Oracle Data Provider para .NET 11*g*

Informe Técnico de Oracle
Octubre de 2007

ORACLE

Oracle Data Provider para .NET 11*g*

Introducción	3
Oracle Data Provider para .NET	4
Desempeño	4
Grupo de Conexiones y Caché de Sentencias	4
Control de Tamaño de los Datos Buscados	5
Optimización de Datos LOB	5
Datos de Matriz	6
Entorno .NET de 64 bits	6
Desempeño – Novedades en ODP.NET 11*g*	6
Caché de Resultados Cliente	6
Búsquedas LOB Más Rápidas	7
Caché de Sentencias Optimizado	7
Notificación de Cambios en la Base de Datos	7
Real Application Clusters (RAC)	9
Características XML	10
ADO.NET 2.0	11
Tipos Nativos de Oracle	11
Otras Características Importantes	12
Conclusión	13

INTRODUCCIÓN

Uno de los mayores beneficios de utilizar los productos Oracle es el soporte de múltiples entornos de programación. Al soportar aplicaciones .NET, Java/J2EE, PHP, y C/C++, todos los desarrolladores pueden utilizar las características avanzadas de la base de datos Oracle y ofrecer verdadera flexibilidad a las empresas de desarrollo. Cada uno de los impulsores para el acceso a datos Oracle está diseñado para maximizar el desempeño de su respectivo entorno y permitir el acceso a las características más recientes de la base de datos.

Dentro del ámbito .NET, Oracle ofrece una gran cantidad de productos para desarrollar aplicaciones con la base de datos Oracle, con inclusión de Oracle Developer Tools para Visual Studio .NET, Oracle Database Extensions para .NET, Oracle Providers para ASP.NET y Oracle Data Provider para .NET (ODP.NET).

ODP.NET es un proveedor de acceso de datos .NET nativo para las bases de datos Oracle. Ofrece acceso de datos ADO.NET estándar para .NET Framework 1.x, 2.0 y versiones posteriores. Para los desarrolladores que han utilizado otro proveedor ADO.NET, el acceso a los datos ODP.NET básico requiere conocer muy pocas cosas nuevas y puede utilizarse con construcciones existentes, como Microsoft Data Access Application Blocks (DAAB) para .NET. Así, los desarrolladores ADO.NET pueden comenzar a utilizar las fuentes de datos de Oracle rápidamente.

A pesar de que ODP.NET comparte elementos comunes básicos con otros proveedores ADO.NET, el valor exclusivo de ODP.NET es su estrecha integración con la base de datos Oracle. ODP.NET brinda a los desarrolladores .NET muchas de las capacidades exclusivas de la base de datos - como el ajuste de Real Application Clusters (RAC), la seguridad avanzada y los tipos de datos complejos. Estas características permiten a los niveles medios .NET aprovechar las capacidades exclusivas de la base de datos Oracle.

Este informe ejecutivo se focaliza en el acceso de datos ODP.NET y sus capacidades exclusivas, y trata sobre las características de desarrollo .NET que son parte de Oracle Database 11*g* y de versiones anteriores. Muchas características ODP.NET 11*g* se activan con versiones más antiguas del servidor de base de datos Oracle, como Oracle Database 10*g* y Oracle Database 9*i*. Este informe ejecutivo no incluye las características ODP.NET presentadas después de Oracle Database 11*g* en Windows, como ODP.NET 11.1.0.6.20.

ORACLE DATA PROVIDER PARA .NET

Como con otros proveedores de datos .NET, cualquier aplicación .NET, con inclusión de C# .NET,
Visual Basic .NET y ASP.NET, puede invocar ODP.NET. Aunque se implementa mayormente en los niveles medios, ODP.NET también puede utilizarse dentro del mismo servidor de base de datos por medio de los procedimientos almacenados .NET.

Oracle Data Provider para .NET permite un desempeño de rápido acceso a los datos de la base de datos Oracle. Soporta la funcionalidad más reciente de la base de datos disponible con Oracle Database 11*g*.

ODP.NET ofrece un mejor desempeño y un acceso más amplio a las características de la base de datos Oracle comparado con cualquier otro proveedor de datos .NET. ODP.NET ha sido específicamente diseñado para maximizar las capacidades de la base de datos Oracle.

ODP.NET 11*g* introduce nuevas características de desempeño disponibles para Oracle Database 11*g* y mejora la funcionalidad existente en la base de datos Oracle. Así, todos los desarrolladores se beneficiarán utilizando la versión más reciente de ODP.NET ya sea que implementen nuevas aplicaciones de la base de datos o mejoren las aplicaciones existentes. Algunas de estas características exclusivas de Oracle incluyen el ajuste de desempeño para el acceso a los datos; la notificación de cambios en la base de datos; el grupo de conexión RAC; el soporte XML; el soporte de los tipos de datos nativos de Oracle; y muchas otras características, las cuales serán descriptas más adelante en este informe.

Desempeño

Algunos de los diferenciadores clave de ODP.NET respecto de otros proveedores son las distintas opciones de ajuste y desempeño listas para usar. Las numerosas optimizaciones garantizan automáticamente el rápido acceso .NET a las fuentes de datos de Oracle sin requerir ninguna codificación específica de desempeño. Asimismo, ODP.NET presenta muchos parámetros ajustables para la recuperación de datos específicos y los escenarios de actualización de datos. Muchas de estas optimizaciones fueron desarrolladas para recuperar y manipular históricamente grandes tipos de datos, como LOBs y REF Cursors.

