Service-oriented architecture and data grids in oracle
Neeraj Bhargava

Abstract
Oracle SOA Suite is a comprehensive, hot-pluggable software suite to build, deploy and manage Service-Oriented Architectures (SOA). The components of the suite benefit from common capabilities including consistent tooling, a single deployment and management model, end-to-end security and unified metadata management.
This paper deals with a data grid infrastructure, built with clustered caching, can help you avoid "weak link" vulnerabilities that can sabotage SOA strategies. Discover why the data grid offers improved data access that can create a competitive edge, improve financial performance, and sustain customer loyalty. Service-oriented architecture (SOA) provides a means of integrating disparate applications within an enterprise, improving reuse of application logic while eliminating duplication of production environments.

Keywords: Data Services, Business Services, Business Processes, Data Grid

Introduction
The increasing rate of change in the modern business environment demands greater agility in an organization’s technology infrastructure. Service-oriented architecture (SOA) provides a means of integrating disparate applications within an enterprise, improving reuse of application logic while eliminating duplication of production environments. SOA avoids silos of disconnected information that make it difficult to service customers, meet production demands, and manage large volumes of information. In addition, it offers the promise of less interdependence between projects, and as a result, greater responsiveness to business challenges. Developing an SOA that guarantees service performance, scalable throughput, high availability, and reliability is both a critical imperative and a huge challenge for today's large enterprises. However, SOA poses its own set of challenges to the underlying middleware infrastructure[1].
A basic principle of SOA is the decoupling of applications and services. This leads to increased changes to existing applications and services, as well as more frequent creation of new ones. Supporting many more applications and services—whose requirements change almost continuously—means more hardware, more infrastructure software, and more administration. When added to IT’s mandate to meet stringent business service-level agreements, keep costs low, and implement environmentally sustainable technologies, these additional support requirements can stretch IT resources to the limit. To meet all these demands, IT needs to be able to support rapid application changes, dynamically adjust resource allocation, and increase the use of shared IT infrastructure. Traditional application architectures—typically involving islands of large enterprise software, installed or provisioned on fixed and dedicated hardware stacks—are not keeping up with the demands of this new world.
Application grid is an emerging architectural approach for middleware infrastructure that leverages existing technologies as well as new innovations. It makes infrastructure more flexible and efficient. Because applications and services rarely hit peak demand at the same time, pooling, sharing, and dynamically adjusting the allocation of hardware and infrastructure software resources with an application grid enables IT shops to be agile and efficient[2][7].
One of the important benefits of an SOA is the opening of legacy applications and their data stores to much wider use. An associated challenge, however, is that the legacy asset may not be architected in a way that supports the transactional demands resulting from that wider use. By providing a buffer against the legacy data store, a data grid can serve as a caching layer that scales linearly to extreme levels. IT does not have to re-architect the legacy asset; the data grid enables IT to offer broader access with high service levels.

The Structure of an SOA Environment
In an SOA environment, there are several types of components to consider. In order of increasing consolidation, these can be grouped into data services, business services, and business processes. Data services provide consolidated access to data. Business services contain business logic for specific, well-defined
tasks and perform business transactions via data services. Business processes coordinate multiple business services within the context of a workflow[4][11].

**SOA Overview**

[Diagram of SOA Overview]

Figure1: SOA environments typically comprise three types of components: data services, business services, and business processes.

Data within an SOA generally falls into one of two categories:
- Conversational state. The conversational state is managed by business services and processes and corresponds to currently executing operations, processes, sessions, and workflows.
- Persistent data. Persistent data is managed by data services and is usually stored in databases.

**SOA Environments Differ from Traditional User-Centric Applications**

Conversational state, such as the hypertext transfer protocol (HTTP) session state utilized by Web services, is often short-lived, rapidly modified, and repeatedly used. The life span of the data may be a matter of seconds, spanning a dozen requests, each of which may need to read or update the data. Moving from traditional user centric applications to an SOA environment means that, in addition to users, machines are now accessing services—at machine speed. This means that the “user count” increases dramatically while the average “think time” decreases to almost nothing, causing the maximum sustained request rate to far exceed the original specification. The result is that technologies that were capable of handling traditional user loads are almost inevitably crushed by the increased load associated with an SOA deployment[6][13][14].

Ensuring the reliability and integrity of conversational state is critical, but its rapid churn rate and transient nature make it particularly difficult to manage by traditional means. Using database servers is the traditional solution for scalable data services, but they cannot cost-effectively meet the throughput and latency requirements of modern large-scale SOA environments. Most in-memory solutions depend on compromises such as queued (asynchronous) updates, master/slave high availability (HA) solutions, and static partitioning to hide scalability issues, all at the cost of substantially reduced reliability and scalability. Most SOA vendors go so far as to strongly recommend avoiding state full services if at all possible, due to these scaling and performance challenges.

