben in einen JTC Snapshot**. Die Stammdaten für einen JTC Snapshot können hierbei aus verschiedenen Datenquellen unter verschiedenen Betriebssystemen ausgewählt werden. Die ausgewählten Tabellen können auch verändert werden. Zusätzliche Views und Indizes können zum Veröffentlichungszeitpunkt angelegt werden. Ein JTC Snapshot ist nichts anderes als ein in sich konsistentes, nachvollziehbares Subset Ihrer Stammdaten. Ob Sie JTC Snapshots periodisch, automatisch oder manuell erzeugen, liegt ganz bei Ihnen.

Jeder JTC Snapshot hat eine eindeutige ID und besteht aus einem Set an JTC Tabellen und deren korrespondierenden Indizes. Sie definieren den Inhalt jeder JTC Tabelle des JTC Snapshots durch die Konfiguration der Suche in einer oder mehrerer Ihrer Datenquellen.

Eine JTC Tabelle ist eine matrix-artige Struktur mit Spalten und Zeilen. Jede Spalte hat einen zugewiesenen Namen und einen Datentyp. Jede Zeile ist ein Datensatz. Da eine JTC Tabelle in Wahrheit eine Datenstruktur in Java ist, sind die Datentypen der Spalten ebenfalls Java-Datentypen. Im Rahmen der Konfiguration des JTC Publishers können Sie spezifizieren, wie die Spaltentypen zwischen Ihren Datenquellen und Ihren JTC Tabellen abgebildet werden sollen.

Der JTC Publisher ist auch **die Zentrale für alle Konfigurationen**, die Sie für JTC benötigen. Die gesamte Verteilung der JTC Snapshots auf die JTC Clients wird zentral über den JTC Publisher konfiguriert. Auf JTC Client-Ebene ist nur anzugeben, wo das Replikationssystem aufzufinden ist und wie sich der JTC Client damit verbinden kann.

JTC zwingt Sie nicht zur Aufgabe Ihrer bereits bestehenden Stammdatenquellen. Ganz im Gegenteil – **Sie können alle Werkzeuge, die Sie bereits für die Pflege Ihrer Stammdaten verwenden, uneingeschränkt weiter nutzen**. JTC kann Stammdaten aus einer Vielzahl von Datenquellen sammeln, unter anderem aus relationalen Datenbanken, NoSQL (i.e. non-relational) Datenbanken, TABEX Datenbanken und TABEX ESA (SHS)-Datenräumen, Excel- oder CSV-Files oder deren Kombination.

Replikationssystem

Information fließt durch das Replikationssystem immer vom JTC Publisher zum JTC Client, nie anders herum. Daher benötigt JTC keine verteilten Transaktionen oder Sperren, und **es gibt keine Beschränkung der Anzahl an JTC Clients**, die die JTC Snapshots der Stammdaten erhalten und cachen können.

Information fließt durch das Replikationssystem immer vom JTC Publisher zum JTC Client.

Das Replikationssystem kann unabhängig vom JTC Publisher und den JTC Clients an Ihre Bedürfnisse angepasst werden und ist daher jene Komponente, die für die flexible Skalierbarkeit sorgt.

Das Replikationssystem erfüllt Ihre Anforderungen in Sachen Skalierbarkeit auf mindestens drei Arten. Die geographische Skalierbarkeit bezieht sich auf die Fähigkeit des Systems, Stammdaten über weite räumliche Distanzen zu verteilen, vielleicht sogar auf der ganzen Welt. Die Skalierbarkeit über heterogene Anwendungen bezieht sich auf die Fähigkeit des Systems, Stammdaten über viele unterschiedliche Typen von Anwendungen und Betriebssystemen zu verteilen. Die Skalierbarkeit über homogene Anwendungen bezieht sich auf die Fähigkeit des Systems, Stammdaten über viele verschiedene Instanzen einer einzigen Anwendung zu verteilen.

Die Entscheidung, welches Replikationssystem Sie nutzen, liegt ganz bei Ihnen. Die JTC Server Edition enthält das BOI-eigene hierarchische Replikationssystem für Stammdaten, welches für die ausschließliche Nutzung innerhalb eines JTC Systems optimiert ist. Das Replikationssystem der JTC Server Edition hat durch seine Replikationsrichtung vom JTC Publisher zum JTC Client und die enge Kopplung mit den JTC Clients den großen Vorteil, die geringste Latenz in der Anlieferung der JTC Snapshots zu erzielen.

