Batch job framework for data intensive jobs

Introduction:

Every major e-commerce company relies on two basic architectures to run their business.

1. System that supports its basic business workflow
2. System that addresses everything that happens after the basic workflow is complete.

The first system usually comprises of public websites or API’s that customers will directly interact with to complete a transaction. These can be build using different technologies available today (JAVA, .net, PHP, Python). The second system usually comprises of multiple batch jobs that pull data from one system to load them into a different system as needed (ETL). As of today, there are multiple tools available in market that help automate the ETL like Informatica, SQL server information services (SSIS) and Data stage. Each of these tools address a basic need to run a batch job and helps setup alerts on such jobs. Issue though with these is that they are licensed products and each come with their own set of challenges. Like for example Informatica needs data to be sorted in certain cases before a job can be run. Also, these products come with a price tag and sometimes need regular upgrade fees + product support charges.

If we can build a homegrown system that will have the maximum flexibility to suite a business need where the basic code is always available for enhancement and can be written in a language that the database where the data resides supports, that will help business avoid challenges and cost implications with out of box products. The batch framework that this project addresses is one such solution. This framework will provide all basic components to setup a batch job and then also help setup alerts. It will also be scalable in a way that we can setup the jobs to load as much data as possible with minimum disruption to any other process accessing same tables (row locks). We will also setup framework to log each step in a operation in a multi-operation batch job. Framework will also have flexibility to setup jobs of various ETL types (Table to table, table to file and file to table).

Implementation:

We will utilize following technologies to code and implement this batch framework.

1. UNIX system (CRON for scheduling and logging job info to files).
2. Oracle database (11g) with single node.
3. ER studio/ER win for data modelling.

Most of the framework will be coded in oracle PLSQL (procedural Language/Structured Query Language) where we will code API’s that batch jobs will use.

Different stages of project:

Following will be different stages of this project.

1. Setup development environment.
   1. Install Linux system.
   2. Install Oracle 11g database.
   3. Install database development tool (SQL developer/Toad/PLSQL developer)
2. Data modelling for framework tables.
   1. Set of tables that will hold job metadata.
   2. Set of tables to log and track jobs in progress.
   3. Set of tables to log error details if the job fails.
3. Code framework logic in PLSQL.
   1. Database packages that will work as API’s. These packages will have logic to make framework scalable and also log job details when running.
4. Code UNIX shell scripts which will be used by CRON jobs to execute jobs at a scheduled time.