Moving to the APEX Listener

10 jul 2012 - Dimitri Gielis

Verschenen in O Gh Visie voorjaar 2012

Dit artikel geeft een korte samenvatting van de voor- en nadelen van elke optie. Het gaat dieper in op de Apex Listener in het bijzonder. Het artikel probeert een antwoord te geven op de vraag of APEX Listener een waardig alternatief is voor EPG en OHS en op de vraag welke voordelen APEX Listener heeft ten opzichte van EPG en OHS.

Moving to the APEX Listener
In order to run Oracle Application Express (APEX) you need to make a choice which web server you want to use. With APEX 4.1 you have three choices:

- The Oracle HTTP Server with mod_plsql (OHS)
- The Embedded PL/SQL Gateway (EPG)
- The APEX Listener (standalone or on top of a Java Application Server)

By Dimitri Gielis

This article will go briefly about the advantages and disadvantages of each option, but will dive deeper into the APEX Listener. The article tries to answer two questions in particular:

- Is the APEX Listener a worthy alternative to the EPG and OHS?
- What advantages does the APEX Listener give me compared to the EPG and OHS?

This article isn’t intended to be an installation guide of a web server or a definitive guide of the APEX Listener, but it will help to guide you to make the right choice for your environment.

APEX Architecture

The APEX architecture is a three-tier architecture consisting of a web browser, a web listener (web server) and the Oracle database containing Oracle Application Express (APEX).

![APEX Architecture Overview](image)

Figure 1 – APEX Architecture Overview

The architecture is simple, you request a page in a web browser and the web listener feeds that request to the APEX engine in the Oracle database. APEX applications are not deployed on the Web Server, they reside in...
metadata in the Oracle database. Both the development and use of an application is done through a web browser, so no extra components need to be installed on the client, it just needs a web browser.

Since applications aren’t deployed on the web server, scaling APEX applications is a bit different than typical Java or .NET applications, which do reside on the Web Server. For those applications you would scale on the Web Server, whereas with APEX the speed of the database is the most important factor. So scaling APEX applications is typically done on the database server, for example by using a grid architecture.

The Web Server however is still an important piece in the APEX architecture as it needs to send the browser requests as fast as possible to the APEX engine and return the result in the most efficient way to the web browser.

The Web Server can be used to store static files for example images, JavaScript files etc. which makes the caching of static content more efficient.

The three choices

Oracle HTTP Server (OHS)

The Oracle HTTP Server has the longest history of the three choices. It has been available since 1999 and ships with the database from version 8.1.7 onwards.

The Oracle HTTP Server is actually an Apache server, with some modules Oracle added and some specific configuration for an Oracle environment. As it’s based on Apache, which is one of the most popular web servers, there is a big community behind it. If you need help or you want to extend Apache with some extra modules (for example virtual hosts and compression), it wouldn’t take long to do so.

Apache is also very flexible, the configuration is done in configuration (text) files and many options are available. There are extensive logging and debugging options available that make it easy to track down if something is not behaving as expected or if you want to trace someones web session.

Mod_plsql is the module that maps the browser requests into database stored procedure calls over a SQL*Net connection. The downside of mod_plsql is that it is not longer actively developed and is even not available or supported in some new Oracle products (for example E-Business Suite R12).

When installing the OHS you can choose to put the OHS on the database server machine or on a different machine. Both have advantages and disadvantages. Having both the database and the web server on one machine means less moving parts and lower latency. Having the web server and database server on different machines allows you to add additional security (for example firewall between OHS and database), it’s more resilient and easier to scale out. There is one catch, you’ll need to check the license of Oracle you have. You can use the OHS at no additional cost if the OHS and database are on the same machine, but putting the OHS on different machines might come at an additional cost. Again, you should check with your Oracle representative to know what applies for you.
**Embedded PL/SQL Gateway (EPG)**

The EPG became available with the Oracle Database 10gR2 and higher. The EPG is actually a webserver inside the Oracle database. So this configuration is the simplest possible. You just have the Oracle database where the APEX engine is running and the EPG is configured. The web browser connects straight into the database (EPG).

![Figure 3 – APEX Architecture with EPG](image)

The nice thing about the EPG is that it can be setup in minutes and it’s very simple, but there are some big downsides to it. Using the EPG you will increase the database hits dramatically as for example for every request of an image it means an extra database hit. The other downside is that there are a lot less things to tweak compared to the OHS. The debugging and logging is also harder than with the OHS as the debug output is not found in log files, but you would need to retrieve that information through an API (dbms_epg).

For small environments or personal (developer) use it might be a fit, but for production environments the EPG wouldn’t be my preference of choice.

**APEX Listener**

The APEX Listener is the new way to connect to the APEX engine. Oracle is putting a lot of effort into this listener, and it seems like it’s their first choice; they are using it in the Oracle Public Cloud, they recommend an APEX architecture with the APEX Listener in the ‘APEX integration with EBS R12’ white paper and even the VP of APEX mentions already in June 2010 in one of his blog posts the Oracle APEX listener is the way to go for most [http://michaelhichwa.blogspot.com/2010/07/as-of-june-28-2010-oracle-apex-listener.html](http://michaelhichwa.blogspot.com/2010/07/as-of-june-28-2010-oracle-apex-listener.html).

