PRIMAVERA

P6 Compression Server White Paper
Release 8.1

May 2011

Copyright

Oracle Primavera P6 Compression Server White Paper

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle and Primavera are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

To view the list of third party component disclosures related to this product, see the Commercial Notices and Disclosures document for this product.

Contents

Business Problem

The current Oracle Primavera P6 Project Management application is a fat client, with most of its required data loaded up-front. The price of loading data is paid mostly during login and when opening projects. This approach has a number of advantages:

- A responsive GUI interface for most purposes.
- An ability to perform complex calculations on the client machine. For example, scheduling, leveling, and applying actuals.

However, there is a downside to this "greedy" approach. Drawbacks include:

- High memory and CPU requirements on the client machine.
- Serious degradation in data load times on a WAN especially one with *low bandwidth* and *high latency*.

In essence, P6 Compression Server is a layer between P6 Professional and the database that compresses data before sending it to the client over the WAN. P6 Compression Server solves the WAN data load problem. There has been and continues to be efforts in reducing the up-front data needs on P6 Professional. In spite of these efforts, we cannot entirely avoid the need for large amounts of data on P6 Professional.

A compression server has a number of advantages:

- It minimizes the number data packets sent to the client.
- It minimizes the amount of handshaking-related traffic that normally happens between database drivers like DBExpress or BDE and the database server.
- It is scalable since multiple compression servers can be run on different machines for the same database.

In This Chapter

P6 Compression Server vs. Citrix

One solution for slow WANs currently proposed for Project Management client users is Citrix. The following is a comparison between Citrix and P6 Compression Server.

Compression vs. Citrix			
Feature	**Citrix**	**Compression**	**Comments**
Advantage Citrix			

Low End Clients	Yes	No	
Multi-Platform clients	Yes	No	
Latency impact for Login/Open projects	No	Yes	In a high latency network the data will still have to be moved from the P6 Compression Server to the client.
Proven Technology	Yes	No	
Central Administration	Yes	Server side	
Advantage Compression			
Cost	$250 per seat	Free to user	Cost may need to be updated
Snappiness: switching screens etc. in application	Degrades with latency	No degradation after login	The cutoff latency when Citrix is unusable is unclear.
Memory requirements on server	150MB/client	10MB/ active client. 64Kb/ passive client	P6 Compression Server has predictable load; in Citrix it depends on the data loaded into the client.
Server CPU	Unpredictable: Depends on actions of multiple users	Predictable	The peak CPU load for P6 Compression Server is a direct function of concurrent clients.
Setup and admin	Training required	Simple	For customers unaccustomed to Citrix there is a learning curve

Overview

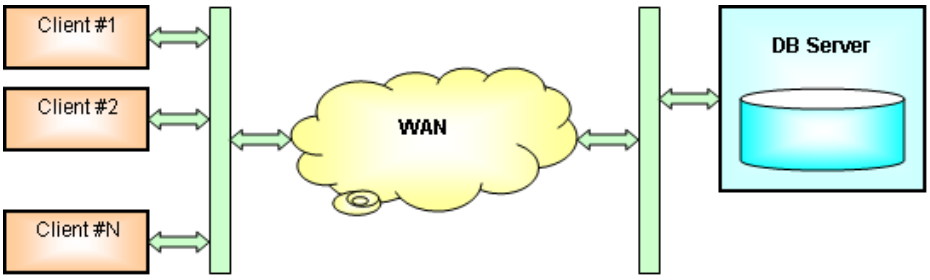


Figure 1. Classic Architecture

Figure 2 illustrates how P6 Compression Server fits into the Primavera Project Management architecture.

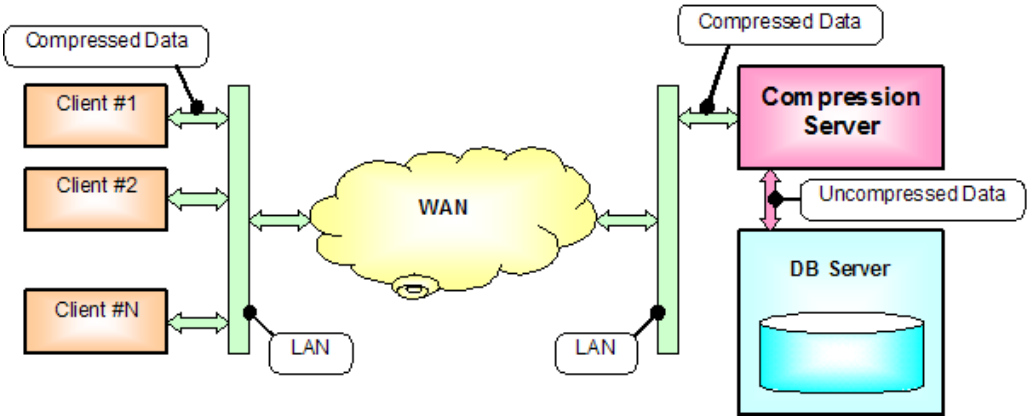


Figure 2. With P6 Compression Server

In the current architecture (Figure 1), clients 1 to N interact with the database server over a WAN. With P6 Compression Server, the clients still get and send data over the WAN. Data goes through P6 Compression Server either way (Figure 2). Data from the database server is first compressed on P6 Compression Server and then sent across the WAN to clients.

