Oracle Unified Directory
Java basierter schlanker Directory Service

Abdi Mohammadi
Oracle Deutschland B.V. & Co. KG
Hamburg

Schlüsselworte:
Directory, LDAP, Java, Oracle Berkeley DB, Identity Store, Oracle Unified Directory, Replication, Hochverfügbarkeit, Skalierbarkeit, Oracle Unified Directory, Cloud Computing

Einleitung

Immer mehr Applikationen im Intra- bzw. Extranet benötigen eine sichere Möglichkeit, die zugreifenden Entitäten zu authentisieren bzw. die Zugriffe zu autorisieren. In den meisten Fällen wird ein LDAP Directory Server als User Repository bzw. Identity Store verwendet, welches sowohl die Nutzeridentität samt Nutzerkennung als auch Applikationsprofiles hosten kann. Bei zunehmender Anzahl von Webapplikation (Portale, Wikis, Soziale Netzwerke) und der Vielfalt an Devices (PCs, Tablets und Smart Phones) ist ein grosser Bedarf für dynamisch skalierbare und hochverfügbare LDAP Directory Services zu verzeichnen.
Während in der Vergangenheit Directory Server meistens zwecks Authentisierung und Auffinden von Nutzeridentitäten benutzt wurden und für lesende Operationen optimiert waren, häufen sich in der Gegenwart die Anzahl der Schreiboperationen. Vor allem Smartphone-Apps möchten z.B. die Koordinaten von Standorten (Location Service) im Directory Server speichern.
Oracle verfolgt mit dem Java-basierten „Oracle Unified Directory Service“ (OUD) das Ziel einen

- einfachen
- sicheren
- für moderne Multi-Thread/Multi-Core Hardware optimierten

LDAP v3 Directory Server inkl lastverteilendem Proxy zur Verfügung zu stellen. OUD zeichnet sich durch hohe Stabilität und eine hohe Performance sowohl bei Lese- als auch Schreib-Operationen aus. Mit OUD wird ein „Next Generation Directory Service“ bereitgestellt, der auf die neuen Anforderungen des Marktes für Directory-Einträgen im Milliardenbereich vorbereitet ist (über 100000 Operationen pro Sekunde) , bei entsprechender Deployment Architektur linear skaliert und eine für geschäftskritische Anwendungen wichtige Verfügbarkeit von 99.999% bietet.

Die klassischen auf der Programmiersprache C basierten Directory Servers wie ODSEE (vormals Sun Directory Server), OpenLDAP und dergleichen können in der Regel die Multithreading Fähigkeiten der Betriebssysteme und Hardware nicht so optimal nutzen, um eine lineare Skalierbarkeit zu erzielen, da C-Threads nur per Library eingebunden sind. Die 100% in Java entwickelte OUD Threads können jedoch unabhängig voneinander die vorhandenen CPU Threads/Cores optimal konsumieren und damit eine nahezu lineare Skalierung erreichen.

