

Oracle Highspeed für .NET mit TimesTen

Maik Sandmann und Claus Jandausch
ORACLE Deutschland B.V. & Co. KG
München

Schlüsselworte:

TimesTen, Oracle, .NET, Microsoft, Visual Studio, Windows, Performance, ODP.NET, ODT.NET, In-Memory, Database Cache, Skalierbarkeit.

Einleitung

Oracle TimesTen ist eine In-Memory-optimierte relationale Datenbank und bietet als solche volle SQL Unterstützung und eine tiefe Integration mit der Oracle Datenbank. Der Oracle Times Ten Data Provider for .NET ermöglicht .NET Anwendungen den schnellen Datenzugriff auf Times Ten und unterstützt dabei ODP.NET Klassen, Enumerationen, Interfaces, Delegates und Strukturen. Mit Verfügbarkeit der Oracle Data Access Components 11.2.0.2 (ODAC 11.2.0.2) unterstützt Oracle nun auch den Zugriff von Microsoft .NET Applikationen auf die Oracle TimesTen. Somit reiht sich TimesTen nahtlos ein in die Palette vieler weitere Oracle Features zur Performance Steigerung im .NET Umfeld – wie beispielsweise im Caching Bereich der Oracle Client Result Cache und die Oracle Continuous Query Notification.

Der Oracle Mehrwert für Microsoft .NET Applikationen

Microsoft .NET Applikationen profitieren in hohem Maße von den Oracle Advanced Features zur Gewährleistung von Skalierbarkeit und Verfügbarkeit, sowie zur Steigerung der Performance von .NET Applikationen. Ein alternativer Einsatz der Oracle Datenbanktechnologie anstelle des Microsoft SQL Server empfiehlt sich bereits aufgrund verschiedener Alleinstellungsmerkmale wie der Plattformunabhängigkeit und Portierbarkeit der Oracle Datenbank, wovon auch der Betrieb von .NET Applikationen in hohem Maße profitieren kann, um beispielsweise nicht alleinig an Windows als Server Plattform gebunden zu sein, oder Hochverfügbarkeit im Cluster mit gleichzeitiger Skalierbarkeit zu verbinden, was mit der aktuellen Microsoft Plattformtechnologie in dieser Form nicht möglich ist. Weitere Vorteile bieten die vielfachen generischen Möglichkeiten der Oracle Datenbank zur Performance-Steigerung, die einer .NET Applikation allesamt transparent zur Nutzung zur Verfügung stehen. Unterschiede im Core-Bereich der Oracle und Microsoft Datenbank stellen die Multi-Version-Read-Consistency der Oracle Datenbank dar, um einen Kompromiss zwischen schnellen Antwortzeiten und korrekten Daten unnötig zu machen, wie sie beim SQL Server immer noch einkalkuliert werden muss (Stichwort: Datenbankabfragen mit der Option WITHNOLOCK). Ein weiterer Unterschied im Bereich der grundsätzlichen Architekturunterschiede, stellt das Locking-Verhalten dar. So kommt es mit dem SQL Server gerne zur Lock-Eskalation, wobei selbst lesende Zugriff Sperren setzen und ein Leser somit einen Schreiber aussperren kann. Lock-Eskalationen dieser Art gibt es mit der Oracle Datenbank nicht.

Um nun einer .NET Applikation all diese Vorteile zukommen zu lassen, können Entwickler auf die von Oracle frei zur Verfügung gestellte Integration mit .NET zurückgreifen. Maßgeblich stehen hierzu zwei Komponenten bereit – der Oracle Data Provider for .NET (ODP.NET) und die Oracle Developer

Tools for .NET (ODT.NET). Diese sind – neben Komponenten wie ODBC, OLE DB, ASP.NET – in den Oracle Data Access Components (ODAC) enthalten. Mit der aktuellen Version 11.2.0.2 ist hier auch die TimesTen Unterstützung integriert.

Transparenz für .NET Entwickler mit den Oracle Developer Tools for Visual Studio

Wichtig bei der Oracle Integration mit .NET ist die Transparenz in der Arbeit des Entwicklers – konkret die Unterstützung von Microsoft Visual Studio. Die *Oracle Developer Tools for Visual Studio* (ODT.NET) sind ein nahtlos in *Microsoft Visual Studio* integriertes ‚Add-in‘. Seit der ODAC Version 11.2.0.1.2 wird nun auch *Visual Studio 2010* unterstützt – neben *Visual Studio 2008* und *Visual Studio 2005*. Entwickler werden mit den ODT.NET dabei im gesamten *Development Lifecycle* unterstützt. Mit den *Oracle Developer Tools* unterstreicht Oracle immer wieder und nachhaltig sein Commitment zur Unterstützung der .NET Entwicklungsumgebung. Exemplarisch sei hier nur die Einführung eines neuen *SQL Tuning Advisor* und *Advanced Queuing Administration Tools* genannt. Selbigen Komfort sucht man in der Kombination Visual Studio mit SQL Server noch vergeblich.