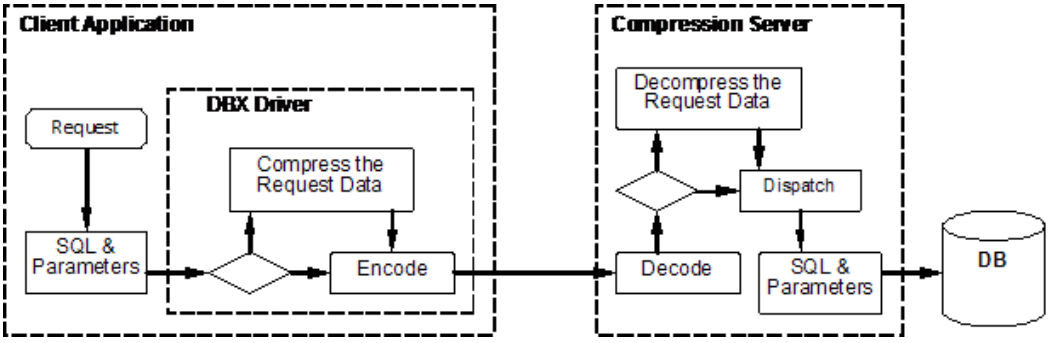


Figure 3. Request Data Flow

A typical scenario for the flow of data can be described as follows (see Figure 3):

The user logs into the application or opens a project. This generates a sequence of SQL queries. (Although P6 Professional can compress data before sending it to P6 Compression Server, we do not compress most SQL statements unless the size of SQL text and parameters exceeds 880 bytes). The data is sent using HTTP. P6 Compression Server receives these SQL queries. P6 Compression Server behaves as a proxy for P6 Professional, and runs the SQL statement on behalf of the client. It receives the result set from the database. If the result set is over a configurable threshold, it compresses the response data (see Figure 4).

The compressed data is wrapped in HTTP and sent across the WAN back to the client. The client decodes and decompresses the data as required. The server does not wait until the entire result set is obtained from the database. Rather, the data is compressed into blocks of a preset size and sent to the client, even as P6 Compression Server is fetching additional rows of the same result set. This keeps the memory footprint on P6 Compression Server to a minimum, since it does not have to compile the entire result set into a huge block of compressed data, and also prevents the client from starving while P6 Compression Server compiles a large result set.

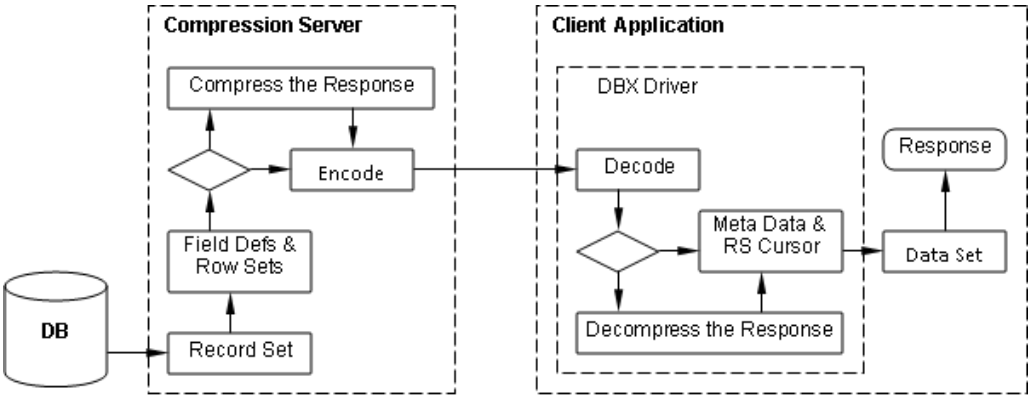


Figure 4. Response Data Flow

Batch SQL

To reduce the network traffic, the communication protocol between P6 Professional and P6 Compression Server supports packing of multiple SQL requests and dataset responses. Using this Batch SQL feature, P6 Professional can minimize the effects of the latency and achieve a better compression ratio. When in batch mode, Oracle Primavera's DBExpress driver (PrCSDrv) for P6 Compression Server compiles multiple SQL statements and parameters together into a single compressed request and decompiles the compressed responses into messages, cursors and output parameters.

Basic Architecture

Figure 5 illustrates more details behind the P6 Professional/P6 Compression Server architecture. The main Primavera Project Management application will read and write data through Borland's DBExpress technology. A DBExpress driver provided by Primavera, which communicates with P6 Compression Server, will do the actual work of fetching and sending requests and response data. This indicates why there is no significant change in the P6 Professional (the batch SQL option was one change in the P6 Professional code to accommodate P6 Compression Server). Instead of a DBExpress or BDE driver connecting to Oracle or SQL Server, the driver connects to P6 Compression Server.

For each P6 Professional request, a worker thread will perform the necessary work of creating a database connection running the query, fetching the dataset, and compressing it before returning the data back to the client.

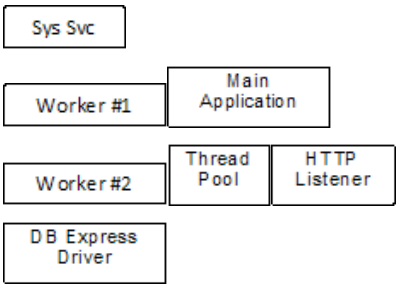


Figure 5. Design Detail

Brief Configuration Details

For this release, P6 Compression Server only supports Oracle and works only for the Project Management (PM) module and its connections with the PM database. Also, some of the features or tools that come with the PM module such as Schedule Analyzer or Update Baselines use the P6 Integration API. Since the P6 Integration API is not configured to use P6 Compression Server, you might see performance issues with these tools if you are connecting over high-latency lines.

The server-side DB Config utility has provisions for connecting P6 Professional to a compression server on a specific port through Primavera's DBExpress driver (PrCSDrv). The same client can connect directly with the database if desired. P6 Compression Server has a number of configuration options as well, including the listener port number, number of worker threads, logging, thread pool, etc. These options are explained further in the *P6 Compression Server Administrator's Guide*.

The client-side DB Config utility, which usually connects to Oracle, SQL Server, or MSDE, will have an additional option of connecting to P6 Compression Server. At this point, specifics such as server name and port number will have to be entered into the client-side DB Config utility.

Testing

A number of tests were conducted with the P6 Professional against P6 Compression Server. At present, P6 Compression Server can run on Windows 2000 Server or Windows 2003 Server operating systems.

In This Section

Test Setup

PM clients are connected through a high-speed LAN (100 Mbit/s) to P6 Compression Server. Shunra Cloud™ was used on each client in order to simulate different network conditions. P6 Compression Server is connected *directly* with the database server without any network layer in between.[1] To do so, we use a Windows PC with two network interface cards (NICs): one connects the client machines to P6 Compression Server and the other connects P6 Compression Server directly to the database server. This two-NIC configuration is detailed in the *P6 Compression Server Administrator's Guide*.