ODP.NET incluye numerosas características para el ajuste de desempeño a fin de optimizar la recuperación de datos y los cambios de datos. Algunas de estas opciones de ajuste incluyen el grupo de conexiones y el caché de sentencias; la utilización de tipos de datos LOB; y la implementación de matrices PL/SQL asociativas. ODP.NET de 64 bits se soporta con Windows x64 y Windows Itanium.

Grupo de Conexiones y Caché de Sentencias

Una de las optimizaciones de desempeño más utilizadas es el grupo de conexiones, el cual es esencial para las aplicaciones con grandes cantidades de usuarios que se conectan y desconectan de la base de datos. ODP.NET crea un grupo de conexiones con parámetros ajustables que incluyen el ciclo de vida de conexión y su desconexión, tamaños mínimos y máximos de los grupos, y la cantidad de conexiones del grupo para aumentarlo o disminuirlo. Estos parámetros proporcionan a los desarrolladores mayor control en cuanto a cómo su aplicación puede manejar las grandes poblaciones de usuarios y los cambios en esas poblaciones a través del tiempo. Esto conduce a un mejor tiempo de respuesta de la aplicación y una mejor la calidad de servicio para los usuarios finales.

Si una consulta particular o sentencia PL/SQL se ejecuta múltiples veces, ODP.NET puede utilizar el caché de sentencias para acelerar la ejecución de la sentencia. Al colocar en caché el cursor del servidor creado durante el ejecución de la sentencia inicial, el caché de sentencias elimina la necesidad de volver a analizar cada sentencia antes de las próximas ejecuciones. Cada ejecución posterior de sentencias vuelve a utilizar la información analizada guardada, y luego ejecuta la sentencia. Los datos resultantes no se colocan en caché, solo la información de sentencia analizada. ODP.NET aún recuperará

los datos más recientes del servidor de base de datos. El caché de sentencias simplemente permite que estas consultas se ejecuten con mayor rapidez.

Al implementar el caché de sentencias, las sentencias SQL o PL/SQL deberían utilizar parámetros en lugar de valores literales. Al hacer esto se aprovecha el caché de sentencias ya que la información analizada de las sentencias parametrizadas pueden volver a utilizarse incluso si el valor de los parámetros cambia en futuras ejecuciones. Si se utilizaran, en cambio, valores lineales y estos valores lineales cambiarían, la información analizada no podría reutilizarse y la base de datos debería analizar las sentencias nuevamente.

Por defecto, ODP.NET colocará en caché las diez sentencias ejecutadas. La cantidad de sentencias a ser colocadas en caché y cuáles sentencias serán colocadas en caché puede ser configurado en el nivel de aplicaciones o en el nivel de la máquina.

Control de Tamaño de los Datos Buscados

Para ajustar el rendimiento de la recuperación de datos, ODP.NET puede especificar la cantidad de datos que se mostrarán en cada recorrido completo de la base de datos. Muchas veces, un desarrollador puede no necesitar recuperar los datos consultados de una sola vez. El usuario final puede consumir parte de los datos durante un período determinado.

La búsqueda de datos de consulta puede organizarse en distintas partes definidas por el desarrollador a través de dos propiedades de ODP.NET OracleCommand: *FetchSize* y *RowSize*. FetchSize informa a ODP.NET qué cantidad de datos recuperar por recorrido de la base de datos. RowSize indica la extensión de cada fila de datos. RowSize es una propiedad de solo lectura que se establece después de que se ejecuta la sentencia. Si un desarrollador desea buscar diez filas de datos por recorrido de la base de datos, todo lo que se necesita es establecer un FetchSize igual a diez multiplicado por el RowSize. Lo sorprendente de RowSize es que su valor puede determinarse en el tiempo de ejecución. Consecuentemente, si se produce un cambio de esquema o un cambio de consulta en el futuro, no hay necesidad de modificar el código para garantizar la búsqueda de diez filas de datos por cada recorrido.

Optimización de Datos LOB

Los tipos de datos LOB ofrecen una característica similar para el ajuste del tamaño de búsqueda. Estos tipos de datos son utilizados para almacenar imágenes y documentos, que algunas veces pueden ocupar varios gigabytes. Para las aplicaciones LOB, el desempeño es a menudo un tema clave debido al posible tamaño de los datos LOB y debido a cómo se consumen los datos LOB. Enviar gigabytes de datos entre el servidor y el cliente puede atascar una red, a menos que la recuperación de datos se maneje de manera inteligente.

Con ODP.NET, los desarrolladores pueden especificar cómo deben recuperarse los datos LOB. Al momento de ejecutar una consulta LOB, los desarrolladores pueden elegir buscar todos los datos LOB inmediatamente o retrasar la búsqueda LOB hasta que el usuario intente leer los datos. El usuario final puede no necesitar leer los datos directamente después de realizar la consulta. Si los desarrolladores eligen retrasar la búsqueda LOB, entonces pueden especificar qué cantidad de datos recuperar para cada llamada de lectura LOB. Si el usuario final solo necesita leer 10KB de datos a la vez, los desarrolladores pueden recuperar solo 10KB de datos para cada lectura LOB. Esto optimiza la forma en que se utilizan los recursos de red entre el servidor y el cliente.