Die Entscheidung, welches Replikationssystem Sie nutzen, liegt ganz bei Ihnen.

Nutzen Sie JTC Enterprise Edition, um Ihre bestehende Enterprise-Caching-Lösung wirksam und mehrfach einzusetzen: sowohl der JTC Publisher als auch die JTC Clients können direkt mit den „Softwarelösungen" Hazelcast, Infinispan und Terracotta verwendet werden. Die JTC Enterprise Edition ist auch mit anderen Objekt-Caching-Lösungen kompatibel. Da der Informationsfluss ausschließlich vom JTC Publisher zum JTC Client erfolgt, reduzieren sich die Anforderungen, die JTC an Enterprise-Caching-Lösungen stellt, auf ein Minimum:

- Sie müssen GET- und PUT-Operationen unterstützen, in denen die Schlüssel Strings sind und Werte aus einer (möglichst großen) Sequenz an Bytes bestehen
- PUT-Operationen werden nur am Knoten des JTC Publishers, GET-Operationen nur an den Knoten der JTC Clients benötigt
- GET-Operationen müssen ein Ergebnis liefern, das konsistent ist mit der Sequenz der PUT-Operationen
- Sie müssen eine Java-API haben

JTC lässt Ihnen die Wahl: Sie haben die Freiheit, sich für das Replikationssystem zu entscheiden, das Ihre Bedürfnisse in Bezug auf Skalierbarkeit, Latenz und Ressourcenverbrauch am besten abdeckt.

TABEX JTC

JTC Client

Ein JTC Client ermöglicht ultra-schnellen in-memory Java-Zugriff.

Ein JTC Client ist eine kleine Java-Bibliothek, die in Ihre Java-Business-Anwendungen eingebunden ist. Er ermöglicht ultra-schnellen in-memory Java-Zugriff auf jene JTC Tabellen, die mit dem JTC Publisher veröffentlicht wurden.

Hierbei macht der JTC Client viel mehr, als einfach nur einen Wert vom Replikationssystem mittels eines GET-Requests anzufordern. Aus der Perspektive der Java-Anwendung ist der JTC Client eine in-memory-Datenbank, die ein Maß an Konsistenz für den Lesezugriff zur Verfügung stellt, die ansonsten nur in den striktesten Isolationslevels von Standarddatenbanken erreicht wird.

Da der JTC Client eine Java-Bibliothek ist, kann dieser dazu beitragen, einige verbreitete Schwierigkeiten in der Programmierung von Business-Anwendungen zu beheben. Diese Schwierigkeiten werden durch die folgenden Fragen und Antworten zusammengefasst:

Sind Sie beunruhigt, dass Ihre Stammdaten nicht transaktionskonsistent sind?

JTC garantiert transaktionskonsistenten Stammdatenzugriff.

Wenn Sie eine Abfrage an eine relationale Datenbank richten, muss das gesamte Resultat dieser Abfrage in sich konsistent sein. Eine Möglichkeit, diese innere Konsistenz sicherzustellen ist, die transaktionale Konsistenz der Abfrage selbst sicherzustellen. Das heißt, Sie müssen die Möglichkeit ausschließen, dass es zu dirty reads, non-repeatable reads oder phantom reads kommen kann, die aufgrund anderer, gleichzeitig laufender Transaktionen eingeschleust werden könnten. Um die gewünschte Transaktionskonsistenz sicherzustellen, werden Sie typischerweise die Isolationsstufe der Transaktion auf „serializable" setzen.

Software-Architekten versuchen oft, diese Isolationsstufe zu vermeiden, da diese zu hohen Lasten für die betroffene Datenbank führen kann. Der Grund hierfür ist, dass die Datenbank fähig sein muss, alle Updates der betroffenen Tabellen zu unterstützen, und dabei möglicherweise mehrfache Kopien (oder mehrfache Sperren) zu halten, solange die Abfrage dauert. Wird aber eine schwächere Isolationsstufe gewählt, um die Last auf die Datenbank zu vermindern, geht damit das Risiko einher, dass das Resultat der Abfrage transaktions-inkonsistent ist.

In JTC werden Ihre Abfragen aus den Business-Anwendungen an den JTC Snapshot Ihrer Stammdaten gerichtet, anstatt direkt an die Stammdatenspeicher. Diese Architektur hat zwei Auswirkungen: Zum einen muss der JTC Snapshot in sich konsistent sein, da ansonsten die Abfrageresultate der Business-Anwendung nicht in sich konsistent sein können. Zum anderen wird die Abfragefrequenz an Ihre Stammdatenspeicher drastisch reduziert, da diese ausschließlich für die seltene Erzeugung der JTC Snapshots benötigt wird.