![Figure 4 – APEX Architecture with APEX Listener](image)

The APEX Listener can run in standalone mode or on top of a Java Application Server. Supported Java Application Servers are Oracle WebLogic, GlassFish and OC4J.
The APEX Listener has many advantages over the other two possibilities. To configure the APEX Listener, there is a GUI interface, which you may find easier than to use an API (EPG) or raw text files (Apache). The 32k limit that mod_plsql has doesn’t exist in the APEX Listener. The APEX Listener also enables some new features in APEX itself and it will continue on that path. If you want to for example provide Restful Webservices or upload and parse native Excel files, the APEX Listener is your choice.

To use APEX in combination with some other products (for example Oracle E-Business Suite R12) the only supported configuration is to use the APEX Listener.

People already using a Java stack won’t need to install another middle-tier, but can just deploy the APEX Listener (written in J2EE) on top of their existing Java web server.

So are there no drawbacks with the APEX Listener you might wonder? I believe there are a couple; as the APEX Listener has only been available for a couple of years now, it’s probably not used as much as the others, but it is being used more every single day. As the APEX Listener is built by Oracle, you can only get help from Oracle or through the OTN Forums, compared to the OHS where you have an entire Apache community behind you. If you want to extend the APEX Listener, you probably need to wait till Oracle releases a newer version. Having said that, I personally find the positives still outweigh the negatives.

Using a Reverse Apache Proxy might be a good solution as well. It will give you the power and extensions of Apache and the better APEX Listener. It will also allow you to tighten security even more.

![APEX Architecture with APEX Listener and Reverse Proxy](image)

**Figure 5 – APEX Architecture with APEX Listener and Reverse Proxy**

**APEX Listener Performance**

Feature wise the APEX Listener has many advantages over the others and the difference with the others will be bigger with every new release of the APEX Listener.

But features are not everything, to use the APEX Listener in a production environment, performance is an important factor too.

To make sure the APEX Listener is as good or better than the other two alternatives I setup a test environment to compare the different web servers. My configuration looks like this:

- OHS 11g (port: 7777)
- EPG installed (port: 8888)
- APEX Listener on Glassfish (port: 8080)

All in the same environment:

- Oracle Linux 2.6.32-100 x86/64
- Oracle Database 11.2.0.2
- APEX 4.1.1
To test the different web servers I used JMeter as Proxy to record while I was navigating through the sample application. I then configured the testplan further and ran the same testplan for every web server.

Configuring JMeter to replay the recorded session wasn’t straight forward. As APEX is using a session id and some other hidden items and cookies I had to create different variables to store that information and pass it with every request.

I ran the testplan multiple times with different settings, but almost all results were the same in my configuration. The APEX Listener could handle most of the load, followed by the OHS + mod_plsql and the EPG was the least performing one as you can see in figure 7.
In my test case the APEX Listener served more pages (21.7) a second than the others.

I would recommend to run those tests on your server and with a recording of a real session. Things might be different or might change over time, but I expect the APEX Listener will become even faster as more caching will happen automatically etc.

Also note that I didn’t tune any of the web servers much, there are many ways to improve Apache for example, but that counts for the others too.

When I saw those results I was convinced the APEX Listener would be a worth alternative for Apache + mod_plsql.

I’ve been using the APEX Listener now in different projects in the last two years and I’m very happy with the results. So today my first choice is using the APEX Listener.

The future – APEX Listener 2.0 and the Oracle Public Cloud

As mentioned before the APEX Listener is actively developed inside Oracle and there is a roadmap available. The APEX Listener 2.0 will bring a much tighter integration between the APEX Listener and APEX itself for creating RESTful services. The RESTful service will also support OAuth 2.0. The APEX Listener 2.0 is not currently released yet, however at the time of writing it is rumored other interesting features will be included, like a built in virus scanner (integration via ICAP), native FOP integration, so to print PDF’s in APEX you no longer need to configure your own print engine.

One final note to proof the APEX Listener is really becoming critical for Oracle itself and it really puts a lot of focus on the APEX Listener is the fact that the APEX Listener will be used to serve the Oracle Public Cloud Database services. All the REST, APEX and SQL Developer requests will go through the APEX Listener.
Summary

I hope that after reading this article you are as much convinced as I am that the APEX Listener is a worthy alternative for Apache + mod_plsql and the Embedded PL/SQL Gateway.

Next to the several extra features the APEX Listener gives you inside and outside of APEX, the performance of your APEX application will be as good or even better than the other two web server alternatives. Not using the APEX Listener now and in the future you will be missing a lot of nice new features.

Dimitri Gielis

Dimitri Gielis is a director of APEX Evangelists, a company specializing in Oracle Application Express. (http://www.apex-evangelists.com).

This article is based on the white paper from Dimitri Gielis which will soon be published and the presentation he held at the OGh APEX-dag. A link to the complete whitepaper can be found on the Apex Evangelists site.