Asimismo, los desarrolladores ODP.NET pueden recuperar una parte de LOB por medio de un acceso aleatorio. Tal vez, el usuario final pueda solo necesitar los últimos 100MB de datos de una LOB de 1GB. Los desarrolladores pueden ajustar la recuperación LOB para buscar solamente los últimos 100MB sin mostrar al cliente los datos de los primeros 900MB. Estas opciones de ajuste brindan a los desarrolladores .NET la flexibilidad para crear aplicaciones con un mejor desempeño.

Las nuevas características de Oracle Database 11*g* ya están disponibles para mejorar el desempeño de ODP.NET. Estas características incluyen un caché de resultados cliente, búsquedas LOB más rápidas y desempeño más rápido con el caché de sentencias.

Debido a que los datos LOB a menudo pueden ser extensos, por defecto, la búsqueda de datos LOB se demora luego de ejecutarse una consulta. Cuando se recupera una gran cantidad de LOBs, este comportamiento es óptimo para evitar la sobrecarga de la red con datos LOB entregados al cliente. No obstante, para las LOB pequeñas, este comportamiento puede ser lento y provocar más recorridos de la base de datos que lo necesario.

Para permitir la búsqueda inmediata de una pequeña cantidad de datos LOB, ODP.NET presenta la propiedad InitialLOBFetchSize en las clases OracleCommand y OracleDataReader. Si InitialLOBFetchSize se establece con un valor mayor a cero, los datos LOB iniciales de todas las LOB consultadas se buscan en un solo recorrido para una cantidad de caracteres o bytes especificada en esta propiedad. Por ejemplo, si InitialLOBFetchSize se estableciera en 10 KB, entonces los primeros 10 KB de todas las LOB seleccionadas serían recuperados para el cliente en un solo recorrido. Esto puede acelerar significativamente las aplicaciones que consumen grandes cantidades de pequeñas LOB.

Datos de Matriz

Una de las características exclusivas de ODP.NET es la capacidad de trasladar matrices entre la base de datos y el Entorno .NET. Las matrices facilitan el hecho de compartir grandes grupos de datos del mismo tipo entre la base de datos y el cliente. ODP.NET utiliza matrices PL/SQL asociativas en la base de datos para transmitir los datos dentro y fuera de las matrices .NET.

Entorno .NET de 64 bits

Con el Entorno .NET de 64 bits, los desarrolladores .NET tienen acceso a los sistemas de hardware más escalables y con un alto desempeño. Tienen una opción entre los procesadores AMD64 e Intel EM64T para Windows x64 y los procesadores Itanium para Windows Itanium. Los sistemas de 64 bits tienen la capacidad de abarcar directamente una mayor cantidad de memoria que los sistemas de 32 bits. Ellos incluyen componentes optimizados de hardware para informática de alto desempeño. A partir de la versión 10.2.0.3, ODP.NET soporta ambos entornos .NET de 64 bits con un *driver* de acceso a datos nativo de 64 bits para cada plataforma. Los desarrolladores pueden ahora implementar su nivel medio ODP.NET como una aplicación de 64 bits para aprovechar un hardware más escalable.

Desempeño – Novedades en ODP.NET 11*g*

Oracle Database 11*g* presenta nuevas optimizaciones de desempeño, muchas de las cuales pueden utilizar los desarrolladores de aplicaciones .NET sin realizar ningún cambio en sus actuales códigos cliente. Estas nuevas características incluyen un caché de resultados cliente, recuperaciones LOB más rápidas y caché optimizado de sentencias.

Caché de Resultados Cliente

Con el cliente y servidor Oracle Database 11*g*, las aplicaciones ODP.NET pueden utilizar el caché de resultados cliente de Oracle para mejorar los tiempos de respuesta de

las consultas ejecutadas repetidas veces. Esta característica activa el caché del lado del cliente respecto de los grupos de resultados de las consultas SQL en la memoria. El caché de resultados cliente es completamente transparente a las aplicaciones ODP.NET, y su caché de datos de resultados se mantiene en concordancia automática con cualquier cambio del lado del servidor de base de datos o con cualquier sesión que pudiera alterar los resultados.

Las aplicaciones .NET que invocan la misma consulta varias veces tienen un mejor desempeño ya que los resultados de las consultas se recuperan localmente. El procesamiento cliente local es más rápido que realizar varios recorridos hacia la base de datos para volver a realizar una consulta y buscar los resultados. Si las aplicaciones frecuentemente ejecutan las mismas consultas, experimentarán una mejora significativa en el desempeño cuando sus resultados se coloquen en el caché cliente, así como una reducción en la carga del servidor de base de datos.