Diese Verringerung der Abfragefrequenz reduziert die Bedeutung der Performance der Isolationsstufe einer Transaktion drastisch. Durch die Erzeugung in sich konsistenter JTC Snapshots können Sie die Sicherheit der Isolationsstufe „serializable" für Ihre Stammdaten- Transaktionen genießen, ohne dass dies mit einer erhöhten Last der Datenbanken einhergeht.

Es gibt **keine Performanceeinbuße**, wenn von den JTC Clients auf einen JTC Snapshot über einen längeren Zeitraum zugegriffen wird: keine Datenbank-Locks müssen gehalten, keine Transaktionsdatensätze müssen für verschiedene Versionen gepflegt werden. Die Verwendung von JTC Snapshots durch die JTC Clients ist auch thread-sicher: Sie können auf denselben oder einen anderen JTC Snapshot von verschiedenen Threads aus ohne Gefahr und ohne Performance-Einbußen zugreifen.

Sind Ihre Programmierer gezwungen, ihre eigene Suche zu kodieren?

Der JTC Client bietet schnellen Schlüssel-Zugriff auf Ihre in-memory JTC Tabellen.

Sie steuern schon jetzt Stammdaten in Ihren Java Anwendungen ein? Dann kennen Sie das Problem wahrscheinlich, dass die Daten zwar schon im Speicher sind, man aber in diesen nicht effizient suchen kann. Die mächtigen Suchwerkzeuge, die von relationalen Datenbanken angeboten werden, stehen nicht zur Verfügung. Es gibt auch keine gebräuchlichen Werkzeuge, um tabellarische Daten innerhalb einer Java-Applikation zu suchen, ohne zu einer vollständigen in-memory Java-Datenbank zu wechseln.

Auch wenn in-memory Java-Datenbanken viel schneller als Client-Server Datenbanken sind, sie sind langsam im Vergleich zu JTC. Denn sie werden durch die Anforderungen, die mit variablen, geteilten Daten verbunden sind, verlangsamt (ganz zu schweigen vom JDBC Interface).

Zusätzlich zum schnellen Schlüssel-Zugriff auf Ihre in-memory JTC Tabellenzeilen bietet der JTC Client komfortable Java-Methoden für die Suche in einer oder mehreren Spalten nach folgenden Suchoperatoren:

- „gleich" (auch zusammengesetzte Schlüssel)
- „ungleich" (z.B. Range)
- „<", „>", „≤", „≥", „enthält", „enthält case-sensitive", „beginnt-mit", „beginnt-mit-case-sensitive", „endet-mit", „endet-mit-case-sensitive", und „regular-expression"

Kodieren Ihre Programmierer ihre eigenen Indizes (oftmals in Form von Hash Tables)?

Die Suchabfragen des JTC Clients sind index-beschleunigt.

Neben Komfort ist Performance oft eine wesentliche Anforderung, wenn Java-Anwendungen Daten durchsuchen. Programmierer greifen dabei häufig auf primitive Indizes mittels Hash Tables oder auf binäre Suchbäume zurück. Diese Praxis führt zu sich wiederholenden, hart-kodierten Indizes, ein Umstand, den Sie und Ihre Programmierer jedenfalls vermeiden sollten.

Der JTC Publisher kann Indizes automatisch erzeugen und verteilen: Sie müssen nur konfigurieren, für welche Tabellen welche JTC Index-Tabellen generieren werden sollen. **Der JTC Client stellt neben den Basistabellen auch die jeweils aktuellen JTC Index-Tabellen für die Abarbeitung von in-memory-Suchabfragen zur Verfügung.** Durch die Nutzung von Index-Tabellen entsteht für Sie folgender Nutzen:

- Sie erhalten ultra-schnelle Suchresultate
- Ihre Programmierer müssen keine eigenen Suchindizes erzeugen
- Sie können Indizes auch später im Rahmen der Konfiguration hinzufügen, und diese stehen allen JTC Clients zur Verfügung
- Sie müssen sich keine Sorgen machen, dass Tabellenversion und Index nicht zusammenpassen
- Sie müssen Ihre Indizes nicht neu aufbauen, wenn ein neuer JTC Snapshot verfügbar ist

Ihre Suchabfragen des JTC Client sind index-beschleunigt.

