Bulk Processing with Oracle Application Integration Architecture

An Oracle White Paper
January 2009

ORACLE

Bulk Processing with Oracle Application Integration Architecture

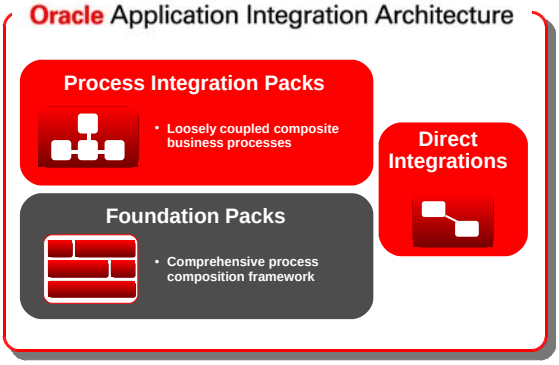
Bulk Processing with Oracle Application Integration Architecture

INTRODUCTION

Integrations are often separated between loosely coupled SOA integrations and direct system-to-system integrations that involve transferring large loads of data from one application to another. For many companies, movement of large loads of data constitutes mission critical transactions. Hence Bulk-processing is a key element of a comprehensive SOA Strategy.

Oracle Application Integration Architecture allows us to use both loosely coupled and direct integrations to solve our data integration needs. This paper describes the techniques and patterns used to implement large volume bulk-processing integrations while supporting the larger SOA infrastructure.

Through the use of Oracle Data Integrator (ODI) we create highly performant, high volume batch data-integrations with cross-referencing support that is reusable for all SOA integrations.



Oracle Application Integration Architecture

Oracle Application Integration Architecture (AIA) Foundation Pack is a holistic approach to orchestrating agile, user-centric business processes in support of business requirements that span your disparate enterprise applications.

Application Integration Architecture (AIA) is the most complete integration solution for orchestrating agile, user-centric business processes across your enterprise applications. AIA products include:

- **Process Integration Packs (PIPs)**: Pre-built, packaged integrations to support composite business processes across both Oracle and non-Oracle applications.
- **Foundation Pack**: Business process composition framework that provides the architectural and programming model, best practices, utilities, and application independent Enterprise Business Objects and Services on which AIA users can develop standardized, cross-application process integrations .

Customers use Foundation Pack to accelerate SOA adoption and quickly build integrations between their best of breed applications.

- **Direct Integrations**: Used where appropriate to augment and support SOA integrations, such as ODI for high volume batch data integrations.

Oracle Data Integrator is used in AIA Direct Integrations to provide high performing, high volume batch data integrations while supporting the larger SOA ecosystem.

Oracle Data Integrator

As part of the Oracle SOA Suite, Oracle Data Integrator (ODI) is a comprehensive extract, load and transform (E-LT) data integration platform. ODI makes use of open connector technology defined in Knowledge Modules.

Combined with a connectivity layer such as JDBC, JMS, JCA or other, Knowledge Modules define an Open Connector that performs defined tasks against a technology, such as connecting to this technology, extracting data from it, transforming the data, checking it, integrating it, etc. ODI supports the SOA ecosystem with Knowledge Modules that maintain SOA cross-references (XREF).

Cross Referencing (XREF)

In composite business processes, business concepts (Customer, Sales Order, Invoice, etc) may be represented across multiple systems. Each system may employ a different scheme in assigning primary keys resulting in one customer having different primary keys across systems. Thus, in almost every integration scenario, cross-referencing is a key requirement to map entities between integrated systems. A central cross-referencing table keeps track of the mappings of records and the integration layer needs to take care to maintain the cross-referencing table. Whenever we receive a message regarding a customer from one system, we have to effectively match to the correct representation of the customer in any of the other systems participating in the integration.

Using the XREF utilities in the SOA suite, we create a middle-layer reference table that includes the primary key for each participating system mapped to a global id. This effectively isolates each participating system yet allows us to properly identify the same business concept across multiple systems.

CHOOSING THE RIGHT INTEGRATION PATTERN: LOOSELY COUPLED VERSES DIRECT INTEGRATIONS

Before we examine the AIA Data Integration Patterns, let's first review some of the considerations we have when choosing whether to use a loosely coupled or a direct data integration.