**SOA Demands High Data Reliability and Availability**

Making sure that services have a consistent, coherent view of data is critical to ensure reliability and availability. Transitionally consistent data services are essential for scalable, reliable data processing. Products used to manage data must have data integrity “in their genes,” supporting both optimistic and pessimistic transactions, synchronous server redundancy, and reliable configurations.

Data-management products for SOA must prioritize availability and reliability over features, because SOA adoption results in enterprise systems that are more prone to outage as the number of service-dependencies increases. This is the natural consequence of compositional complexity and represents an engineering trade-off resulting from the elimination of application silos. This risk becomes further pronounced as systems are consolidated, because service interruptions will have an increasingly greater impact on the organization.

**Clustered Caching Ensures Reliability and Availability**

Oracle Coherence is a trusted in-memory data management solution for ensuring reliability and high availability for Java-based service hosts, such as Java Platform, Enterprise Edition (Java EE) application servers. It makes sharing and managing data in a cluster as simple as on a single server. It accomplishes this by coordinating updates to the data by using cluster wide concurrency control, replicating and distributing data modifications across the cluster by using the highest-performing clustered protocol available, and delivering notifications of data modifications to any servers that request them.

Oracle Coherence, which provides replicated and distributed (partitioned) data management and caching services on top of a reliable, highly scalable peer-to-peer clustering protocol, has no SPOFs. It automatically and transparently fails over and redistributes its clustered data management services when a server becomes inoperative or is disconnected from the network. When a new server is added or when a failed server is restarted, it automatically joins the cluster and Oracle Coherence fails services back to it, transparently redistributing the cluster load. Oracle Coherence includes network-level fault-tolerance features and transparent soft-restart capabilities to enable servers to self-heal[8][10].
Figure 2: Caching is used to decouple components, yielding increased performance, throughput, and reliability.

Oracle Coherence provides a fully reliable in-memory data store for services, transparently managing server faults and making it unnecessary for the service logic to deal with complicated leasing and retry algorithms.

Oracle Coherence is well known for its singular ability to handle enormous transaction volumes (300,000+ transactions per second) for conversational state without compromising read performance or fault tolerance.

By shifting state into Oracle Coherence and using Oracle Coherence’s dynamic mesh architecture to dynamically scale data management, applications can achieve near-real-time provisioning without risking loss or abortion of requests.

Deployment Flexibility Through the Data Grid

Oracle Coherence enables capacity on demand in two key steps. First, it helps move conversational state out of the application and into the Oracle Coherence Data Grid. This enables requests to be routed to any application instance without the need for manual provisioning of data. Oracle Coherence’s mesh architecture also means that additional application instances can be started on the fly, without the need for manual repartitioning of data and with minimal delay, because application state is already prepared in the data grid. This compares admirably to products that depend on static partitioning and “buddy” replication for failover.

Second, the Oracle Coherence Data Grid is designed for lights-out management/zero administration (LOM/ZA), which provides the ability to expand and contract Oracle Coherence almost instantaneously in response to changing demand. The Oracle Coherence mesh architecture becomes increasingly nimble as the cluster size increases, with rebalancing occurring even faster and server failures having smaller and smaller impacts.

Figure 3: In an SOA environment, state management is the responsibility of the data grid, which easily and cost-effectively scales data access far beyond what can be achieved with a database and also delivers significantly better scalable performance and data integrity for conversational state. At the same time, the data grid results in a substantial increase in deployment agility.

Web Services Require Scalable Performance

SOAs and Web services in general exhibit the same requirements for scalable performance as any line-of-business or outward-facing application. In the same way that advanced Web applications manage HTTP sessions to provide conversational state on the server on behalf of the user, Web services and other SOA infrastructures often have to implement state-full conversations and workflow. In fact, many Web services implementations simply use HTTP sessions[7][10][13].

On a request-by-request basis, the data access requirements for Web services appear to be significantly higher than for Web applications, due to the nature of Web services, in which ancillary data is often included to eliminate the need for subsequent requests. In some cases, the request volumes are also significantly higher and growing at a much higher rate, largely because the service clients are no longer humans impeded by think time.

Conclusion

For years, SOA has been lauded as a sure way for enterprises to lower IT costs through improved reuse of technology assets and to become more responsive to the needs of the business. Today, leading companies are adopting Service Oriented Architecture to...
simplify their IT environments and expand their business capabilities. SOA installations are in wide production. SOA proponents often site studies showing that most IT organizations spend three-quarters of their budgets on maintenance, leaving only one-quarter for new initiatives that drive value to their organizations. SOA, they say, will enable IT to reverse this ratio and spend more time on new initiatives.

Clustered caching and data grid infrastructures ensure availability, reliability, and scalable performance for SOA. SOA environments are adopting these two technologies much more rapidly than earlier architectures, due to their combined value. As the recognized market leader in clustered caching, Oracle has been at the forefront of making SOA a reality, with Oracle Coherence already powering many of the world’s largest and most demanding SOA environments.

References