JTC: Erweiterte Stammdatenverteilung

Bisher haben wir jede der drei Komponenten von JTC einzeln beschrieben und einige der Features hervorgehoben, die Sie dabei unterstützen, Ihre Stammdaten konsistent und automatisiert in Ihrer Organisation zu verteilen. Nun ist es an der Zeit, JTC im Gesamten zu betrachten:

Zeitbezogene JTC Snapshots

Jeder JTC Client kann mehrere aufeinanderfolgende JTC Snapshots aufnehmen.

Stellen Sie sich vor, Sie müssen einen Bericht oder ein Angebot auf Basis der Stammdaten von gestern schreiben, und diese haben sich bereits geändert. Wie können Sie dies bewerkstelligen?

Jeder JTC Client kann mehrere aufeinanderfolgende JTC Snapshots aufnehmen. Hierdurch stehen gleichzeitig mehrere JTC Snapshots für Ihre Business-Anwendungen zur Verfügung. Haben Sie in Ihren Datenquellen fachliche Gültigkeiten für Ihre Tabellen angegeben (Gültigkeitsdatum von – bis), so sucht der JTC Client automatisch jene Version der gewünschten Tabelle aus, die für das von Ihrer Business-Applikation angegebene Datum gültig war. Sie können somit Ihre bestehenden TABEX-Tabellen im JTC unverändert einsetzen.

Haben Sie keine fachliche Gültigkeiten in Ihren Tabellen angegeben, so haben Sie die Möglichkeit, eine Tabellenversionierung in JTC umzusetzen. Durch Konfiguration des JTC Publisher aktivieren Sie die automatische JTC Tabellenversionierung: Jede JTC Tabelle, die kein fachliches Gültigkeitsdatum besitzt, wird mit dem Gültigkeitsdatum des Erstellungszeitpunktes des JTC Snapshots versehen. Somit besitzt jede Tabellenversion ein gültig-ab-Datum. Der JTC Client sucht somit automatisch jene Version der gewünschten Tabelle, die für das von Ihrer Business-Applikation mitgegebene Datum gültig war. Sie können auch, wenn Sie möchten, auf verschiedene JTC Snapshots gleichzeitig zugreifen (zum Beispiel in einer multi-thread-Umgebung wie ein Java Application Server).

Mit einer einfachen Parameterangabe im Rahmen der JTC Publisher-Konfiguration legen Sie fest, wie viele JTC Snapshots gleichzeitig im JTC Client verfügbar gehalten werden.

Datenräume und Mandantenfähigkeit

JTC Datenräume unterstützen Mandantenfähigkeit.

Benötigen Ihre Geschäftsprozesse Stammdatensätze, die auf eine bestimmte geographische Region, eine organisatorische Abteilung oder einen Prozessreifegrad (zum Beispiel Entwicklung, Test, Produktion) angepasst sind?

Um eine gewünschte Region oder Abteilung auszuwählen, müssen Sie normalerweise Ihre Abfragen aus den Geschäftsanwendungen modifizieren, um Filterlogik einzubauen. Dies ist mit JTC nicht länger nötig. **JTC erlaubt in jedem JTC Client unabhängige Datenräume innerhalb desselben JTC Snapshots.** Jeder Datenraum kann die gleichen Tabellen- und Indexstrukturen enthalten, aber mit unterschiedlichen Dateninhalten.

JTC Datenräume werden oft herangezogen, um Mandantenfähigkeit zu unterstützen. Hierfür wird der JTC Publisher so konfiguriert, dass ein Datenraum je Mandant erzeugt wird. Der JTC Client muss nur mehr spezifizieren, welchen Datenraum er nutzen möchte. Es ist keine Anpassung der Abfragen nötig, da alle Tabellen bereits bei der Publikation vorgefiltert werden.

Revision der Stammdatenverteilung

JTC bietet Revisionssicherheit.

Arbeiten Sie in einer Branche, die höchste Transparenz und Nachvollziehbarkeit verlangt? Müssen Sie zu jedem Zeitpunkt nachweisen können, welcher Stand Ihrer Stammdaten von Ihren Business-Anwendungen verwendet wurde?

JTC bietet Ihnen die Möglichkeit der lückenlosen Protokollierung. Jeder JTC Snapshot hat eine eindeutige ID, wobei Sie JTC so konfigurieren können, dass die Daten zu jedem JTC Snapshot vollständig archiviert werden. Die Java Methoden des JTC Clients bieten die Möglichkeit, die JTC Snapshot-ID des verwendeten JTC Snapshots an Ihre Business-Anwendung zurückzuliefern. Sie können diese JTC Snapshot-ID protokollieren und mittels des JTC Snapshot-Archivs die verwendeten Daten Ihrer Business-Anwendung zu jedem Zeitpunkt zuordnen.