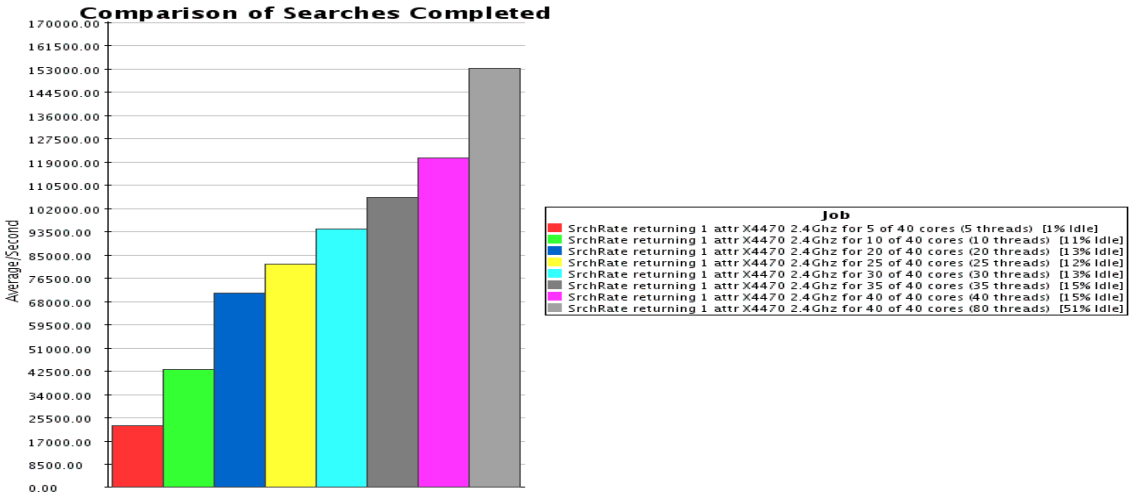


Abbildung 1: Lineare Skalierbarkeit von OUD

Auch die Cloud Services, die für Millionen für Internet Nutzern dimensioniert werden, benötigen einen hochverfügbaren und skalierbaren Directory Service, der sowohl für die Registrierung der Nutzer als auch die Überprüfung der Berechtigungen verwendet werden kann. Speziell bei Applikations- oder Mobile-Providern besteht der Bedarf für einen großen, zentralen Directory Service, der auch Anforderungen wie Mandantenfähigkeit erfüllt. Stellt beispielsweise ein Cloud Provider ein und dieselbe Applikation mehreren Mandanten zur Verfügung, oder ermöglicht ein Mobile Provider seinen Kunden ihre Telefonbücher Online zu betreiben, können die Anzahl der Einträge in der Datenbank die Millardengrenze leicht übertreffen. Ein aus kleinen Directory Servern

zusammengestellte Verbund könnte zwar technisch funktionieren, ist jedoch aus Gründen der Managebarkeit nicht akzeptabel. Ein solcher Directory Service muss für Millionen Lese- und Schreib-Zugriffe innerhalb weniger Sekunden aus dem Internet gewappnet sein.

OUD basiert auf OpenDS (http://opends.org) und läuft als eine standalone Applikation in einer Java VM (d.h. ohne Applikationsserver). Damit ist es gewährleistet, dass OUD technisch auf beliebiger Hardware und Betriebssystem Plattform laufen kann und sowohl für kleine Unternehmen als auch für große Internet Cloud Service Anbieter in Frage kommt. OUD zeichnet sich durch hohe Stabilität und eine hohe Performance sowohl bei Lese- als auch Schreibzugriffen aus. Je nach Umgebung beträgt die Steigerung der Performance bis zu einem Faktor 3 beim Lese- wie bis zu einem Faktor 5 bei Schreiboperationen verglichen zu ODSEE 11g (Oracle Directory Service Enterprise Edition 11g). OUD liefert neben klassischen LDAPv3 Services auch LDAP-Proxy und -Virtualisierungs-Dienste. Mittels des OUD Proxy Server kann jedem LDAP-Client eine spezifische und angepasste Sicht auf die Daten der Remote LDAP-Servers gezeigt werden. Der Proxy Server unterstützt sowohl eine klassische Lastverteilung (Load Balancing) als auch eine Datenverteilung (Distribution).

Sicherheit

Neben den im LDAP Serverumfeld gängigen Sicherheitsmerkmalen wie

- Verschlüsselung des Datentransports
- Verschlüsselung gespeicherter Attributwerte
- Unterstützung verschiedener Authentisierungsverfahren (z.B. Kerberos)
- Unterstützung von GSSAPI

bietet OUD

- Zugriffsschutz auf Teilebereiche des DITs (Diretcory Information Tree) in Abhängigkeit von IP-Quelladresse, Binding Parameter etc) mittels Network Groups und Quality of Service
- Schutz vor Denial of Service Angriffen