Oracle tests the performance of logging in and opening projects via the PM client using one client and five concurrent clients. For larger concurrent access, we use a tool called **SQLPlayer** that emulates the SQL load generated by the Project Management application during login. Up to three iterations are performed for each configuration (Number of clients, Network conditions) except for the Slow WAN case where one iteration was performed.

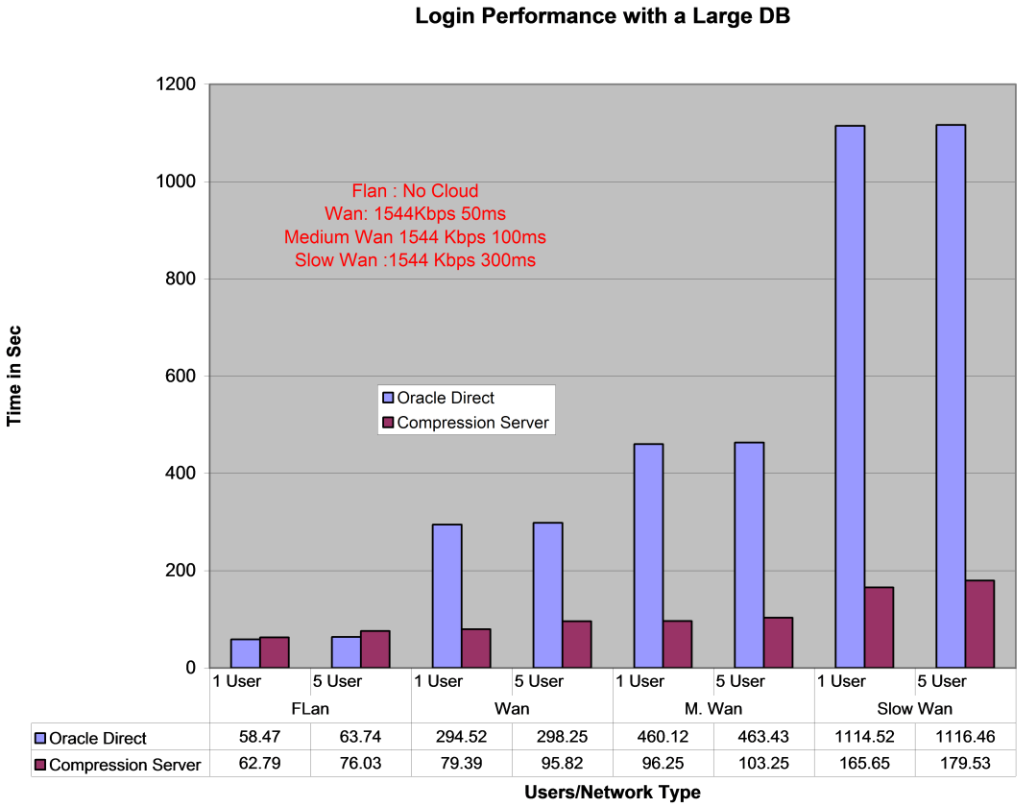
The machines used had the following configurations:

- **Client machine**: Intel P4 2.8 GHz, 2GB RAM
- **P6 Compression Server**: 2 CPUs Intel Xeon 3.2 GHz, 4 GB RAM
- **Database server**: 2 CPUs 3.2 GHz, 8GB RAM

Two sets of data were used to test performance:

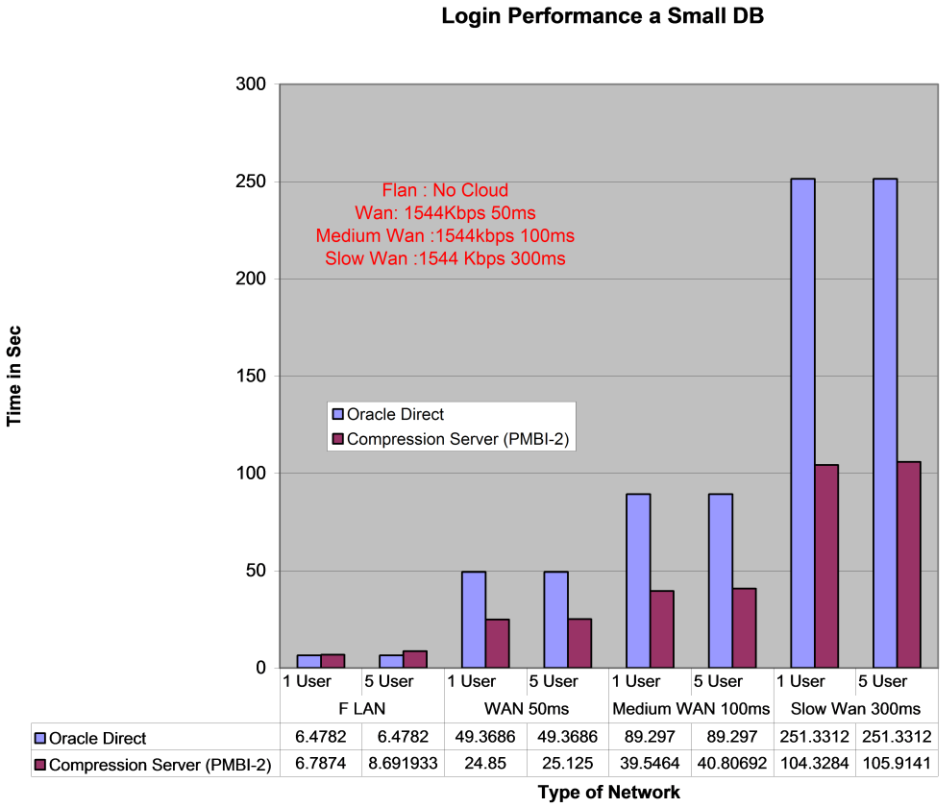
- A Large Database consisting of 16000 projects and 37000 resources. The project used for tests on this database contained 6000 activities.
- A Small Database consisting of 286 projects and 23000 resources. The project used for tests on this database contained 1000 activities.

