

Oracle BI Publisher Enterprise Cluster Deployment

*An Oracle White Paper
August 2007*

INTRODUCTION

This paper covers Oracle BI Publisher cluster and high availability deployment. Oracle BI Publisher Enterprise is a standard lightweight J2EE application. By lightweight, we mean that Oracle BI Publisher is based solely on the JSP and Servlet specifications of the J2EE 1.4 platform. This enables it to be installed on a lightweight servlet container, for example Oracle Container for Java (OC4J), without requiring a fully fledged J2EE application server. Oracle BI Publisher can be deployed on different commercial and open source application servers, such as Oracle Application Server 10g, BEA WebLogic, IBM WebSphere, Apache Tomcat and others.,

This paper provides a generic explanation for how to perform a cluster deployment. It is beyond the scope of the paper to describe the specific steps required for a particular application server. For information about how to implement cluster on a specific application server please refer to the appropriate documentation.

TYPES OF CLUSTERS

This section describes the different types of computer clusters.

High-Availability (HA)

The main function of High-Availability clusters is to improve the availability of the computing system in case of a breakdown. In HA clusters not all nodes may be contributing to the operation at a given moment. Some of the nodes are redundant standby nodes that operate only in case of the failure of the main node. This configuration is referred to as 'active-passive' deployment.

Load-Balancing

Load-Balancing clusters are used to increase performance by distributing the load among different computing nodes in the cluster deployment. A load balancer, which is installed in front of all computing nodes, is responsible for distributing the load. In case of a node failure the remaining node(s) will take over to service future requests. Load-Balancing clusters also act as a HA clusters. This configuration is referred to as 'active-active' deployment.

High-Performance Computing (HPC)

The purpose of HPC is to build a massive parallel computer from a set of small machines. HPC is focused on raw performance and not on high availability. HPC installations are not highly available and are usually rebooted often. HPC are often used in scientific calculations.

Grid Computing

Grid Computing achieves some of the HPC features; unlike HPC the Grid treats each node in the cluster as a separate machine and not as part of a single computer. This provides Grid with the robustness missing in HPC but at the expense of performance compared to HPC.

CLUSTER OVERVIEW

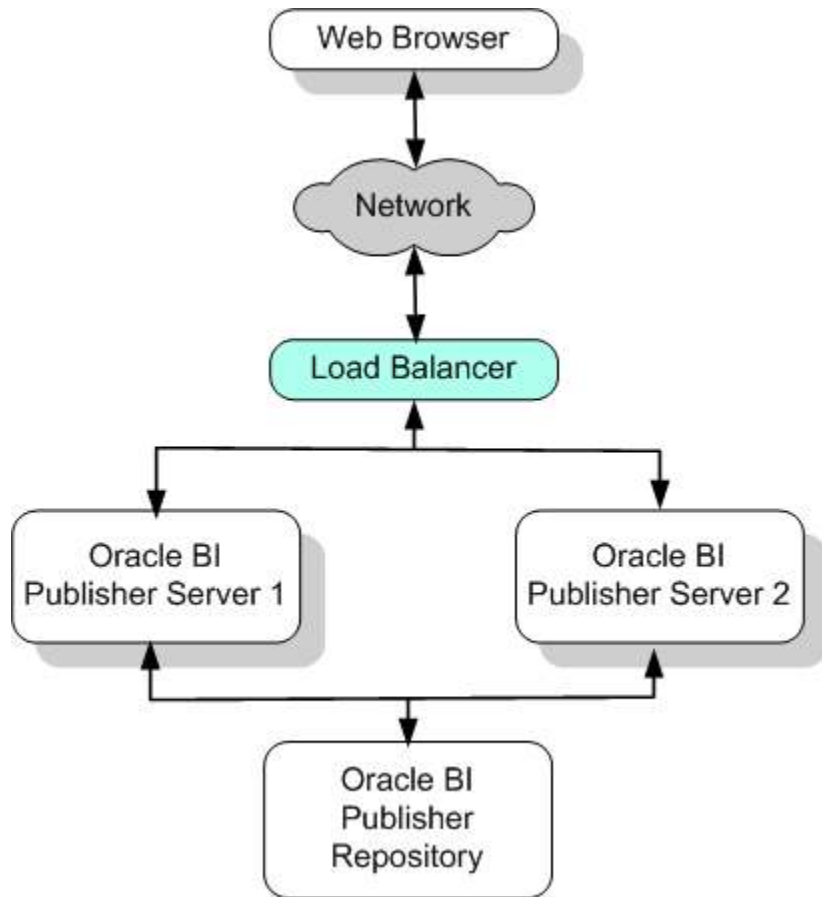


Figure 1 Minimal Cluster Deployment

Oracle BI Publisher should be installed in a load-balancing cluster setting. In the following sections we give an overview on how to do so. Figure 1 shows a minimal cluster installation of Oracle BI Publisher. The minimal install requires two instances of Oracle BI Publisher server, fronted by a load balancer. The instances

share a common repository, where all report information is stored. In the next section we discuss the different components of the cluster installation.