Darüber hinaus protokolliert der JTC Publisher auch die Verwendung des Replikationssystems, da jede PUT-Operation aufgezeichnet wird.

JTC: Höchste Zugriffsperformance

JTC Zugriffe sind bis zu 550-mal schneller als direkte Zugriffe über JDBC auf eine relationale Datenbank.

Der JTC Publisher erzeugt aus Ihrer Stammdatenauswahl sogenannte JTC Snapshots, welche mittels des Replikationssystems zu den JTC Clients transportiert werden. Ihre Business-Anwendung verwendet die Java-Methoden des JTC Client für den hoch-performanten Datenzugriff. Wie schnell ist der Datenzugriff im JTC im Vergleich zur gleichen Abfrage in einem Client-Server-Datenbanksystem oder einer in-memory Java-Datenbank?

Beschreibung der Zugriffsperformance Tests

Um diese Frage zu beantworten, führte BOI vergleichende Einzeltabellen-Abfragen in drei Szenarien durch:

1. JTC Enterprise Edition mit Hazelcast über JTC Client
2. Oracle 11g Release 2 über Oracle Thin JDBC-Treiber
3. H2 in-memory über H2 JDBC-Treiber

Die Testumgebung für alle drei Szenarien war identisch und bestand aus:

1. Virtuelle Maschine: IBM x 3650 M4 Server bestückt mit 12 Intel Xeon E5-2630 Prozessoren (2,30 GHz)
2. Virtualisierte Gastumgebung: SUSE Linux Enterprise Server 11 64-Bit mit vier Rechenkernen und 3833 MB RAM
3. Oracle Java-Version 1.6

Der IBM-Server wurde während der Tests nicht anderweitig genutzt. Alle Testkomponenten liefen in einer einzigen virtuellen Maschine.

Jeder Test bestand aus nachfolgenden Schritten:

1. Zufällige Auswahl eines Primärschlüsselwertes
2. Abfrage einer einzelnen Tabelle über alle Spalten nach der Zeile, die den Primärschlüsselwert enthält
3. Wiederholen des Ablaufs für eine feste Anzahl von Iterationen

Alle Tabellenfelder enthielten Zeichendaten variabler Länge von bis zu 255 Zeichen. Die Tests untersuchten den Performanceunterschied in Abhängigkeit von:

1. Tabellengröße (Zeilen- und Spaltenanzahl)
2. Spaltenanzahl des Primärschlüssels (eine oder zwei)
3. Abfrage-Engine (Oracle, H2 und JTC)

Um zu einem aussagekräftigen Testszenario zu gelangen, wurden vorbereitete Kompilierungsanweisungen für Oracle und H2 verwendet. Für JTC wurde der Erstzugriff nicht gewertet. Zusätzlich wurden in jedem Szenario entsprechende Indizes verwendet.

Die Testszenarios wurden so gewählt, dass Medieneigenschaften wie IOPS, Festplattenlatenz und Festplattendurchsatz keinen Einfluss auf die Testergebnisse hatten. Daher wurden keine Daten mutiert. Für Oracle wurden vorgefüllten Tabellen verwendet, für H2 wurden die Tabellen in den Speicher geladen. JTC verwendete JTC Snapshots.

Testergebnisse

Die beschriebenen Tests ergaben, dass Zugriffe in JTC bis zu 550-mal schneller sind als direkte Zugriffe in Oracle. Selbst im Vergleich zu Zugriffen in der in-memory Java Datenbank H2 ist JTC um den Faktor 10 schneller.

Die genauen Ergebnisse der Tests im Vergleich zu Zugriffen in Oracle sind in Tabelle 2 zusammengefasst.

Tabelle 2: Testergebnisse für Daten-Abfragen unter Verwendung von Oracle, H2 und JTC.

Tabellengröße		Oracle 11g Release 2 via JDBC Thin client-side driver	In-Memory H2 Database Engine via JDBC		JTC EE + Hazelcast with stable mode (without a near cache)	
		Accesses/Sec	Accesses/Sec	X mal schneller als Oracle	Accesses/Sec	X mal schneller als Oracle
100x10	1*	10.799	434.782	40	6.250.000	579
	2*	10.504	392.156	37	3.676.470	350
1,000x25	1*	10.570	279.329	26	5.263.157	498
	2*	10.482	245.700	23	3.115.264	297
10,000x50	1*	9.765	165.016	17	3.690.036	378
	2*	9.718	159.744	16	2.267.573	233

* Spaltenanzahl des Primärschlüssels

Sind die extrem hohen Performance-Gewinne mit JTC plausibel?

