Report Engine Scalability and Performance

Version: Draft 1

Abstract

This document describes the performance tests to quantify report engine performance and the enhancement of report engine for better scalability and performance in BIRT 2.0.

Document Revisions

Version	Date	Description of Changes
Draft 2	08/17/2005	Updated version
Draft 1	8/04//2005	Initial draft.

Table Of Content

1. Introduction..2

2. Architectural Changes and New Features that Affects Engine Performance..................................2

2.1 Architectural Changes...2

2.2 New Features that Help Performance...3

2.3 Other performance enhancement directions 3

3. Performance Tests 3

3.1 Benchmark Tests...3

3.2 Memory Profiling 4

3.3 Performance Profile 4

3.4 Finer Control on Resource Usage 4

3.5 Recommended Configurations 5

3.6 Recommended Hardware Requirements 5

1. Introduction

BIRT 1.0 focuses on reporting functionalities. Even though report engine was able to support up to a couple of hundred concurrent users, engine performance testing and tuning was not systematically performed. As a result, it is reasonable to claim that BIRT 1.0 targets small reports with limited number of concurrent users. BIRT 2.0 intends to broaden the appeal of BIRT by improving engine performance in many areas to

- suport very large report (up to 100K pages)
- improve viewing performance for report with <1K pages
- support more concurrent users

Three types of work will help achieve these goals. First, there are architectural level changes and new features that would improve engine performance. Second, systematic engine performance tests will be conducted so that performance bottlenecks could be identified and removed. Third, performance tuning guides will be provided so that system integrators could choose the right configuration that achieves the best performance under their specific load.

2. Architectural Changes and New Features that Affects Engine Performance

2.1 Architectural Changes

The following architectural changes will be implemented in BIRT 2.0, which should improve engine performance. Please see spec for each feature for detail.

1. Separation of factory and presentation engine. This allows report viewing to be based on report document, which could have already had report data fetched, data transformations applied, and certain layout calculations performed.

2. PDF emitter rewrite. BIRT 1.0 uses FOP for PDF generation. The report is first converted to FO format and then passed to FOP. By rewriting PDF emitter, we would skip the FO generation phase and directly render a report to PDF. Because FOP puts everything in memory during PDF generation, it has already led to some complaints that BIRT PDF generation uses too much memory. The new PDF emitter would remove this bottleneck, improving BIRT memory usage and consequently improving engine performance.

2.2 New Features that Help Performance

BIRT 2.0 includes several features that would help improve performance. These features are:

1. Page-on-demand Viewing. Instead of always render a whole report for viewing, BIRT 2.0 supports viewing a specific page, and allows the user to navigate to other pages. This reduces the size of the output, and should improve engine performance accordingly.

2. Progressive Viewing. If a user wants to generate and view a report, the viewing process does not need to start until report generation is finished; instead, after 1st page is generated, the page can already be converted to HTML for viewing. This significantly improves viewing experience (even though the actual generation time is not reduced).

3. Data Engine Performance Tuning. Retrieving data from database or report document is one area that is usually time-consuming. Performance enhancement in such area would help boost engine performance.

Because there is a separate BPS project for enhancing data engine performance, the current document will not cover performance tuning for data engine.

2.3 Other performance enhancement directions

Engine performance may also be enhanced in the following directions:

- Share the resources among different sessions, such as data set, report handle, report document.
- All operations such as render, generator should be thread-safe to avoid resource locking.
- Create a pool to cache expensive resources such as Rhino context.

3. Performance Tests

3.1 Benchmark Tests

BIRT 2.0 will create tests to establish benchmark numbers for different viewing scenarios. The benchmark test should establish:

1. Report generation time under different load and concurrency scenarios.
2. Report viewing time based on pre-generated report documents.
3. Report viewing time under “run and generation” load.

4. Response time under progressive viewing.
5. Response time under interactive viewing scenariors.

Other dimensions to be considered for these usage scenarios are:

- Response times under different load scenarios, i.e., different levels of concurrency, report sizes, viewing operations (HTML and PDF).
- Scalability under different hardware configurations.
- Memory usage under different load scenarios.
- Concurrency thresholds. With reasonable configuration, what is the maximum number of concurrent users that the system can support?

These tests will be designed over time and run regularly once they are available.

3.2 Memory Profiling

Memory profiler should be used to profile BIRT engine to establish a reference point for engine memory utilization. Such results could be used in the future for comparison. Memory usage bottlenecks identified by memory profiler should be fixed (if possible).

In particular, BIRT 2.0 memory usage should be bounded. It can not depend on the size of the report (document). This is achievable because report processes data sequentially, so a large report document does not translate into more memory requirement. If some components (for example DTE) require more memory, it could serialize some data to disk to reduce memory footprint.

Memory profiling should also focus on identifying areas where temporary objects are created and discarded. Enhancements could be implemented to reduce object creation.

3.3 Performance Profile

Performance profile should be obtained to identify what tasks engine spends most of its time on. Such performance bottleneck may be a result of repetitively performing certain expensive tasks. In such cases, improvements might involve using more memory to cache expensive operations.

3.4 Finer Control on Resource Usage

Performance tests could be the first step that leads to recommendations to create finer controls on resource usages in engine. For example, if the load is mixed and involves page-on-demand viewing and downloading large reports, it may be desirable to optimize response time for page-on-demand viewing requests. It would then be useful to limit the number of concurrent large report downloading tasks.

As another example, best system throughput is often achieved by not overloading the system so that context switches can be avoided. This might involve limiting the number of concurrent threads that are allowed for engine. The actual situation could be more complex in a web-environment because the application server creates threads for request processing.

Initial performance tests should provide guidelines on whether such performance enhancements are necessary. Both the system and the tests can be enhanced iteratively over time to yield best system performance.

3.5 Recommended Configurations

BIRT targets various usage scenarios. In some cases, report are small and most people only view report page by page; in other cases, large reports is the norm, and long-running requests are common. It is not possible to have a single configuration that performs well for all scenarios. It is crucial to identify several most frequently used scenarios and provide configuration guidelines for each, so that system developers can configure their system based on their unique needs.

At least the following scenarios need to be covered:

1. A small number of users use the system (i.e., 10-100). Report size ranges from several pages to tens of pages (1-30 pages). People view report page-by-page. Occasionally, they download whole report into PDF.
2. A large number (thousands) of users use the system to view reports that are small or large. They use page-by-page viewing. Occasionally some people may download a report that is large.
3. Each report is large. However, everyone view it page by page. It is very rare to download large reports as a whole.
4. The environment is a mixed generation and render environment.

3.6 Recommended Hardware Requirements

Frequently, system developers know their system's load. They would like to know what hardware configurations are needed to support their load. BIRT 2.0 should provide such information based on appropriate performance tests.