

Scalability Testing

This section documents scalability testing performed on Yellowfin. Results of this testing process show that Yellowfin can scale to support large user installations.

Test cases were designed to simulate a real world, enterprise-reporting environment in which users conduct a variety of business intelligence reporting activities concurrently.

What is Scalability?

Scalability refers to the ability of a system to perform well in the face of increasing user demands. For example, a system with a high degree of scalability will behave predictably when the number of users increases from 100 to 1000 to 10,000 and more. Performance refers to the amount of time it takes to complete a particular task.

Why is Scalability Important?

As the number of individuals using a reporting system grows, and the number of queries and reports generated increases, performance can become an issue. Additionally, as users learn how BI reporting helps them make better business decisions, they tend to generate more sophisticated reports and queries that put a heavier burden on the system.

A system's scalability is thus important when considering existing and projected needs. Also important are the unique reporting requirements of different user communities across departments, divisions, and geographic locations, the range of disparate data sources used for reporting, and the languages in which reports must be provided. Scalability should be a key criterion when determining the hardware and software environment in which to run your BI solution.

Testing Procedures & Results

To judge any system's scalability, a realistic evaluation of system performance in a carefully defined and controlled test situation is needed as a benchmark, or guideline, to use when configuring server environments. The testing described in this section was designed to develop reliable benchmarks for Yellowfin.

Goals

Testing was set up with the following goals in mind:

- To determine the performance and scalability characteristics of Yellowfin with an increasingly large number of users performing common tasks such as dashboard navigation, report viewing, report execution, and report scheduling (batch reporting).
- To ensure that the test users were truly concurrent, meaning that they were simultaneously stressing the server.

Approach

The hardware and software components used in the testing were designed to simulate an enterprise-reporting environment in which user activities include logging in, running dashboards and reports, and logging off.

An important approach to the testing of the Yellowfin application was to uncover any bottlenecks within the application itself and monitor performance during the process of sending and receiving queries, and rendering the results in the application engine.

Note: The tests performed are isolated to the Yellowfin application and does not include potential environmental impacts such as network latency, database server speed and browser rendering performance.

One of the other aims of testing was to try and replicate real world usage by incorporating reasonably short think times or wait times of 2 seconds. The benefits of using short think times are:

- Test results reflect high usage patterns with user behavior mimicked for active concurrency testing.
- Customers are better able to extrapolate from the results based on the level of concurrency in their reporting environment.

The following test scenarios were employed for the concurrent users:

- Logged into Yellowfin
- Navigated the Yellowfin dashboard
- Loaded 6 unique reports (that included advanced charts)
- Changed to a different dashboard tab and loaded 6 unique reports (with drill down, drill through, drill anywhere, formatters, conditional formatting, report summary, and advanced charts)
- Maximized and closed report (with chart and table)
- Logged out of Yellowfin

Software

Table below outlines the software used in the testing.

Software	
----------	--

Tested System	Yellowfin v7.3
Operating System	Microsoft Windows 7 Professional
Web Server	Tomcat 8.5.6
Database Operating System	Microsoft Windows 7 Professional
Database	Microsoft SQL Server 12
Load Testing	JMeter 2.13

Hardware

Both Yellowfin and the JMeter load testing software were run on:

Hardware	
Server	Commodity Desktop Server
Processor(s)	Intel(R) Core(TM) i7-5820K CPU @ 3.30GHz (6 Core hyperthreading equiv. 12 Core CPU)
RAM	24 GB

Users

To accurately judge the number of users that can be supported in a real-world scenario based on performance in a test environment, you must distinguish between **named**, **concurrent**, and **active** users.

- **Named** users make up the total population of individuals who can be identified by and potentially use the system. They represent the total user community, and can be active or concurrent at any time. In a real-life BI environment, this is the total number of individuals authorized to use the system. It is the number of most interest when planning a BI implementation, because it tells you how many users you can expect to support in a given environment with the response times reported in a test environment.
- **Concurrent** users are the number of user sessions logged on to the system at a given time. They include users who are simply viewing the results returned from a query. Although these subset of concurrent users are logged on to the system, they are not necessarily sending requests. Based on Yellowfin's experience, a good assumption is that 20 percent of named users are concurrent at any given time. Therefore, 200 concurrent users represent in an environment with 1,000 named users. **Note:** This ratio may vary significantly for your BI application.
- **Active** users are not only logged on to the system, but represent the subset of concurrent users who are sending a request or waiting for a response. They are the only users actually stressing the system at any given time. A good assumption is that 50 percent of concurrent users are active users. Therefore, as an example, 100 active users represent an environment with 200 concurrent users and 1,000 named users. In this case, the ratio of active user to named user is 10:1. **Note:** This ratio may vary significantly for your BI application.

Testing

The Yellowfin load testing scenarios were carried out on the demonstration SkiTeam data mart. Testing was performed on a fresh installation of Yellowfin v7.2 with appropriate values for the Yellowfin connection pool, application server max threads, and source database connections management to match the increased concurrency. This operation was also performed on a single instance of Tomcat – as seen below, clustering options should be implemented as concurrency reaches a certain threshold to ensure low response times.

Results

Report Interaction					
Active users	100	200	300	400	500
Average Viewing Response Time (seconds)	0.6	0.9	1.4	2.1	5.2
Named Users*	1000	2000	3000	4000	5000

*Estimated named users based on the number of active users on a 10:1 ratio.

Test Conclusions

The Yellowfin test results presented in this section are a result of the modern, open, and scalable architecture that was built to provide true enterprise level reporting for global organizations.

The test results show that:

- Yellowfin demonstrated that when deployed appropriately, a level of scalability that can meet the needs of extranet reporting deployments that span thousands of users.
- Yellowfin supports large reporting deployments that span multiple departments in Enterprise deployments.
- Yellowfin provides high throughput for scheduled reporting (batch reporting) environments.

It is important to note that in a real-world environment, other factors such as the network and database layers could be potential key performance inhibitors. It is recommended to perform clustering on the application and database server layers for high concurrency rates, and is the best approach with high availability and failover requirements in mind.

In conclusion, the results of the Yellowfin benchmark tests indicate that when deployed appropriately, Yellowfin is a high performance solution that can be the basis for enterprise reporting initiatives.

Previous topic: [Server configuration](#)

Next topic: [Server usage examples](#)