Oracle BI Publisher cluster has the following characteristics:

- **Common repository:** All cluster members have to share a common repository; where reports and their metadata are stored.
- **Share-nothing architecture:** Different cluster members don't share anything except for the common repository.
- **No session sharing:** Cluster members don't share session information at runtime, due to the negative performance implications of synchronizing session data across different cluster members at every request. This is one of the main components of the share-nothing architecture.
- **Sticky session:** Once a session has been established between a certain user and a certain server, all requests from that user should be redirected to the same server.
- **No transparent job migration:** In case of a server failure any job running on the failed server has to be restarted.

CLUSTER COMPONENTS

This section describes the different components in an Oracle BI Publisher Cluster.

Load Balancer

The load balancer's job is to distribute requests coming from the client to the different Oracle BI Publisher servers installed in the cluster. The minimum requirement for the load balancer is to support the *sticky session* feature, by redirecting requests from each unique client to the same server based on the *jsessionid* HTTP memory cookie.

Oracle BI Publisher works with many types of load balancers, such as software load balancers, for example Oracle Web Cache, or hardware load balancers, for example F5 BIG-IP. The load-balancing algorithm could vary from a simple round-robin scheme, to a more sophisticated intelligent routing algorithm that senses server load and adapts depending on that. This is purely a customer choice based on requirements and budget.

Servers

Each server unit in the cluster is composed of two main components: an HTTP server, for example Apache, and a J2EE container, for example Oracle Application Server. The HTTP server acts as a front end to the J2EE container. It should be configured to service all static content in Oracle BI Publisher, such as images. The J2EE container should service all the dynamic content, such as report repository navigation or report generation. This arrangement is used to utilize the Web server's optimized architecture for serving static content and to offload some of the load from the J2EE container. Figure 2 shows the details of a server.

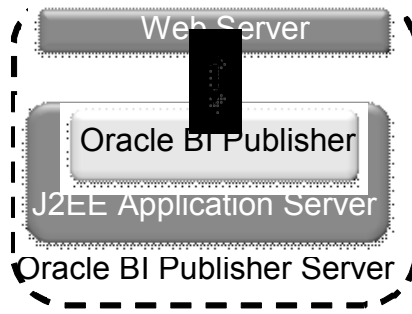


Figure 2 Oracle BI Publisher Server Details

The application server could be used to create a cluster group that is composed of the different machines that you want to deploy on. Then Oracle BI Publisher's WAR file should be deployed on that cluster group, using the application server's admin interface. The exact steps are specific to each application server; please refer to the manual of your application server. For Oracle Application Server Enterprise Deployment Guide at http://download-west.oracle.com/docs/cd/B31017_01/core.1013/b28939/toc.htm.

For advanced installations, you might choose to install the Web server and the application server on two physically separate machines. This option is preferred for high load installations. We discuss this option later in the document.

Repository

The repository is shared among different servers in the cluster. Oracle BI Publisher supports both file-based repositories and Oracle XMLDB repositories. The different options are as follows:

- **File-Based:** This is basically a shared file system accessible by a local network segment connecting cluster members. It can be an NFS directory mounted on a common server, an NFS on a NAS or a SAN running a global file system, such as Oracle Cluster File system. NFS and NAS are the simplest solutions, while SAN is more complex to implement but offers the best performance.
- **Oracle XMLDB:** This is Oracle's XML repository solution implemented on top of Oracle RDBMS. It has the added benefit of being able to be deployed on an Oracle Grid, which provides performance advantages, the removal of single point of failure, and integration with DBMS backup mechanisms.

VARIATIONS OF A CLUSTER DEPLOYMENT

Previously we described the minimal cluster deployment. In this section we describe the three possible variations of a cluster deployment. Those variations

cover the range from simple to scalable to ultra-scalable depending on the customer load needs. They also vary in the complexity of deployments.

Simple Cluster

This is similar to the minimal cluster deployment described above. The only difference is in the number of cluster nodes. **Error! Reference source not found.** shows this variation.