Diese Sicherheitsmechanismen werden im folgenden kurz näher erläutert.
Die „Network Groups“ definieren die erste Sicherheitsstufe für einen ankommenden LDAP Request. Abhängig von bestimmten Kriterien (IP-Adresse, Binding Parameters usw.) kann ein Request auf ein oder mehrere bestimmte Teilbäume des DIT (Directory Information Tree) beschränkt werden, für die dem Benutzer die entsprechenden Zugriffsrechte erteilt worden sind.
Die nächste Stufe der Sicherheit bilden die so genannten „Quality of Service Policies“. Hier kann bei „Request Filtering Policy“ noch granularer bestimmt werden, ob z.B. nur lesende Requests erlaubt sind oder nur bestimmte Attribute abgerufen werden dürfen. Die Filter Policies können eingesetzt werden, auch wenn nur bestimmte Teilbäume eines Naming Context nach Attributwerten durchsucht werden dürfen oder manche Teilbäume unsichtbar bleiben sollen.

Um „Denial of Service“Angriffen zu vorbeugen, werden für jede „Network Group“ entsprechende „Resource Limits“ vereinbart. Dabei kann die maximale Anzahl der Verbindungen eines LDAP Clients von individuellen IP-Adressen und die Anzahl der gleichzeitigen Operationen pro Verbindung festgelegt werden. Auch die Anzahl, die Dauer aber auch die Grösse der Antwortpakete können an dieser Stelle limitiert werden. Die Zugriffsberechtigung auf die Daten selbst werden dann über die sogenannten „ACIs“ (Access Control Instructions) sowohl global (Global ACI) als auch explizit in der Datenbank definiert. Damit kann einem Benutzer oder einer Gruppe, die Berechtigung erteilt werden, auf Directory Einträge lesend und/oder schreibend zuzugreifen.

Replikation

OUD unterstützt Multimaster Replikation mit einer beliebigen Anzahl von Mastern. Ein neu entwickeltes Replikationsprotokoll ermöglicht typischerweise die Replikation auf die anderen Systeme im Millisekundenbereich. Neu hierbei ist im Unterschied zu ODSEE ein eigener Replikationsdienst, der Bestandteil der Directory Server Instanz ist und auf einem eigenen TCP Port lauscht.