Oracle tested a number of network conditions: high speed LAN (100 Mbps and 0 ms latency), WAN (1544 Kbps and 50 ms latency), a medium WAN (1544 Kbps and 100 ms latency) and a slow WAN (1544 Kbps and 300 ms latency). Charts 1 and 2 show login times with a large and small database for different network configurations:



	1 User	5 User	1 User	5 User	1 User	5 User	1 User	5 User
	FLan		Wan		M. Wan		Slow Wan	
Oracle Direct	58.47	63.74	294.52	298.25	460.12	463.43	1114.52	1116.46
Compression Server	62.79	76.03	79.39	95.82	96.25	103.25	165.65	179.53

Chart 1: Login Performance Chart for a Large Database



	1 User	5 User	1 User	5 User	1 User	5 User	1 User	5 User
	F LAN		WAN 50ms		Medium WAN 100ms		Slow Wan 300ms	
Oracle Direct	6.4782	6.4782	49.3686	49.3686	89.297	89.297	251.3312	251.3312
Compression Server (PMBI-2)	6.7874	8.691933	24.85	25.125	39.5464	40.80692	104.3284	105.9141

Chart 2: Login Performance Chart for a Small Database

Test Results

Charts 3 and 4 show the same results as charts 1 and 2 in the ***Test Setup*** (on page 13) topic but use the concept of *gain*. Gain is the ratio between the time to perform an action using a direct Oracle connection and performing the same action using P6 Compression Server (every other parameter viz. number of users etc. is held constant). As is obvious with charts 1 and 2, charts 3 and 4 show that the benefits of P6 Compression Server are magnified with larger databases. For example, on a Medium WAN P6 Compression Server gets data 4.8 times faster than a two-tier setup, but 2.3 times faster on a small Database. This is a result of P6 Compression Server being able to compress more data with larger loads. The greater the latency, the greater the benefit of P6 Compression Server.

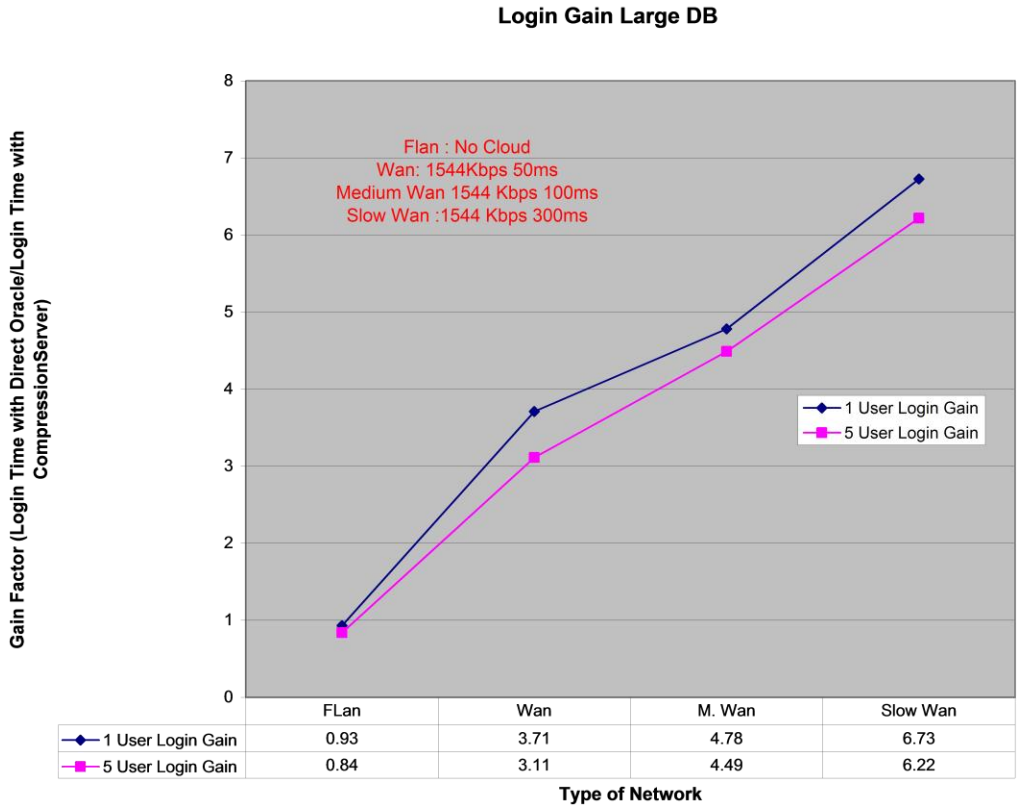
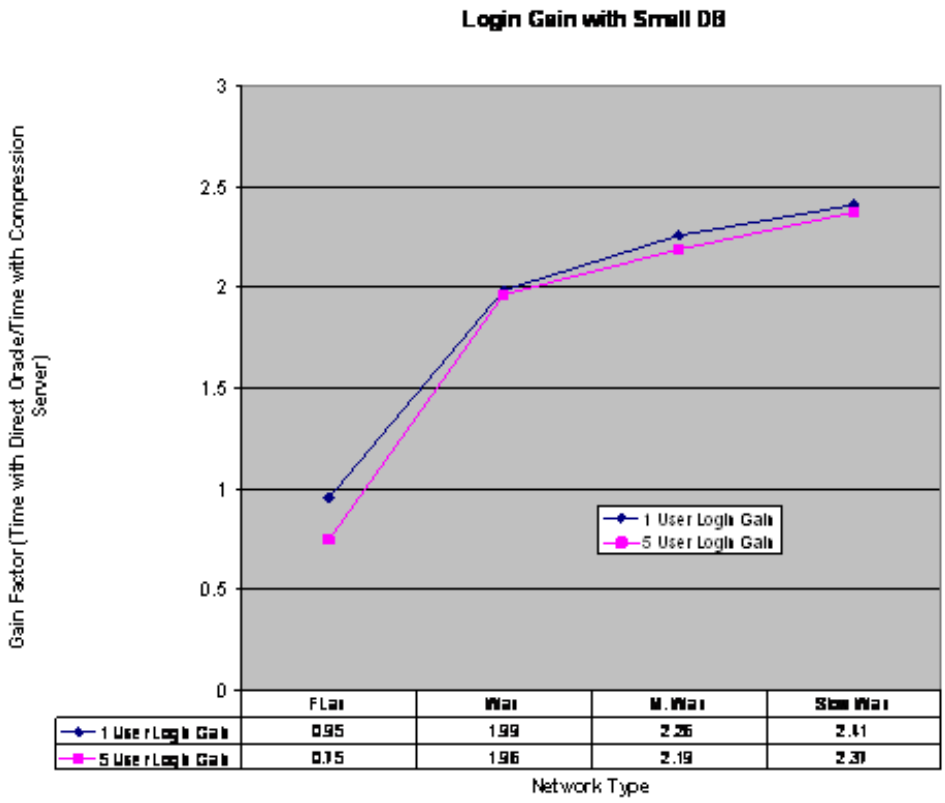