En el servidor de base de datos, el caché cliente reduce la carga del tráfico de red y del CPU servidor que se habría consumido para el procesamiento y la devolución de los resultados de consulta, mejorando así la escalabilidad del servidor. Las sentencias ODP.NET de múltiples sesiones pueden concordar con el mismo grupo de resultados en caché en la memoria del proceso cliente si tienen parámetros de sesión, valores de enlace, texto SQL y esquemas similares. Esto implica que múltiples usuarios ODP.NET tienen acceso al mismo caché de resultados, lo cual minimiza las redundancias de caché y ahorra memoria.

Un desafío constante para los caches del cliente es mantener los datos en sincronización con los cambios en los datos del servidor. Al utilizar la notificación de cambios de la base de datos, los clientes ODP.NET reciben un alerta cuando se modifican los datos en el servidor, incluso si no hay conexión activa de vuelta hacia la base de datos. Esto permite al cliente garantizar que su caché de datos permanezca en sincronización con la base de datos.

Debido a que el caché cliente mantiene una concordancia automática con los datos del servidor de base de datos, los desarrolladores no necesitan escribir el código para garantizar que el caché y el servidor están sincronizados. Si ocurre algún cambio en el servidor que pueda invalidar los datos en el caché cliente, el cliente Oracle automáticamente invalidará el caché y lo actualizará la próxima vez que se realice la consulta.

Búsquedas LOB Más Rápidas

ODP.NET 11*g* mejora el desempeño de recuperación de LOB pequeñas al reducir la cantidad de recorridos a la base de datos necesarios para realizar búsquedas previas sobre datos LOB, extensiones y tamaño de partes. Esta mejora está disponible a partir de Oracle Database 11*g* para ser utilizada ya sea con LOBs tradicionales o SecureFiles. Esta mejora es transparente para el desarrollador y puede utilizarse como cualquier otro tipo de datos LOB sin ningún cambio de código sobre el código ODP.NET LOB existente.

Caché de Sentencias Optimizado

ODP.NET 11*g* mejora la actual infraestructura del caché de sentencias para el contexto de parámetros ODP.NET en caché. Esta mejora trabaja con cualquier versión del servidor de base de datos Oracle soportada. Los desarrolladores .NET experimentarán una mejora de desempeño al ejecutar consultas de sentencias en caché. Esta mejora es transparente para los desarrolladores, y no requiere cambios de código.

Notificación de Cambios en la Base de Datos

La notificación de cambios en la base de datos permite a las aplicaciones cliente recibir notificaciones cuando se realizan cambios DML o DDL a un objeto de interés de la base de datos, incluso cuando un cliente ya no tiene una conexión con el servidor de base de datos. Los desarrolladores .NET ahora pueden colocar sus datos en caché en el nivel medio sin tener que preocuparse por los datos en caché que pierden sincronización con la base de datos. Si ocurre un cambio en uno de estos objetos de datos en caché o en las

filas de datos, entonces ODP.NET recibirá una notificación de la base de datos. Esta característica puede utilizarse en el Entorno .NET 1.x, 2.0 y en versiones posteriores.

Para utilizar la notificación de cambios, la aplicación cliente registra una consulta con la base de datos. Cuando una consulta tiene dependencias en los objetos de base de datos subyacentes y se produce un cambio en un objeto dependiente, la base de datos emite una notificación de cambio a la aplicación cliente. La notificación solo puede contener metadatos de los datos u objetos que cambiaron; no contiene los datos cambiados. Los desarrolladores .NET pueden crear un administrador de eventos cliente para reemitir la consulta registrada con el fin de obtener los datos cambiados.

La notificación de cambio de la base de datos es particularmente útil para las aplicaciones que utilizan los resultados en caché. Tradicionalmente, el caché de datos es en sí mismo efectivo para mejorar la escalabilidad de las aplicaciones al permitir el rápido acceso a los datos sin realizar costosos recorridos a la base de datos. No obstante, esta escalabilidad se presenta con un compromiso, debido a que ya no es una garantía que los datos concuerden con el servidor de base de datos después de la primera consulta. Así, los datos en el caché cliente corren el riesgo de perder vigencia.

La notificación de cambios en la base de datos resuelve el problema de la pérdida de vigencia de los datos en caché. A pesar de que la notificación de cambios en la base de datos es similar a un *trigger* en cuanto a que responde a un evento particular, un *trigger* actúa inmediatamente, mientras que una notificación de base de datos es solo un alerta, no una acción. Determinar qué acción tomar, si existe alguna, y cuándo tomarla, depende de la aplicación. La aplicación puede actualizar inmediatamente los objetos estancados, posponer la actualización o ignorar la notificación. Cada aplicación .NET puede responder de distintas maneras a un cambio particular en la base de datos. Asimismo, a medida que se agregan aplicaciones adicionales a la base de datos, a menudo es más fácil modificar el administrador de eventos de la aplicación que modificar un *trigger* de base de datos. Modificar el *trigger* requiere volver a probar cómo trabajan las aplicaciones existentes con el nuevo código *trigger* de la base de datos, mientras que modificar solo la nueva aplicación .NET aísla mejor los límites de prueba.