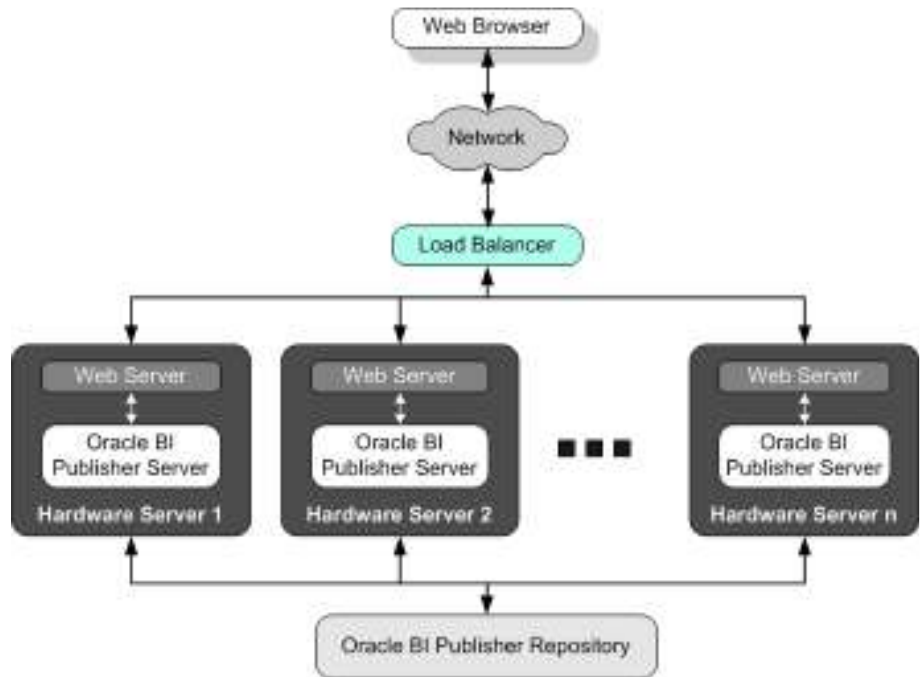


Figure 2 Simple Cluster Deployment

Scalable Cluster

In this deployment, as shown in **Error! Reference source not found.** we physically separate the Web server from the application server making each run on a separate hardware server. This is used to handle heavier loads. Each Web server is paired with one and only one application server. This is a simple configuration but if any one of the server pairs breaks, the entire pair is rendered unusable.

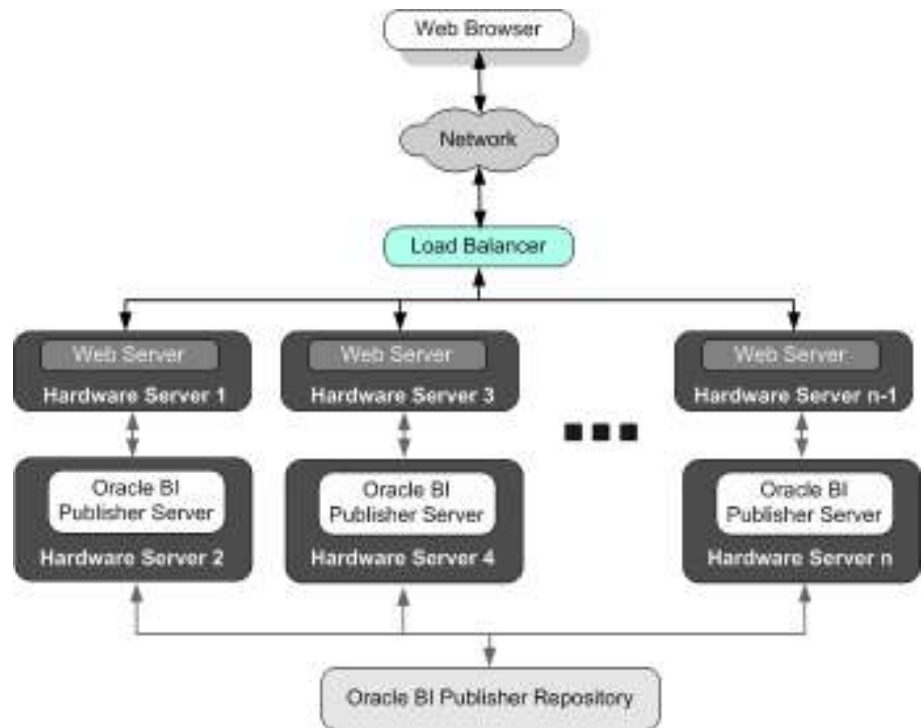


Figure 3 Scalable Cluster Deployment

Ultra-scalable Cluster

This deployment, **Error! Reference source not found.**5, is a step up from the previous deployment. The main benefit of this deployment is that there is no pairing between the Web servers and the application servers, which make this deployment more robust to the breakdown of any single node. It is more complex and expensive to install.