Loosely Coupled

Loose coupling through a canonical (application independent) model is true SOA. Participating applications in loosely coupled integrations communicate through a virtualization layer. Instead of direct mappings between data models, transformations are used to map to the canonical data model. While this allows for greater reusability, the transformations both increase the message size and consume

more computing resources. For instance based integrations, this is the ideal integration pattern since the reusability we gain is worth the slight overhead cost.

Point-to-Point Data Integration

When data integration involves either a large batch of records, or very large data, we have to consider a point-to-point integration specializing on the movement of data with high performance with a trade-off of reusability.

A point-to-point data-integration in AIA uses the Oracle Data Integrator (ODI) to provide us with the best balance of performance and reusability. Before ODI, any ETL solution had to be augmented with logic to maintain the cross-references so any SOA based data-integration would be able to handle ongoing updates to the data. This process typically required programming, making the integration more costly to maintain.

ODI allows the development of a high performing data integration that, while being point-to-point in nature, is reusable in a SOA context by leveraging the ESB_XREF knowledge module.

Maintaining cross-references (XREF) to support ongoing DML operations is as simple as including the ESB-XREF knowledge module and defining the properties describing your XREF data-store. The open connector technology in ODI eliminates the need to hand-code logic to maintain the cross-references increasing the reusability of your ODI projects.

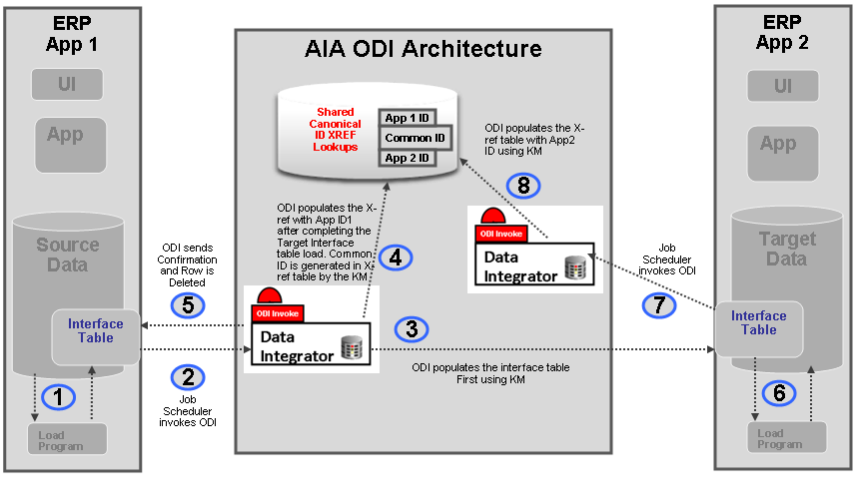
DATA-INTEGRATION PATTERNS

Application Integration Architecture supports the following data-integration patterns.

- Initial Data Loads
- High Volume Transactions with XREF table
- Intermittent High Volume Transactions
- High Volume Transactions without XREF

Initial Data Loads and High Volume transactions with XREF table

When implementing an integration, there may be some records that need to be synchronized. Using ODI allows us to handle a potentially large data-set and maintain the cross-reference if necessary.