Las aplicaciones web a menudo colocan en caché una variedad de datos, y no todos deben ser actualizados en tiempo real. Por ejemplo, un pronóstico de tiempo solo puede actualizarse periódicamente. Los usuarios finales no deben consultar la base de datos cada vez que se visita la página web. Como muchas personas solicitarán los mismos datos, la escalabilidad y el desempeño de la aplicación se mejoran enormemente al colocar en caché los resultados y recuperar los datos del caché. En algún momento, el pronóstico de tiempo se actualiza, y el caché debe actualizarse también. Esto puede realizarse en el momento que cambia el pronóstico de tiempo en el servidor de base de datos.

Recibir notificaciones de cambio de base de datos requiere que el administrador de base de datos otorgue el privilegio CHANGE NOTIFICATION al usuario de aplicaciones. Después de conectarse a la base de datos, los usuarios .NET pueden registrar sus consultas específicas para una notificación de cambio. El desarrollador crea un administrador de eventos del cliente para indicar qué deberían hacer las aplicaciones al momento de recibir una notificación de cambio de base de datos. Generalmente, el administrador de eventos volverá a consultar el servidor de base de datos y actualizará el caché.

Las siguientes clases ODP.NET se utilizan al crear aplicaciones para la notificación de cambios:

- OracleDependency – Crea una dependencia entre una aplicación y una base de datos Oracle. Permite a la aplicación recibir información de un cambio de datos (por ejemplo, una sentencia UPDATE), un cambio de esquema (por ejemplo, ALTER TABLE), o un evento global (por ejemplo, un corte de servicio de la base de datos). El administrador de eventos OnChange en esta clase ofrece al cliente la lógica para saber qué hacer después de recibir la notificación.
- OracleNotificationEventArgs – Brinda todos los detalles del evento cuando ocurre una notificación de cambio.
- OracleNotificationRequest – Especifica las características de un pedido de notificación y su notificación, como el valor de intervalo del registro de notificación.

Las notificaciones de cambio de la base de datos Oracle tienen varias características que no están disponibles en el Servidor SQL. Oracle soporta todos los tipos de uniones, mientras que el Servidor SQL no soporta las consultas que contienen uniones externas o automáticas. El Servidor SQL no soporta las notificaciones para las sentencias que utilizan visualizaciones, mientras que la notificación de cambio de la base de datos Oracle soporta las visualizaciones con la excepción de las visualizaciones fijas (por ejemplo, las tablas V$) y las vistas materializadas. Las notificaciones del Servidor SQL también requieren referencias de columnas explícitas, mientras que las notificaciones de la Base de Datos Oracle soportan tanto las referencias de columnas explícitas como las de SELECT *
.

ODP.NET tiene dos características de grupo de conexiones para las bases de datos RAC: 1) Las conexiones ODP.NET equilibran automáticamente la carga en todos los nodos existentes de acuerdo con las métricas de cluster en tiempo real y 2) Las conexiones difíciles de base de datos son automáticamente eliminadas del grupo de conexiones.

Los administradores de notificaciones del Servidor SQL no son continuos. Cuando se publica una notificación del Servidor SQL, su administrador de notificaciones es eliminado de la base de datos. Si el administrador de notificaciones aún es necesario, la aplicación debe registrar un nuevo administrador de notificaciones. La notificación de cambios de la base de datos Oracle brinda la opción de continuar con el registro incluso después de repetidos cambios. Esto es posible al configurar OracleNotificationRequest.IsNotifiedOnce como falso.

Real Application Clusters (RAC)

RAC es una base de datos en cluster con una arquitectura compartida que supera las limitaciones de los enfoques tradicionales de bases de datos federadas y no compartidas. La base de datos RAC en cluster se aloja en múltiples nodos informáticos que permiten una alta disponibilidad y mejor escalabilidad que un simple servidor. Debido a que no se requieren cambios especiales de aplicaciones .NET ni de hardware, RAC puede crearse con hardware de commodity como *back end* para las aplicaciones existentes sin requerir ningún cambio de codificación.

ODP.NET siempre ha soportado el acceso a los datos para RAC de manera transparente. Para mejorar la administración de conexión ODP.NET basada en la información de carga de trabajo de la base de datos en tiempo real, Oracle introdujo dos propiedades del grupo de conexión para ODP.NET 10.2. La primera característica, el balanceo de la carga de conexión en tiempo de ejecución, mejora la carga de trabajo del balanceo de carga en todas las instancias RAC, especialmente después de ser agregadas o eliminadas de un cluster. La segunda característica, el failover de conexión rápida, elimina automáticamente las conexiones RAC afectadas del grupo de conexiones.

Con el balanceo de carga de la conexión en tiempo de ejecución, el modo de asignación de las conexiones ODP.NET se basa en el objetivo de servicio durante el tiempo de

ejecución y en el asesor de balanceo de carga de la base de datos. El balanceo de carga distribuye el trabajo en todas las instancias de base de datos RAC disponibles.

En general, las conexiones no se crean muy frecuentemente y tienen una larga duración. El trabajo ingresa al sistema con mucha frecuencia, utiliza estas conexiones y tiene una duración relativamente corta. El asesor del balanceo de carga indica qué instancias RAC ODP.NET deberían asignarse a los trabajos entrantes a fin de obtener el nivel óptimo de calidad de servicio. El balanceo de carga minimiza la necesidad de reasignar luego el trabajo a una instancia diferente para garantizar que los trabajos existentes se completen rápidamente.