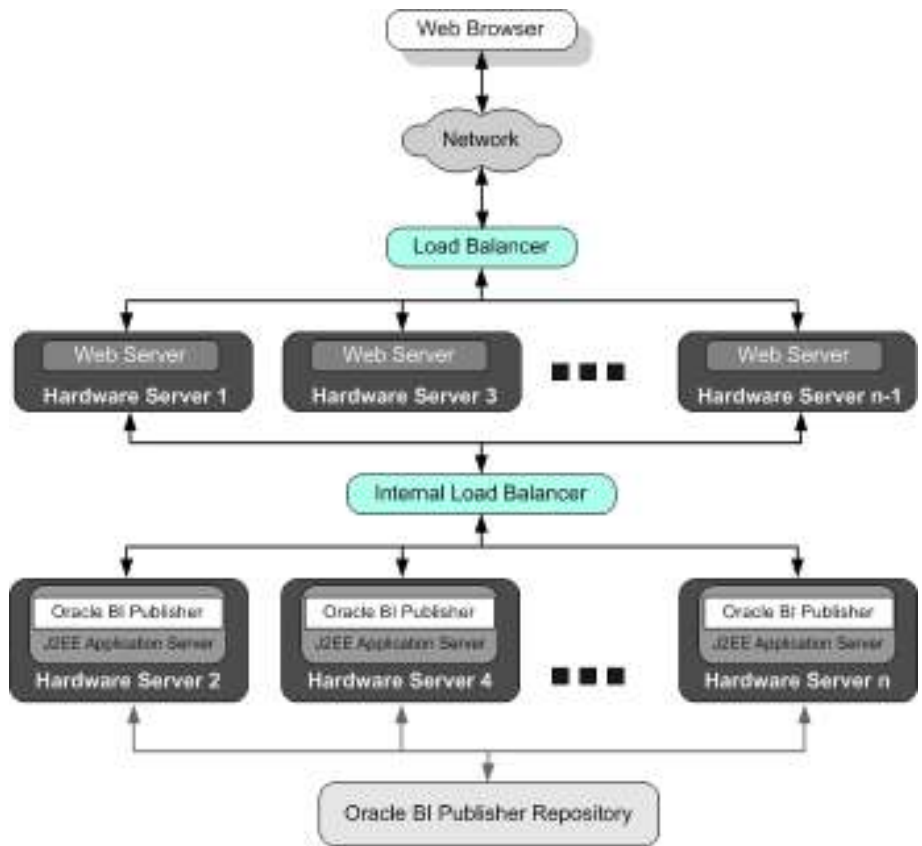


Figure 4 Ultra-scalable Cluster

SYSTEM CONFIGURATION

This section discusses the different configuration requirements.

Clock Synchronization

The clocks of all servers participating in the cluster, must be synchronized to within one second difference. This is accomplished using a single network time server and then pointing each server to that network time server. The actual procedure of pointing to the network time server is different from Windows to Linux. Please refer to your operating system's manual.

Front Web Server

The front Web server, for example Apache HTTP server, must be configured to service the following directories in Oracle BI Publisher's deployment directory:

- cabo/*
- js/*
- xdo/images
- xdo/jslib

- xdo/styles
- xdo/tmp
- xdo/uix

Those directories appear in the URL directly under the root context of Oracle BI Publisher's URL. For example, http://<SERVER_NAME>/xmlpserver/cabo, will be used to access the /cabo directory. Please refer to your Web server documentation for information about how to configure this.

Scheduler configuration

Make sure that the scheduler is configured to operate in a cluster node as described in the *Oracle BI Publisher User's Guide*. This could be done by using Oracle BI Publisher's administration UI, please refer to the documentation.

Independent Temp directories

Make sure that servers don't share the temp directories used to cache reporting data and generated reports artifacts. You can do that by specifying that each temp directory be mapped to a non-shared local drive on each server.

Sticky Session

Make sure you configure your load balancer to redirect requests to the same server based on the *jsessionid* HTTP header cookie. Please refer to your load balancer's documentation for information about how to do this.

CONCLUSION

In summary, Oracle BI Publisher Cluster deployment follows that of a normal J2EE application in a cluster deployment with a *sticky session* configuration. This configuration is both simple to install and manage and offers the best performance. The only drawback an end user will experience in case of a server failure is to have to re-login and re-execute the report he/she was seeing online.

Additional information can also be found at:

<http://www.oracle.com/technology/products/applications/publishing/index.html>

Please contact your Oracle sales representative to schedule a demonstration of Oracle BI Publisher.



Oracle BI Publisher Enterprise
August 2007
Author: Ragy Eleish

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

Copyright © 2007, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, JD Edwards, and PeopleSoft, are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.