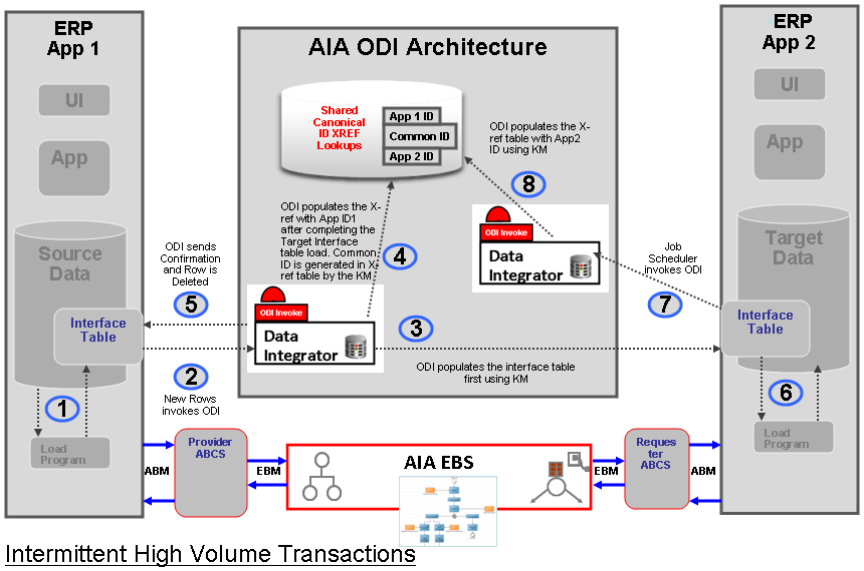
Data Load with Xref

The AIA Order-to-Cash Process Integration Pack uses this pattern to synchronize accounts in Oracle E-Business Suite with customers in Siebel CRM. The loosely coupled AIA Enterprise Business Service (EBS) could be used to achieve this integration, but ODI is used given the potential of an ultra-high volume of records. Since there can be upwards of millions of accounts in Oracle E-Business Suite, the risk of incurring an unacceptable installation time is mitigated by leveraging the point-to-point data-integration with ODI. After the integration is implemented, further account creation or updates in Oracle E-Business Suite are synchronized in Siebel CRM using AIA EBS, and ODI is no longer used.

This same pattern is used in ongoing high volume transactions, typically in a batch process. For example, in a retail environment, store locations may send order information in a batch to a central office. This is an ongoing batch where a high volume of orders may occur.

Intermittent High Volume Transactions

The Intermittent High Volume Transactions pattern covers the scenario where a data integration is managed both by ODI and by AIA EBS. Normal (Instance-based) synchronization is done through AIA EBS, but if there is the need to run transactions in batch, then ODI can be used to perform the synchronization. ODI can also be used for intermittent transactions that may contain a very large object, such as an order with an exceptional number of lines.

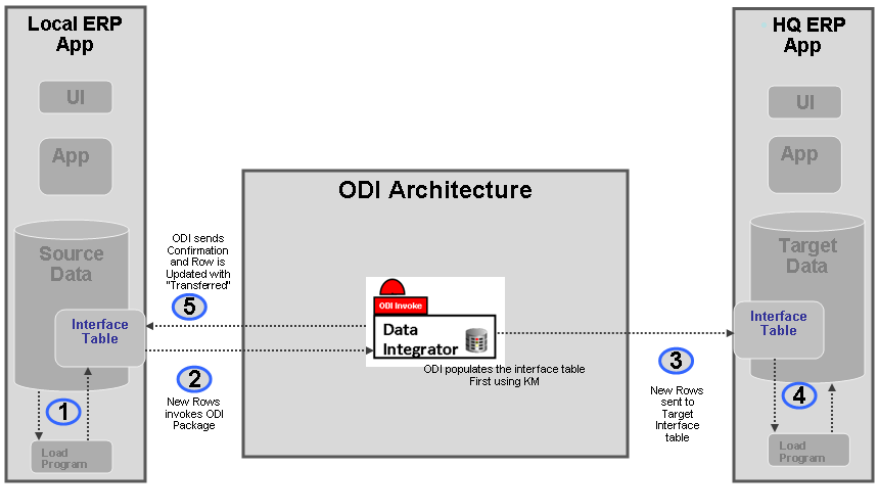


Intermittent High Volume Transactions

When using this pattern, there is typically some switching logic employed to decide which route to take. If a system were receiving incoming orders entered through a CRM system for example, it may use the AIA EBS route, since these orders are singular in nature, and may also require some level of feedback to a customer service representative. If this same system were to also receive batches of orders through an EDI gateway, then it may place those orders in an interface table to be picked up by ODI and processed in batch.

High Volume Transactions without XREF

If no further DML operations need to occur with synchronized records, maintaining a cross-reference (XREF) record would not be necessary. An example of this may be the retail environment where store locations transmit their daily



High Volume Transactions Without Xref

sales transactions to the HQ location. If the HQ location does not need to perform DML operations such as the case of a data-warehouse, then maintaining the cross-reference is not necessary.

CONCLUSION

When developing data integrations we are faced with the performance versus reusability challenge. High volume bulk data transactions, which are often mission critical require the highest performance but using traditional ETL tools and methods forgoes reusability. On the other hand, the reusability we achieve in a loosely coupled integration comes at a slight cost of processing overhead.

With AIA, we choose not only the appropriate pattern for the integration, but we also leverage features that make our integrations reusable. When the data integration requirements require a bulk load pattern, the Oracle Data Integrator provides us with easy to define cross-reference maintenance making our synchronized data reusable in the SOA ecosystem

Bulk Processing with Oracle Application Integration Architecture
January 2009
Author: Robert Wunderlich
Contributing Author: Sathya Phanindra

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com