Las métricas por las cuales este trabajo se distribuye en todas las instancias están determinadas por el objetivo de servicio. El administrador de base de datos establece el objetivo de servicio por tiempo de servicio o por rendimiento.

La métrica del tiempo de servicio se basa en la rapidez con la cual la base de datos puede completar las tareas, esencialmente su tiempo de respuesta. Los datos del asesor de balanceo de carga se basan en el tiempo transcurrido para el trabajo realizado en el servicio, así como el ancho de banda disponible para el servicio. El tiempo de servicio es principalmente útil para las aplicaciones de base de datos en donde las tareas se completan en distintos niveles, como en un sistema de compras por internet.

Asimismo, la métrica de rendimiento rastrea la rapidez con la que se completan las tareas de la base de datos, asignando más trabajo a los nodos en dónde las tareas se completan con mayor rapidez. A diferencia del tiempo de servicio, la métrica de rendimiento intenta medir la eficacia de cada instancia. Mide la cantidad de recursos disponible del procesador para realizar tareas adicionales. Cuando se genera una nueva operación, ésta puede ser direccionada al nodo con mayor tiempo disponible de procesador. Esta métrica se aplica mejor a tareas relativamente uniformes, como ejecutar procesos *batch* homogéneos.

Al utilizar estos objetivos de servicio, el sistema genera un feedback. El trabajo es ruteado para brindar el mejor tiempo de servicio globalmente y responder positivamente a las cambiantes condiciones del sistema. Cuando un nodo RAC se incorpora o elimina del cluster, el balanceo de carga permite trabajar para una distribución más rápida y uniforme en todos los nodos a fin de justificar el cambio del sistema. Los usuarios finales luego experimentarán menos interrupciones o servicios lentos. En un estado estable, el sistema se acerca al equilibrio con un mejor rendimiento en todas las instancias RAC.

Para utilizar esta característica en ODP.NET, la aplicación debe activar el grupo de conexiones y establecer el parámetro del grupo de conexiones, Balanceo de Carga, en verdadero.

La segunda característica configurable para el grupo de conexiones RAC, el failover de rápida conexión, permite a ODP.NET liberar automáticamente los recursos relacionados con las conexiones separadas que fueron causadas por un nodo, miembro del servicio o servicio RAC inactivo. Sin esta característica, si un nodo RAC falla, el grupo de conexiones retendrá los recursos de conexión que ya no son válidos. Los usuarios finales pueden intentar utilizar estas conexiones separadas del grupo. Sin una manera de identificar estas conexiones, los administradores tendrían que volver a establecer todas las conexiones ODP.NET en el grupo cada vez que falla alguna parte del cluster RAC.

Estas conexiones ODP.NET separadas se eliminan automáticamente sin requerir ninguna intervención del administrador durante el tiempo de ejecución. Para permitir esta característica ODP.NET, utilice el grupo de conexiones y establezca el parámetro del grupo de conexión, HA Events, en verdadero.

Características XML

ODP.NET brinda soporte para Oracle XML DB e interopera con System.XML. ODP.NET puede trabajar con XML basado en esquemas y no basado en esquemas con soporte para tipos de datos Oracle XML nativos utilizando OracleXMLType y OracleXMLStream.

Debido a que XML es un lenguaje popular para la integración de datos y servicios web, muchos programadores .NET lo incorporan como parte integral de sus aplicaciones. XML es parte clave tanto de la base de datos Oracle como del entorno .NET. ODP.NET permite a los desarrolladores explotar estas dos tecnologías conjuntamente: Oracle XML DB y los servicios .NET System.XML.

XML DB representa la tecnología de Oracle para la recuperación y el almacenamiento XML nativo y de alto desempeño que se encuentra disponible dentro del servidor de base de datos. Ofrece la capacidad exclusiva de administrar y almacenar tanto los datos estructurados como no estructurados con un modelo de datos W3C XML estándar. XML DB brinda transparencia y una capacidad de intercambio completa entre las metáforas XML y SQL. ODP.NET ofrece a los clientes .NET la funcionalidad XML DB, permitiendo a los desarrolladores compartir y realizar cambios a XML entre la base de datos y .NET. Este soporte se extiende a XML basado en esquemas y no basado en esquemas a fin de brindar flexibilidad para distintos requisitos de aplicaciones. Asimismo, ODP.NET contiene dos tipos de datos XML nativos, OracleXMLType y OracleXMLStream para una mejor administración de datos XML cliente. Estas características permiten una administración más fácil entre los servicios Oracle XML DB y Microsoft System.XML.

ODP.NET soporta las características de ADO.NET 2.0, con inclusión de proveedores de fábrica, creador de cadenas de conexión, APIs para reconocimiento de esquemas y actualizaciones *batch DataAdapter*.

System.XML es un grupo de interfaces para manipular los grupos de datos XML de los proveedores de datos .NET. ODP.NET interopera con las interfaces de programación de System.XML, alimentando los datos por medio de las interfaces ODP.NET DataAdapter. Una de las diferencias clave entre XML DB y System.XML es que el primero brinda servicios XML en el lugar que residen los datos, en el servidor de base de datos; el último manipula XML del lado del cliente. Como tal, ODP.NET ofrece una opción más amplia para que los programadores tomen la tecnología XML que mejor se ajuste a los requerimientos del proyecto.