Chart 3: Gain using P6 Compression Server on a Large Database



	FLat	Wan	M. Wan	Slow Wan
1 User Login Gain	0.95	1.99	2.26	2.41
5 User Login Gain	0.75	1.96	2.19	2.39

Chart 4: Login Gain on a Small Database

The next set of charts explains how P6 Compression Server scales with larger numbers of concurrent users. Chart 5 shows the results with up to 15 concurrent users on the large database. Chart 6 shows login performance on the small database. As expected, the performance of P6 Compression Server degrades beyond five concurrent users except for the slow WAN. The bottlenecks in this case are two-fold:

1) The CPU load on the server box.
2) The data load on the network card on the box itself.

Both sets of data were generated using the SQLPlayer application which emulates P6 Professional.

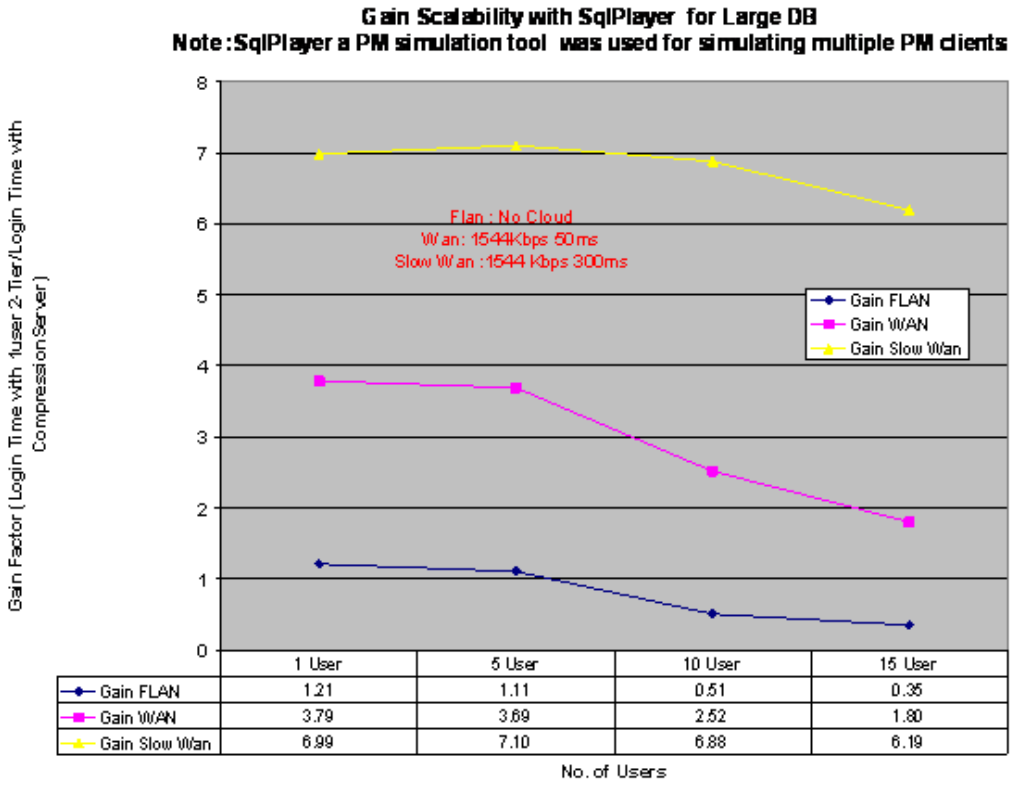