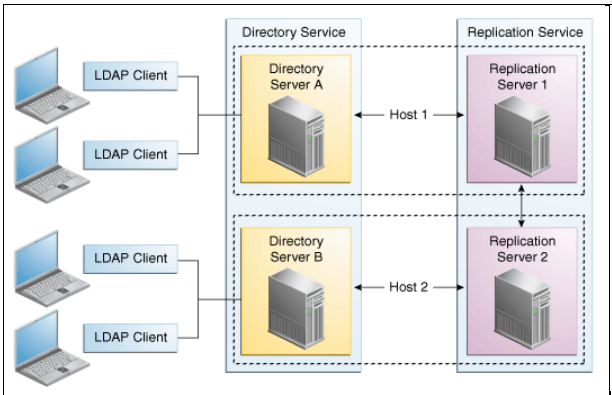


Abbildung 2: Replication Service

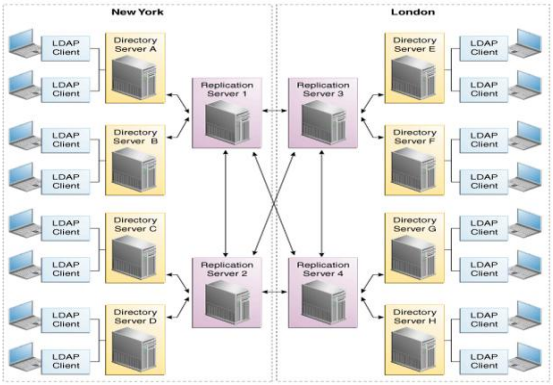


Abbildung 3: Multi Data Center

Dadurch werden zwei unterschiedliche Replikationskonfigurationen definiert. Die erste Konfiguration legt fest, an welchem Replication Server ein Directory Server sich registriert (von diesem erhält er dann auch die Änderungen). Bei der Voreinstellung von OUD ist der Replication Server lokal und innerhalb derselben Directory Server Instanz konfiguriert.
Die zweite Konfiguration betrifft das Verhältnis zwischen den Replication Servern, die gekoppelt sind und gegenseitig Änderungen in ihrer Changelog Datenbank austauschen können.
Zum Schutz von Konsistenz und Integrität, kann „Assured Replication“ eingesetzt werden, um sicher zu stellen, dass bei einer schreibenden Operationen mindestens zwei Replication Servers die Änderung erhalten haben, bevor die Operation dem LDAP Client bestätigt wird.
Eine Besonderheit gegenüber vielen anderen Produkten auf dem Markt ist Fractional Replication, wie es schon im ODSEE (vormals Sun Directory Server) zu finden war. Bei Anwendung von Fractional Replication kann eine Teilmenge der Attribute eines Eintrags repliziert werden. Dies kann aus Sicherheitsgründen notwendig sein, wenn z.B. die Kontonummer eines Benutzers auf einem Directory Server, der in einer DMZ ist, nicht sichtbar sein soll.
Während die Replikationsrollen (Master Supplier Replica, Consumer Replica, Hub) und die Replikationsvereinbarungen (Replication Agreements) im ODSEE bei jedem Directory Server explizit konfiguriert werden mussten, reduziert sich die Konfiguration der Replication Agreements auf die wenigen „Replication Servers“, die ähnlich zu den Replication Hubs bei ODSEE fungieren.
Die Directory Servers suchen sich den aktuellsten Replication Server innerhalb ihrer definierten Replikationsgruppe und binden sich an ihn. Die Replication Servers senden Updates zu allen anderen Replication Servern der Replication Domains und an die direkt gebundenen Directory Servers.
Gerade bei einer großen Anzahl von Directory Server Instanzen, die weltweit verteilt aufgesetzt werden können, wird hierdurch der Konfigurationsaufwand stark verringert. Es wird keine N:N Topologie erzeugt und die Replikationstopologie ist dadurch viel übersichtlicher.
Eine spezielle Art der Replikation wird von dem „Replication Gateway“ geleistet.

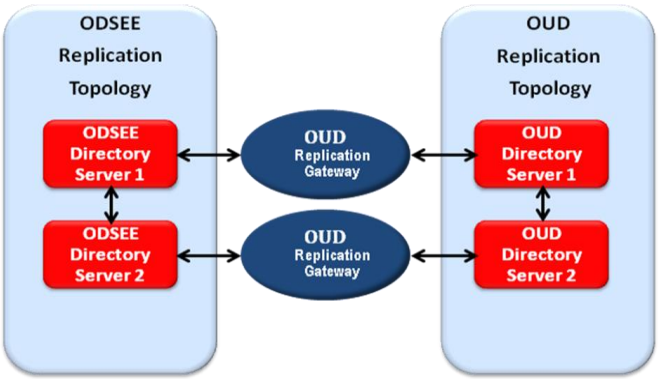


Abbildung 4: Replication Gateway

Dieses Gateway kann zwischen dem ODSEE (Oracle Directory Server Enterprise Edition 11g) und OUD synchronisieren. Dadurch ist eine Migration von bestehenden ODSEE Instanzen zu OUD einfach möglich.

Architektur

OUD ist in Java entwickelt und integriert die Java Version des Oracle Berkeley DB als lokale hierarchische Datenbank.