Wer an einer genauen Beschreibung aller in ODT.NET enthaltenen Features interessiert ist, findet weiterführende Informationen, Video Demonstrationen und schrittweise Anleitungen (*OBE - Oracle By Example*) auf der ODT.NET Homepage.

<http://www.oracle.com/technetwork/developer-tools/visual-studio/overview/index-097110.html>

Datenzugriff für .NET Applikationen mit den Oracle Data Provider for .NET

Mit dem *Oracle Data Provider for .NET* (ODP.NET) profitieren .NET Anwendungen vom optimierten Datenzugriff auf die Oracle Datenbank im Sinne von Skalierbarkeit, Hochverfügbarkeit und Performance. Insbesondere die umfassenden Performance Features stellen für .NET Entwickler einen hohen Mehrwert dar. Hier zeigen sich die besonderen Vorteile gegenüber dem früher parallel zur Verfügung gestellten und im .NET Framework ausgelieferten *Microsoft .NET Oracle Data Provider*.

Seit der ODAC Version 11.2.0.1.2 unterstützt der ODP.NET sowohl das *.NET Framework Client Profile*, als auch die Vollversion des *Microsoft .NET Framework 4*.

Ein Upgrade von ODP.NET 11.1.0.7.0 auf die neueste Version macht auf jeden Fall Sinn und erfordert keine Code-Anpassungen. *Out-of-the-Box* und ohne Tuning ergaben sich in einem konkreten Kundenbeispiel bei identischer Last im Vergleich zu ODP.NET 11.1.0.7.0 folgende Performance-Steigerungen:

- Schnellere Ausführung (24%)
- Mehr Durchsatz (60%)
- Bessere Antwortzeiten (38%)
- Weniger Netzwerk-Traffic (21%)
- Weniger CPU auf sowohl Client (19%) als auch Server (17%)
- Mehr Transaktionen pro Sekunde auf Client (99%) und Server (92%)

Wer an den technischen Details von ODP.NET interessiert ist, findet weiterführende Informationen und aktuelle technische Beschreibungen der einzelnen Mehrwerte auf der ODP.NET Homepage.

<http://www.oracle.com/technetwork/database/windows/index-085163.html>

Skalierbarkeit und Highspeed für .NET Applikationen mit Oracle Caching Technologien

Mit der grundsätzlichen Unterstützung der .NET Anwendungs- und Entwicklungsumgebung für den Zugriff auf die Oracle Datenbank als Single-Instance, auf unterschiedlichen Plattformen (Windows, Linux, Unix, etc.), oder innerhalb einer Grid-Infrastruktur (Stichwort Real Application Clusters), endet die Oracle Integration mit .NET jedoch lange nicht.

Insbesondere der ganze Bereich der *In-Memory Datenbanken* steht .NET Anwendungen transparent zur Nutzung zur Verfügung, um die Anforderungen an Highspeed Performance und Skalierbarkeit auf unterschiedlicher Ebene zu adressieren.

Verteilter In-Memory Cache mit Oracle Coherence for .NET

Vor der Unterstützung von *Oracle TimesTen* war es bereits möglich mit *Coherence for .NET* die Skalierbarkeit der *Application Tier* zu erreichen oder *Extreme Transaction Processing* zu realisieren.