Chart 5: Scalability on a Large Database with Number of Concurrent Users

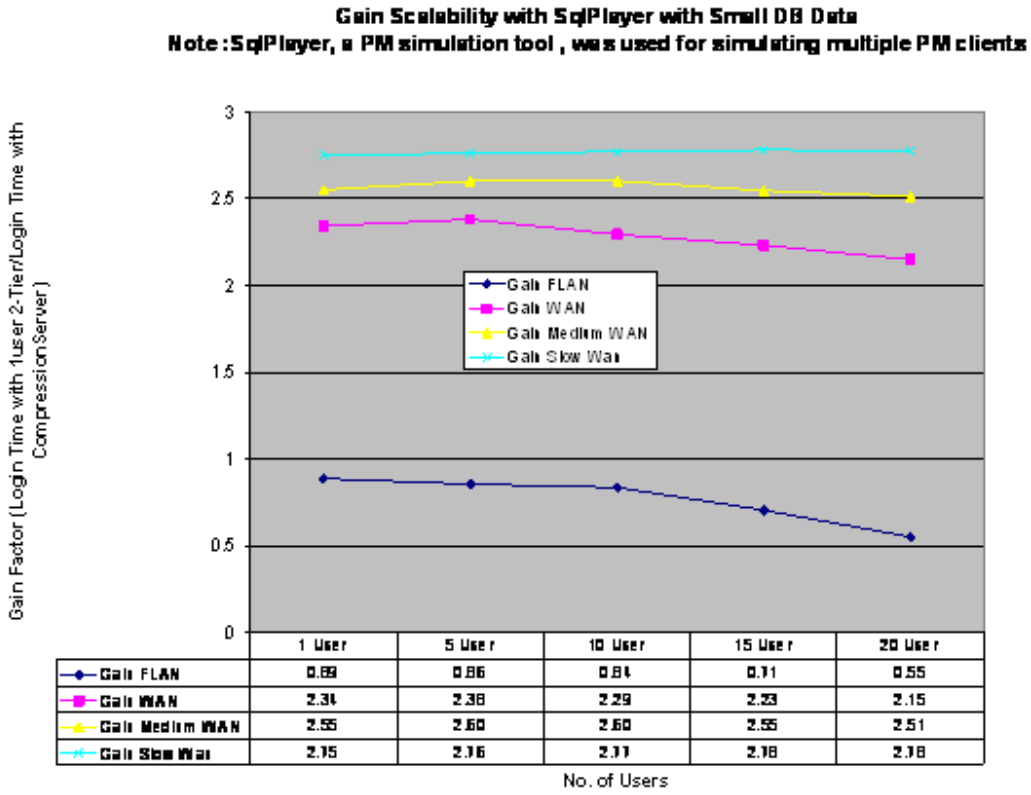


Chart 6: Scalability on a Small Database with Number of Concurrent Users

The next set of charts shows the impact of using a single NIC on P6 Compression Server rather than two NICs as recommended in the *P6 Compression Server Administrator's Guide*. Chart 7 shows the performance on a large database. It is immediately obvious that performance degrades over 50% when using a single NIC from one to five users. Chart 8 shows the same behavior for a small database. The degradation in performance is not as dramatic in this case but is still evident.

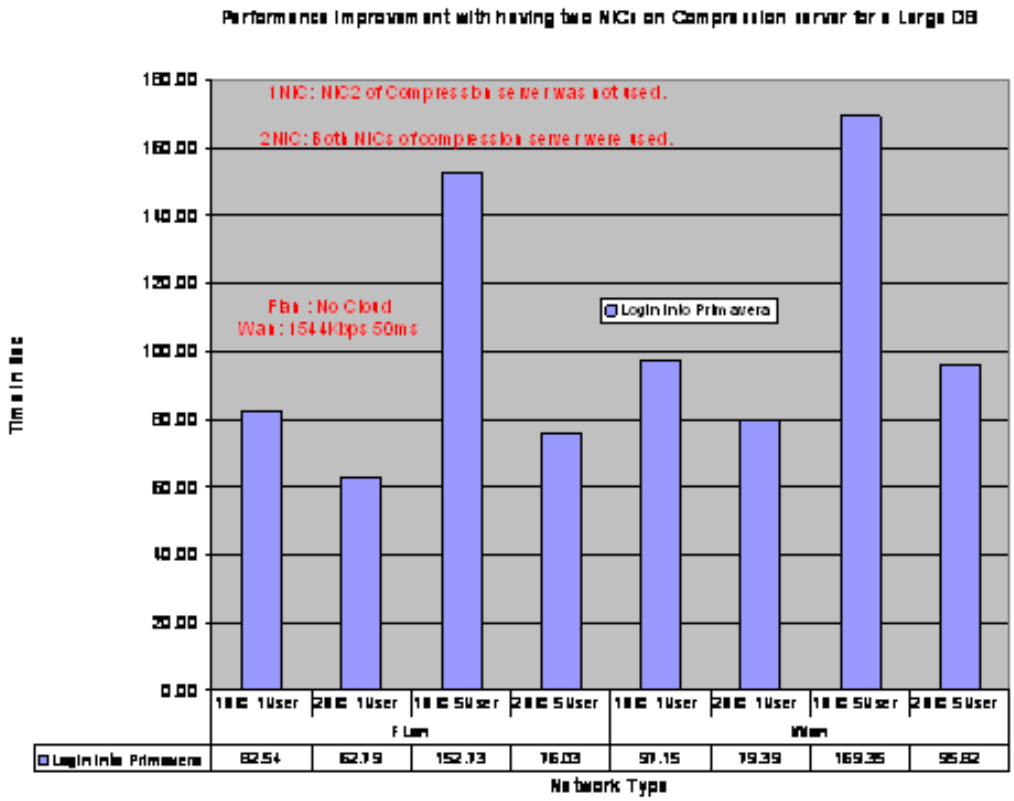


Chart 7: Performance Difference between One and Two NICs on P6 Compression Server for Large Data