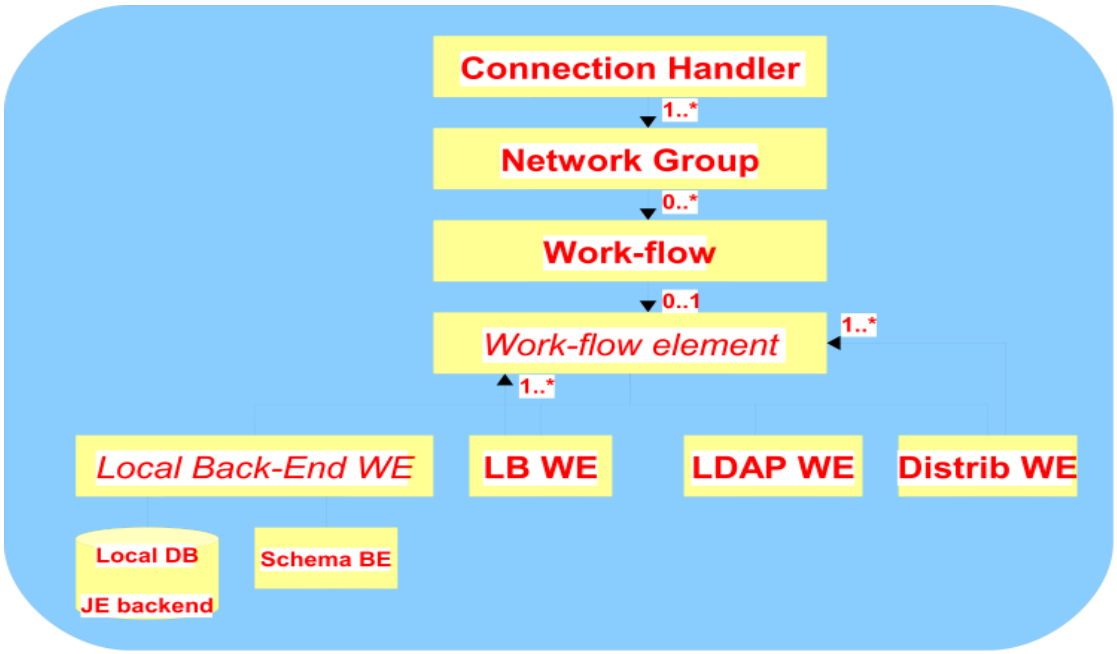
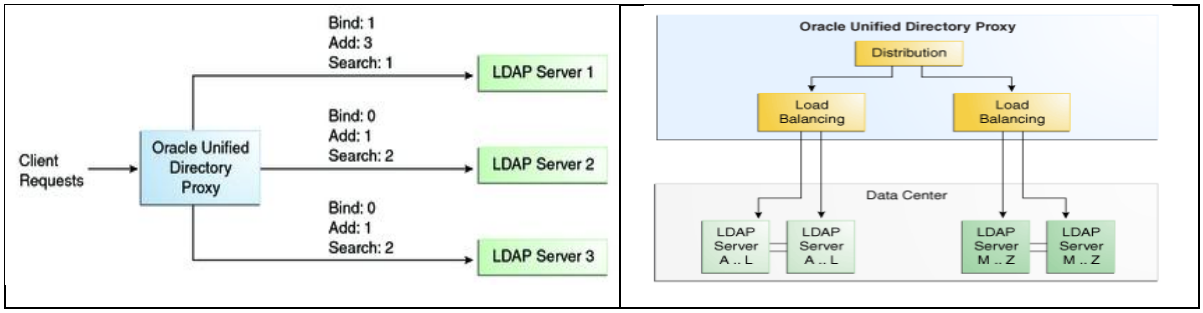