Con ODP.NET, los datos relacionales y de objeto relacional de la base de datos Oracle pueden ser accedidos como XML y proporcionarse en un entorno Microsoft .NET. Los cambios en XML cliente pueden realizarse y guardarse en el servidor como XML o datos relacionales. ODP.NET incluye el soporte para OracleXMLType basado en esquemas, así como XML no basado en esquemas.

ODP.NET soporta las características de ADO.NET 2.0, con inclusión de proveedores de fábrica, creador de cadenas de conexión, APIs para reconocimiento de esquemas y actualizaciones *batch DataAdapter*.

ADO.NET 2.0

A partir de la versión 10.2.0.2, ODP.NET brinda soporte para ADO.NET 2.0., que introduce un nuevo nivel de abstracción para el Nivel de Acceso a Datos (DAL). En lugar de utilizar clases de proveedores específicos que implementen una interface genérica, ADO.NET 2.0 ofrece las clases DbCommon, las cuales son heredadas de los espacios de nombre System.Data.Common y permiten a los desarrolladores utilizar las clases de fábrica. Utilizar las clases de fábrica del proveedor de base de datos es una manera sencilla de crear un solo grupo de código genérico de acceso a datos en una base de datos.

La clase OracleClientFactory soporta la creación de todas las clases en System.Data.Common. Debido a que estas clases concretas se heredan de las clases de base abstracta DbCommon, los desarrolladores pueden escribir códigos DAL genéricos utilizando nombres de objetos de clase base de DbCommon. Existen algunas áreas de DAL que continúan siendo datos de origen específico, con inclusión de la cadena de conexión, SQL, y las llamadas a procedimientos almacenados.

Una incorporación clave en ADO.NET 2.0 es una mejor administración de la cadena de conexión. La clase DbConnectionStringBuilder tiene un diccionario que mapea los nombres de parámetros genéricos hasta los nombres de los parámetros específicos del proveedor. La clase OracleConnectionStringBuilder se hereda de la clase genérica DbConnectionStringBuilder, y la extiende, para exponer las propiedades de la cadena de conexión específica de Oracle. Los programadores pueden utilizar OracleConnectionStringBuilder dinámicamente para establecer los parámetros de la cadena de conexión en tiempo de ejecución y/o obtener los parámetros de la cadena de conexión del archivo de configuración de aplicaciones. Esta característica permite la administración más fácil de las cadenas de conexión, especialmente aquellas con una gran cantidad de atributos.

ODP.NET soporta tipos nativos de base de datos Oracle, como REF Cursors, dentro del entorno .NET. Estos tipos de datos nativos pueden proporcionar un mejor desempeño y más flexibilidad para la recuperación de datos.

Las API de reconocimiento para el esquema ADO.NET 2.0 ofrecen una manera genérica de recuperar metadatos desde cualquier fuente de datos. Los desarrolladores pueden obtener los metadatos de Oracle al utilizar una llamada del método OracleConnection.GetSchema. Existen cinco tipos de metadatos comunes que pueden ser expuestos: MetaDataCollections, Restrictions, DataSourceInformation, DataTypes, y ReservedWords. Asimismo, existe información adicional de fuentes de datos específicas de ODP.NET que puede ser recuperada. Con un reconocimiento de esquema, los desarrolladores cuentan con una forma simple de recuperar los metadatos desde cualquier fuente de datos Oracle.

Al trabajar con grandes .NET DataSets, los desarrolladores querrán minimizar la cantidad de recorridos a la base de datos requeridos para realizar las actualizaciones. Al utilizar la propiedad OracleDataAdapter.UpdateBatchSize, los programadores pueden especificar la cantidad de filas a actualizar por recorrido. Para actualizar todas las filas modificadas en un solo recorrido, UpdateBatchSize puede establecerse en cero. El soporte de este procesamiento de lotes permite a DataSet actualizarse para ser implementado de manera más rápida.

Tipos Nativos de Oracle

Microsoft ha incorporado un grupo de tipos de datos unificados entre los diferentes lenguajes de programación .NET. Con ODP.NET, los programadores .NET tienen acceso a los tipos de datos .NET así como los tipos de datos de Oracle. Los tipos de datos de Oracle pueden manipularse totalmente dentro de una aplicación .NET e interoperar con los tipos de datos .NET. Los tipos nativos de Oracle ofrecen funcionalidad avanzada para almacenar y manipular las estructuras de datos de la base de datos, como XML, los grupos de resultados, imágenes, texto, o documentos de Microsoft Office. Incluso con *scalar types*, como OracleDecimal, el equivalente al tipo decimal .NET, los tipos de Oracle brindan funcionalidad adicional. En el ejemplo de OracleDecimal, este tipo de datos ofrece un mayor nivel de precisión de 38 dígitos que el decimal .NET con una precisión de 28 dígitos.