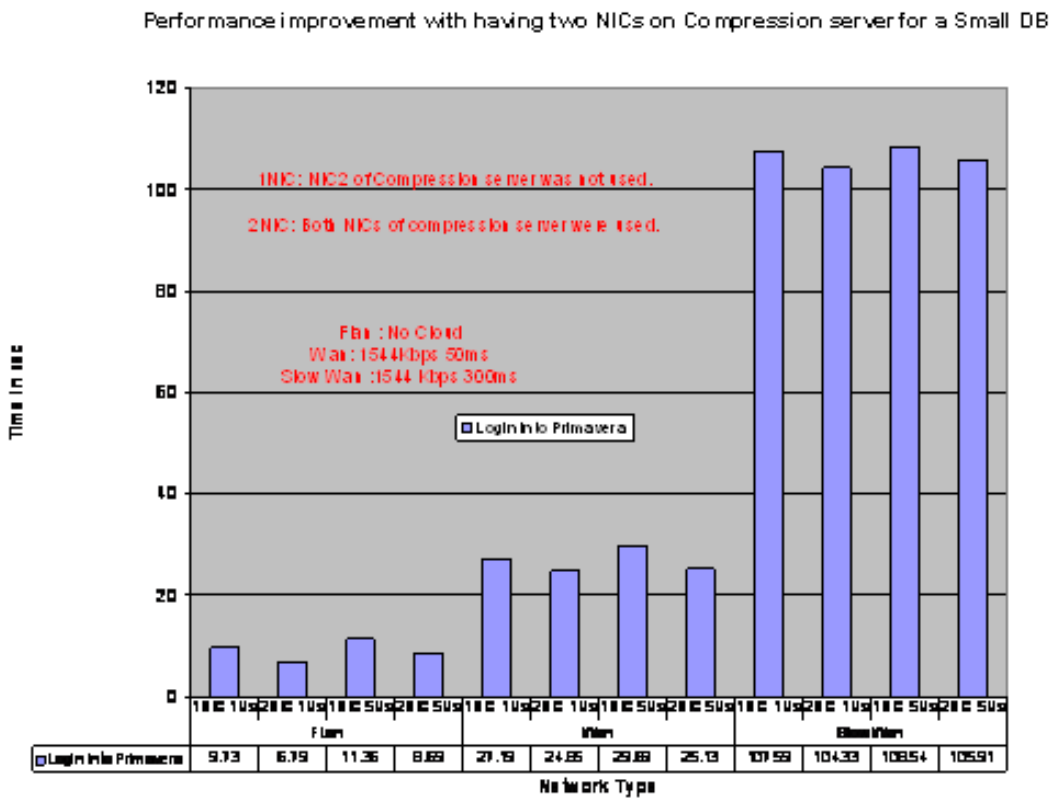


Chart 8: Performance Difference between One and Two NICs on P6 Compression Server for Small Data

The next set of charts shows open project performance with the small database. Chart 9 shows the raw numbers comparing P6 Compression Server with a direct connection to the database. Chart 10 shows how this translates into gain. As seen here, P6 Compression Server has a greater gain for opening a project when compared to login. This indicates that on high-latency networks, using P6 Compression Server provides performance gains even on small databases.

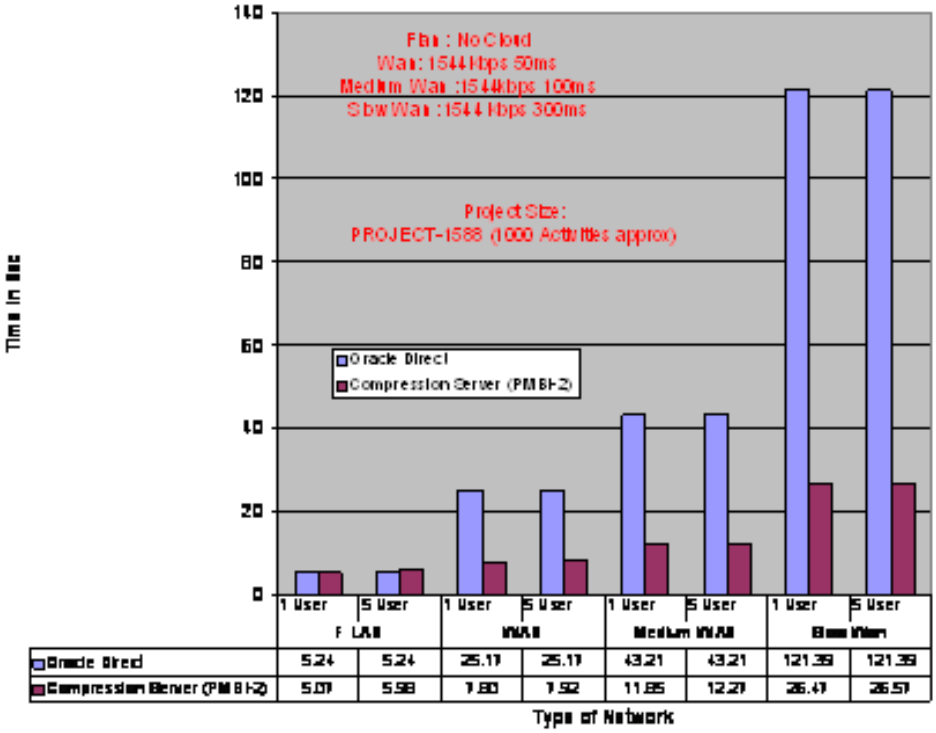


Chart 9: Open Project Performance with a Small Database

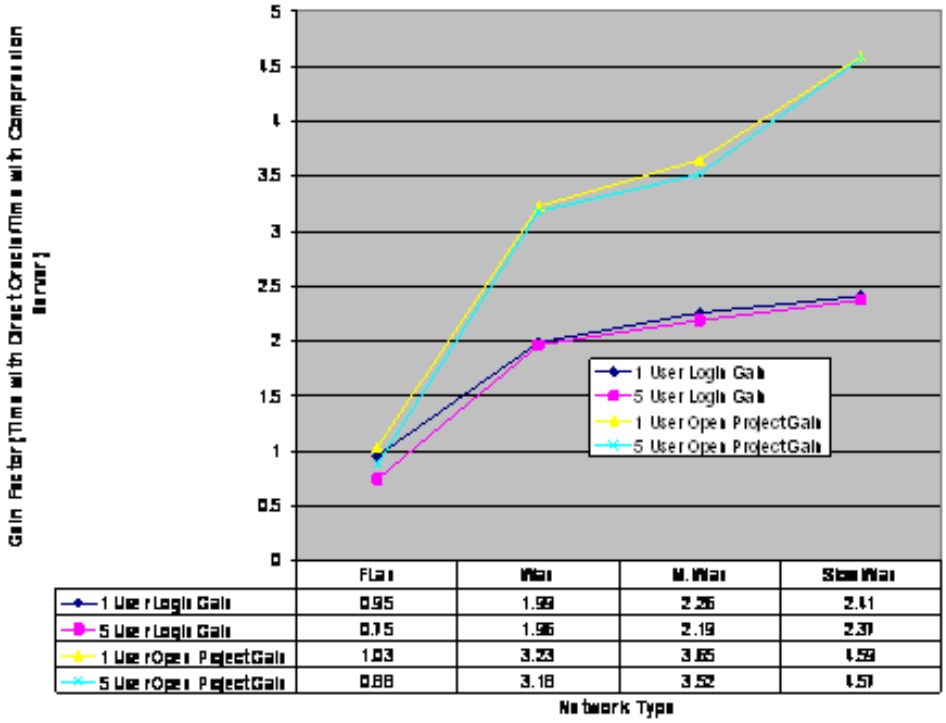


Chart 10: Login and Open Project Gains for a Small Database

The next chart shows the impact of CPU on P6 Compression Server performance. It can be seen that the use of a better CPU translates to better performance even for a single user. Furthermore, the difference between a single user and four concurrent users is more pronounced with the low-end box.