Abbildung 5: Architektur von Oracle Unified Directory

Das Front End zu den LDAP Clients bildet der Connection Handler. Er stellt die TCP-Listener bereits, um auf LDAP und LDAPS Anfragen zu reagieren. Hierbei kann die Anzahl der max. Client-Verbindungen konfiguriert werden.
Innerhalb des „Network Group“ Layers findet eine Berechtigungsprüfung des eingehenden Requests statt. Diese basiert auf Filter und Resource Policies. Abhängig von bestimmten Kriterien (Binding DN, IP-Addresse des Clients, Target DN, usw.) wird dann der Zugriff auf einen bestimmten Naming Contexts erlaubt. Ein Naming Context definiert den Root-Eintrag eines Teilbaums mit einem eigenen Datenbank Backend wie z.B. dc=oracle,dc=com oder cn=schema .
Der Naming Context kann dabei entweder auf lokale Backends (Berkeley DB) oder aber auf Remote LDAP Servers zeigen. Bei einer international agierenden Firma mit Teildatenbanken in unterschiedlichen Kontinenten ist eine Implementierung üblich, dass die Daten in drei unterschiedliche Rechenzentren in Europa, Asien und Amerika (dc=europe,dc=oracle,com dc=asia,dc=oracle,dc=com dc=america,dc=oracle,dc=com) ablegt.
Der gleiche Naming Context kann aber auch auf mehr als einen Remote Server zeigen. In so einem Fall können Load Balancing und/oder Distribution „Workflow Elemente“ eingesetzt werden, um die eingehenden Anfragen zu verteilen. Hierbei kann die Verteilung in Abhängigkeit der LDAP Operation (SEARCH, ADD, DELETE, MODIFY, COMPARE, BIND) erfolgen: Schreibzugriffe werden zur Server A1 und A2 weitergeleitet, Lesezugriffe auf B1, B2, B3 und B4 usw. Darüber hinaus kann ein Naming Context über mehrere Remote Server partitioniert werden (Distribution).
Die Architektur des OUD erlaubt, dass „Workflow Elemente“ aufeinander zeigen. D.h. es kann eine Verschachtelung von Distribution und Load Balancing Elemente konfiguriert werden.
Damit können sehr große, leistungsfähige Architekturen geschaffen werden, um hohen Anforderungen an Lese und Schreib-Operationen gerecht zu werden.

LDAP Proxy / Virtualisierung

OUD ist nicht nur ein Directory Server, sondern kann auch als ein LDAP Proxy bzw. Load Balancer eingesetzt werden. Dabei fungiert der OUD Proxy als ein „Single Point of Access“, der eine virtualisierte Sicht auf die Daten der vorhandenen Directory Server Backends dem Client zur Vefügung stellt. Dabei können DN (Distinguished Name), Attribute oder Objektklassen so abgebildet werden, wie die der LDAP Client Anwendung es erwartet.



Load Balancing kann auf LDAP Operationsebene (SEARCH, BIND, ADD usw.) definiert werden, um z.B. alle BIND Operationen an zwei spezielle Directory Server zu schicken, während Schreiboperationen an einen dedizierten Master und sämtliche SEARCH Operationen mit entsprechender Verteilungsmetrik an alle verfügbaren Directory Server geschickt wird.
Durch Partitionierung der Daten (Distribution) kann der gleiche „Naming Context“ mit Hilfe eines Algorithmus an unterschiedlichen Server geschickt werden. Die Partitionierung wird häufig dann eingesetzt, wenn entweder die Anzahl der Einträge in einer Datenbank zu gross wird, sehr hohe Anzahl der Schreiboperationen notwendig sind oder aber auch, wenn aus Verfügbarkeits- und Sicherheits-Gründen, dies erforderlich wird.
Durch „Global Index“ wird eine Möglichkeit geschaffen, in einer solchen partitionierten Umgebung, in Abhängigkeit des LDAP Search Filters das richtige Directory Server Backend zu finden, der die entsprechende Partition der Daten beherbergt.
Die globalen Indizes werden dann zwischen den Instanzen der OUD Proxy Servers repliziert.