ODP.NET soporta el espectro de los tipos avanzados de Oracle dentro del entorno .NET, con inclusión de REF Cursors, XMLType, LOBs (CLOBs, BLOBs, NCLOBs), BFILEs, LONGs, RAWs, LONG RAWs y tipos de datos N. Una de las limitaciones

para utilizar proveedores de datos .NET de terceros es que los usuarios pueden ser limitados en cuanto a la funcionalidad del tipo de datos Oracle. Por ejemplo, en ODP.NET, múltiples grupos de resultados mostrados como parámetros de salida de REF Cursor de un procedimiento almacenado pueden accederse de manera arbitraria. Uno puede leer los resultados del segundo REF Cursor sin recuperar los resultados del primero. Con otros proveedores .NET, se puede acceder a los datos en forma lineal en donde el primer grupo de resultados debe ser recuperado antes de accederse al segundo. Esto afectará negativamente el desempeño.

Otras características importantes de ODP.NET incluyen poder utilizar cualquier tipo de PL/SQL; transacciones locales y distribuidas; y aplicaciones internacionalizadas con Unicode.

A partir de ADO.NET 2.0, los tipos de datos Oracle pueden almacenarse nativamente dentro de.NET DataSet. Anteriormente, solo los tipos de datos .NET podrían utilizarse en DataSet. Al establecer OracleDataAdapter.ReturnProviderSpecificTypes como verdadero, DataSet se completará con los tipos de datos específicos de ODP.NET cuando se convoca a OracleDataApdater.Fill. Esta característica puede mejorar el desempeño porque le ahorra a .NET la tarea de realizar conversiones del tipo de datos.

Otras Características Importantes

ODP.NET se expone a muchas otras características de la base de datos Oracle, con inclusión del soporte de Unicode, transacciones y PL/SQL.

Los usuarios de ODP.NET pueden ejecutar plenamente las funciones y los procedimientos PL/SQL almacenados en la base de datos. PL/SQL puede estar empaquetado o no empaquetado, o incluso existir como PL/SQL anónimo dentro de .NET. En ODP.NET, el PL/SQL anónimo generalmente se implementa para realizar un proceso *batch* de un grupo de sentencias SQL y ejecutar las sentencias en el recorrido de una base de datos. Las sentencias *batch* son una técnica útil para la optimización de desempeño.

ODP.NET puede participar en aplicaciones transaccionales con la base de datos Oracle como administrador de recursos. ODP.NET implementa Microsoft Distributed Transaction Coordinator (DTC) para supervisar una transacción en un entorno Windows. Oracle Services para Microsoft Transaction Server (OraMTS) actúa como proxy entre ODP.NET, DTC y la base de datos Oracle para coordinar estas transacciones. OraMTS brinda una sólida arquitectura para que los programadores ODP.NET mantengan una alta disponibilidad y escalabilidad en sus aplicaciones transaccionales. En .NET Framework 2.0 y versiones posteriores, ODP.NET 10.2.0.3 soporta tanto las transacciones locales como las distribuidas por medio del espacio de nombre System.Transactions.

ODP.NET tiene soporte total de Unicode, de modo que los usuarios .NET pueden globalizar sus aplicaciones fácilmente en múltiples idiomas escritos. Este soporte de globalización permite a los desarrolladores crear un solo grupo de códigos para múltiples configuraciones de cultura/idiomas. La globalización de ODP.NET extrae la configuración del idioma nacional informático del cliente para desplegar información en el formato local específico. Por ejemplo, un browser establecido en japonés tendrá su moneda expresada en yen. Sin una codificación adicional, la misma aplicación puede implementarse en Alemania para mostrar la moneda en euros. Esto hace que el desarrollo de aplicaciones para múltiples escenarios sea más fácil y más rápido, ya que no se necesita una codificación adicional.

CONCLUSIÓN

En Oracle Database 11*g*, ODP.NET ofrece nuevas capacidades de desempeño para el desarrollador de base de datos Oracle, las cuales están creadas sobre características actuales de seguridad, facilidad de uso, escalabilidad y desempeño. Mientras se introducen nuevas características de la base de datos Oracle y de .NET Framework, ODP.NET sigue brindando soporte a estas nuevas características.

Para obtener más información de ODP.NET y Oracle sobre .NET en general, visite el Centro de Productos ODP.NET de OTN:
http://www.oracle.com/technology/tech/windows/odpnet/

Centro de Desarrollo .NET de OTN: http://otn.oracle.com/dotnet

Oracle Data Provider para .NET 11*g*
Octubre de 2007
Autor: Alex Keh

Oracle Corporation
Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Consultas:
Teléfono: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Copyright © 2007, Oracle. Todos los derechos reservados.
El presente documento tiene solo fines informativos y su contenido está sujeto a cambios sin que medie notificación alguna. El presente documento puede contener errores y no está sujeto a ninguna otra garantía ni condición, ya sea oral o que se encuentre implícita en la ley, con inclusión de garantías y condiciones implícitas de comerciabilidad o aptitud para un fin específico. En especial, negamos cualquier responsabilidad con respecto al presente documento, el cual no crea obligación contractual alguna, sea en forma directa o indirecta. El presente documento no podrá ser reproducido ni transmitido de ninguna forma ni por ningún medio, sea electrónico o mecánico, con ningún fin, sin que hayamos otorgado previamente nuestro consentimiento por escrito. Oracle es marca registrada de Oracle Corporation y/o sus afiliadas. Otros nombres pueden ser marcas comerciales de sus respectivos propietarios.