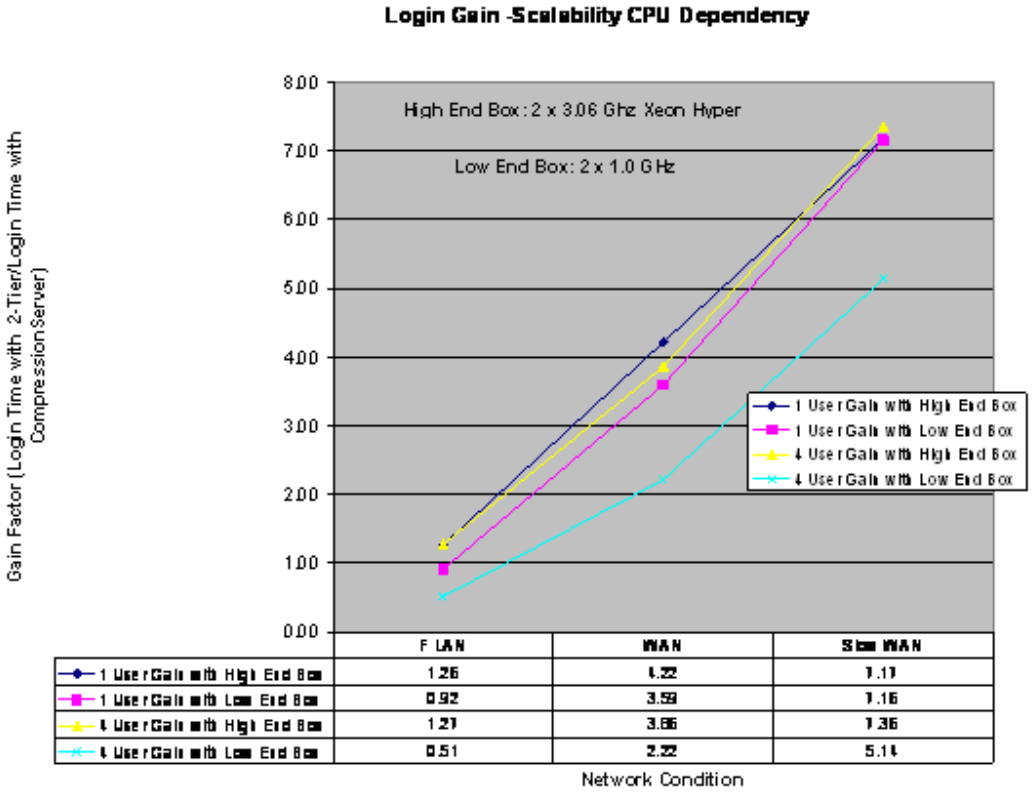


Chart 11: P6 Compression Server performance on high end and low end box

Conclusion

P6 Compression Server provides faster data load for P6 Professional compared to the standard two-tier connection. Its benefits in comparison to the two-tier setup are magnified on high-latency networks and with larger databases. That said, even with smaller databases there are significant gains under certain situations as our tests with open project showed.

Our tests show that using a faster CPU and two NICs on P6 Compression Server improves its performance considerably.

APPENDIX: P6 Compression Server FAQ

Question #1: How many total concurrent sessions were simulated?

Answer #1: We simulated 200 concurrent sessions with 5 of them being simultaneously and continuously used in processing client requests. The limit of five clients was dictated by the CPU frequency and the amount of data to be compressed.

Question #2: What was performance like under larger loads?

Answer #2: There are two kinds of large loads for P6 Compression Server.

The first type addresses the case of many clients sending Web-like (small) requests, expecting small responses. In this case the compression benefits are minimal (next to none) since there is not enough data to compress. As a consequence of this, the number of requests served simultaneously is limited by the overhead involved in maintaining threads and finding sessions. For the recommended hardware, the limit in this case is between 75 and 100 requests processed simultaneously.

The second type of large load addresses the case of few clients updating or loading large amounts of data through P6 Compression Server (large databases). In this case most of the CPU power is used for compressing the data. For the recommended hardware, the limit is between 5 and 10 requests processed simultaneously.

Question #3: How many users can one P6 Compression Server support?

Answer #3: The number of users depends on the hardware power, most used scenario and the database size. See answers #1 and #2 for details.

Question #4: Is the size of the project a consideration?

Answer #4: Yes. The size of the project directly affects the loading time during open project.

Question #5: How much RAM does each user session require?

Answer #5: The amount of RAM required by one user session is about 64KB. However, the amount of RAM used while processing requests is 10MB on average and can grow beyond 200MB when large blob data is uploaded or downloaded from the database. The 200 MB statement is true only if there are very large blobs (over five MB) in the data.

Question #6: Can multiple P6 Compression Servers be used to support large user populations?

Answer #6: Yes. However, because of the high traffic, each P6 Compression Server requires a dedicated NIC in the database server machine and therefore the number of P6 Compression Servers that can be used could be limited by the Oracle database machine. One of the problems we had was not being able to support massive reads and writes on the same NIC. This could be a problem that could be solved differently by the customer; our approach was to use multiple NIC's.

Question #7: Does each P6 Compression Server require a dedicated hardware platform or can it host other processes, i.e., P6 Progress Reporter, P6, etc?

Answer #7: P6 Compression Server, when configured and used correctly, requires more than 80% of the CPU power. It is not recommended to run any other CPU intensive applications on the same hardware with P6 Compression Server.