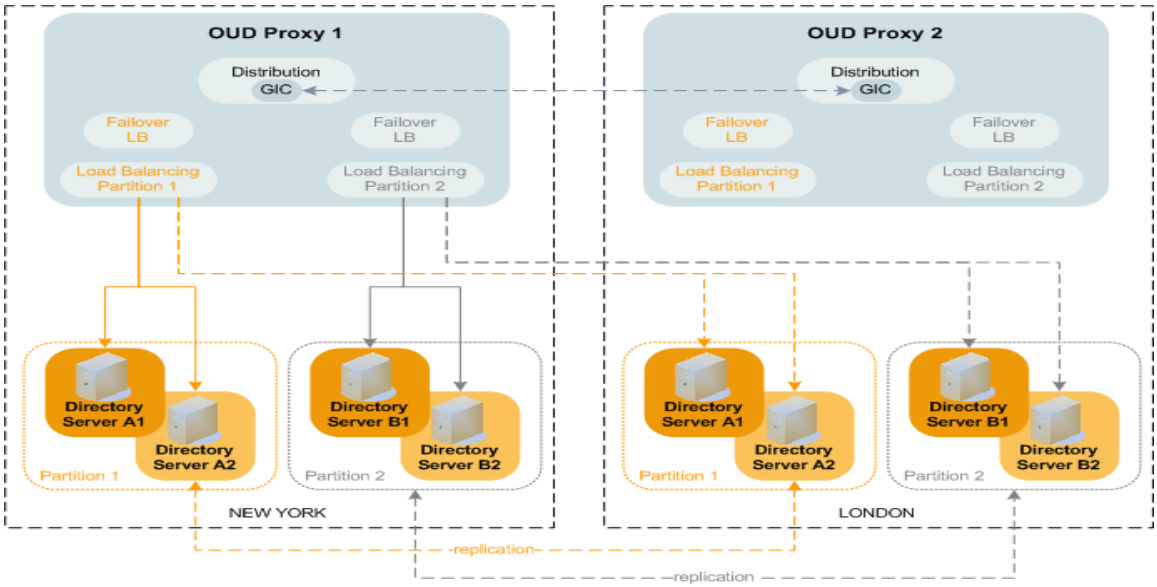


Abbildung 6: Komplexe verteilte Umgebung mit Global Index

Damit kann eine hoch verfügbare und dynamisch skalierbare aber genauso leistungsfähige Directory Service Architektur implementiert werden, die Daten zwischen verschiedenen Standorten und Kontinenten repliziert und den hohen Anforderungen einer Cloud gerecht wird.

Performance

Nachdem OUD eine reine JAVA Applikation ist, können die kompletten Vorteile von Multithread Prozessoren optimal genutzt und eingesetzt werden. Es wird eine fast lineare Skalierung der Performance sowohl bei lesenden als auch schreibenden Operationen erreicht.
Bei geeigneter Konfiguration von Heap und DB Cache aber auch bei Verwendung von Filesystem Cache können möglichst viele Disk I/O Operationen gespart werden, um eine noch bessere Performance zu erzielen.
Anwendung und Tuning von Java Garbage Collecters spielt eine erhebliche Rolle beim stabilen und performanten Einsatz von OUD.
Im Vergleich zu ODSEE, OpenLDAP oder ähnliche Directory Server Produkte kann eine bis zu 5 fache Performance Steigerung bei Lese- und bis zu 3 fache Steigerung bei Schreib-Operationen auf derselben Hardware Plattform erreicht werden. So wurden mehr als 100000 Operationen pro Sekunde auf einer einzelnen Instanz von OUD erreicht.

Fazit

Durch JAVA ist der sehr schlanke und optimierte Next Generation Directory Service „OUD“ auf unterschiedlich klein oder grossen Hardware und Betriebssystem Plattformen lauffähig.
Hohen Performance, Stabilität und Skalierbarkeit Anforderungen an Directory Service bei Telekommunikation Unternehmen oder Cloud Providers können mit geeigneter Architektur von OUD Instanzen Rechnung getragen werden.

References

http://www.oracle.com/technetwork/middleware/id-mgmt/overview/oud-433568.html
http://download.oracle.com/docs/cd/E22289_01/index.htm
http://www.oracle.com/technetwork/middleware/id-mgmt/whitepaper-oud-434007.pdf

Kontaktadresse:

Abdi Mohammadi
Oracle Deutschland B.V. & Co. KG
Kühnehöfe 5
D- 22761 Hamburg

Telefon: +49 (0) 40-89091 624
Fax: +49 (0) 40-89091 250
Abdi.mohammadi@oracle.com
Internet: www.oracle.com