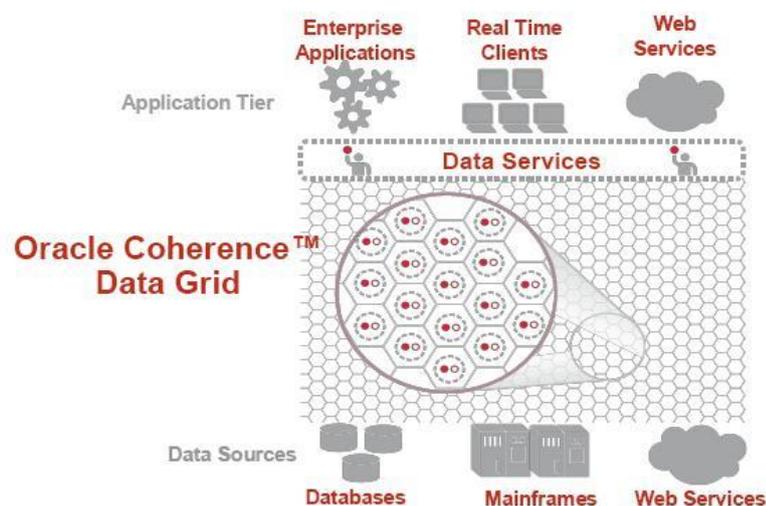


Abb. 1: Oracle Coherence Data Grid

Wie und wo soll man diesen verteilten Cache nun gedanklich einordnen? Als erstes muss man sich von der Vorstellung trennen, es handle sich hier um eine ähnliche Technologie wie die im Einsatz mit den *Oracle Real Application Clusters* bekannte Implementierung des *Cache Fusion*. Aus mehreren Gründen ist dies nicht der Fall. Coherence adressiert eine ganz andere Ebene als die der Datenbank und fungiert dabei als Datenservice für die *Application-Tier*. In dieser Aufgabe bringt eine ‚*Coherence Wolke*‘ idealerweise solche Daten nahe an die Applikationen, die vor allem dem häufigem lesenden Zugriff ausgesetzt sind. Stellt man sich ein ERP-System vor (wie beispielsweise SAP), lässt sich auf diese Art eine elegante Möglichkeit realisieren, um für häufig lesende Zugriffe die relevanten Daten, oder ein für die Anwender bereits fertig aufbereitetes Subset an Informationen, in eine Coherence Wolke zu verlagern, und somit das ERP-System zu entlasten. Der ansonsten notwendige direkte Zugriff wird dadurch sogar obsolet. Diese Lastreduzierung gemeinsam genutzter Datenquellen im Backend, stellt eine hochinteressante Möglichkeit dar, um die Skalierbarkeit einer gesamten Infrastruktur unter Kostengesichtspunkten möglichst effektiv zu gestalten, anstelle sich auf abgetrennte Segmente zu fokussieren.

Oracle Timesten für Highspeed Performance von .NET Applikationen

Timesten ist 1998 als Spin-Off gegründet worden und im Jahre 2005 von Oracle übernommen worden. Warum ist Timesten so schnell? Die nachfolgende Grafik soll den Unterschied im Zugriff von Timesten zu einem „Disk-Based“ RDBMS darstellen. Timesten ist entwickelt worden mit der Kenntnis, dass sich alle Daten im Hauptspeicher befinden. Damit ist es möglich einen direkten Pfad auf die Daten zu bekommen und dabei sowohl die Zugriffsalgorithmen und –strukturen einfach zu halten.

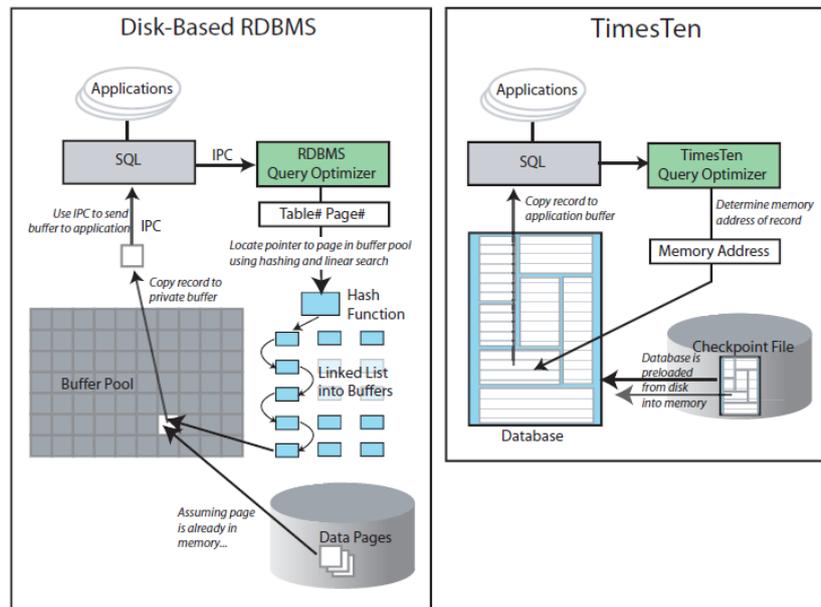


Abb. 2: Überblick Oracle Timesten vs eines Disk-Based RDBMS

Neben diesem architekturellen Unterschied verfügt Timesten über Funktionalitäten einer relationalen Datenbank, inklusive Hochverfügbarkeit und Skalierung, dem so genannten Cache Grid. Mittels Cache Grid skaliert Timesten horizontal / vertikal und kann als Active-Standby Lösung eingesetzt werden, wobei die integrierte Oracle Clusterware Failover und Recovery automatisiert. Die Administration und das Management von Timesten erfolgt über die zahlreichen Timestenklassen, den SQLDeveloper oder dem Timesten Plug-In für den Enterprise Manager. Weitere und sicherlich interessante Punkte, die in dem Vortrag angesprochen werden sind BI / OLAP Unterstützung, analytische Funktionen oder Komprimierung. Abgerundet wird der Vortrag mittels einer Online Demo zu Timesten und .net.

Kontaktadresse:

Maik Sandmann
ORACLE Deutschland B.V. & Co. KG
Thurnithistrasse 2
D-30519 Hannover

Claus Jandausch
ORACLE Deutschland B.V. & Co. KG
Riesstraße 25
D-80992 München

Telefon: +49 (0) 89-1430 2212
Fax: +49 (0) 511 95787 100
E-Mail: maik.sandmann@oracle.de
Internet: www.oracle.de

+49 (0) 89-1430 1520
+49 (0) 89-1430 2150
claus.jandausch@oracle.com
www.oracle.de