Ja. JTC nutzt eine Systemarchitektur, die auf höchste Performance ausgelegt ist. JTC bringt die Stammdaten in Ihre Business-Anwendung und bietet clientseitig optimierte Java-Methoden für den direkten Datenzugriff. Folglich ist die JTC Client-Zugriffsgeschwindigkeit unbeeinflusst von:

1. Zusätzlichen Software-Schichten wie z.B. JDBC-API
2. Abfragekompilierung während der Laufzeit oder Look-up auf vorbereitete Anweisungen
3. Netzwerkverkehr durch Abfrage- und Antwortpakete
4. Kommunikation zwischen Prozessen
5. Kontextwechsel

Diese Tests zeigen das Einsparungspotential bei Einsatz von JTC. BOI ist sich darüber bewusst, dass allgemeine Testszenarios nie die individuelle Kundensituation widerspiegeln können. Wir bieten Ihnen gern an, die Einsparungen durch den Einsatz von JTC in Ihrem Anwendungsszenario auszuarbeiten.

TABEX JTC

Zusammenfassung

JTC unterstützt Sie dabei, den Nutzen Ihrer Stammdaten zu maximieren und ermöglicht Ihnen, Ihre etablierten Werkzeuge für Ihr Stammdaten-Management beizubehalten. JTC ist die ideale Erweiterung Ihrer TABEX-Installation für Java Anwendungen.

Der JTC Publisher unterstützt Sie bei der Erzeugung maßgeschneiderter, in sich konsistenter JTC Snapshots Ihrer Stammdaten von praktisch jeder erdenklichen Datenquelle oder Kombination von verschiedenen Datenquellen. Sie wählen ein optimales Replikationssystem, um Ihre JTC Snapshots zu verteilen. Dies kann entweder eine Enterprise-Caching-Lösung eines Drittanbieters sein, die Sie bereits im Einsatz haben, oder die BOI-eigene JTC Server Edition. Der JTC Client bietet Ihren Java-Business-Anwendungen revisionssicheren, ultraschnellen und thread-sicheren in-memory-Zugriff auf Ihre replizierten JTC Snapshots.

Zusammen ermöglichen die drei Komponenten eines JTC Systems für Ihre Business-Anwendungen Stammdatenzugriffe, die skalierbar, komfortabel, ultra-schnell und revisionssicher sind und sich durch geringe Latenz auszeichnen. Mit dem Deployment von JTC bewahren Sie volle Kontrolle über alle Aspekte Ihres Stammdatenmanagements und Ihrer Geschäftspolitik. JTC erlaubt Ihnen, Ihre etablierten Geschäftsprozesse und Ihre Werkzeuge für die Sammlung und Pflege Ihrer Stammdaten unverändert beizubehalten.

Mit JTC maximieren Sie den Nutzen Ihrer Stammdaten für Ihr Unternehmen.

Maximieren Sie den Nutzen Ihrer Stammdaten.

Spezifikationen

JTC Publisher

- Betriebssysteme: AIX, Solaris, Unix, Linux, Windows, z/OS, BS2000, VSE
- Datenquellen: TABEX, DB2, Oracle, MySQL, Informix, PostgreSQL
- Web-basiertes Frontend

JTC Client

- Reines Java
- Benötigt Java Version 1.6 oder höher
- Keine Java Abhängigkeiten von Drittanbietern

Replikationssystem

JTC Server Edition

- JTC eigenes Replikationssystem
- Betriebssysteme: AIX, Solaris, Unix, Linux, Windows, z/OS, BS2000, VSE

JTC Enterprise Edition

- Hazelcast
- Infinispan
- Terracotta
- Jedes Replikationssystem, welches GET und PUT Operationen unterstützt

BOI BETTER ORGANIZED INFORMATION

BOI – Ihr Spezialist für hoch-performanten Zugriff auf Ihre Stammdaten

Für mehr Information zu JTC besuchen Sie bitte unsere Website www.boi.at oder kontaktieren Sie Frank Sinner unter sales@boi.at.

Email: sales@boi.at , Web: www.boi.at, FN 81632y Landesgericht Linz, UID: ATU24421409

BOI


BOI
BETTER
ORGANIZED
INFORMATION