Crystal Reports Server 2008 Sizing Guide

SAP

Applies to:

Crystal Reports Server 2008 V0, Crystal Reports Server 2008 V1,

Summary

Describes how to estimate load, and how to determine the amount of memory and CPU required to service a given load in Crystal Reports Server 2008

Authors: Blair Wheadon, Julie Ingram

Company: SAP

Created on: July 21, 2009

Author Bio

Blair Wheadon is a Product Manager in the Volume Business Unit at SAP. His team works on Crystal Reports, Crystal Reports Server, and the Volume Xcelsius Products (Present, Engage, and Engage Server).

Table of Contents

Overview

Crystal Reports Server system sizing involves the process of determining how many resources are required to support a given workload.

As input in the sizing process we often have:

> The number of users
>
> A description or understanding of types and percentage of usage types (how many users are doing what)
>
> The frequency of user activity

As output in the sizing process we want to understand:

> The number of processors (CPUs) required to support a load
>
> The amount of memory required to support a load
>
> The appropriate configuration for a load type (number of services)

The general assumption through this document is that each main service or component can be assessed independently through its capacity to respond to simultaneous requests from users in a time period with a certain amount of hardware resources (CPU and memory).

Caution is advised when utilizing this document because the results derived are approximate based on the input data and the configuration of the system used by Crystal Reports Server 2008. ***We recommend that you gauge your company's deployment hardware against information in the base system configuration section of this document***

Introduction

There are three aspects to consider when designing a scalable enterprise reporting application: the application itself, the security and system data, and the actual hardware and configuration. It's important to recognize the set of factors that can influence these three aspects of scalability. This document will focus on the major aspects that influence sizing decisions including information on actual hardware and configuration of a Crystal Reports Server 2008 deployment.

A step-based methodology for sizing is used in this guide. The following four steps will help you plan and estimate the size of a deployment:

Step 1 - Load

Load defines the amount and types of user/client activity, and transactions that will interact with the Crystal Reports Server system. The results obtained in this step are a pre-requisite for step 2.

Step 2 - Service Thresholds, CPU and Memory

In Step 2 of the sizing process the thresholds of each required service in the system is analyzed. Services are the software components required to provide the functionality and support the anticipated load. The number of CPUs and amount of memory required is also determined. CPUs and memory are incorporated in the supporting infrastructure that must support the services and load.

Step 3 - Single Machine Configuration

Step 3 of the sizing process helps determine the appropriate configuration of machines. Since Crystal Reports Server is licensed as a single server reporting solution, only a single machine configuration is considered.

Step 4 - CMS System Database Tuning

Step 4 addresses tuning of the database server. Tuning the database server and network to take full advantage of Crystal Reports Server will help to provide optimal performance and reliability.

How to use this guide for Windows and Linux environments

This document is a guide. Crystal Reports Server is a flexible system and numerous variables may contribute to an optimal configuration. This guide offers conceptual information and measurements that reflect observed Crystal Reports Server system and component behavior.   Sizing formulas in Step 2 can assist in understanding the nature of relationships between user interactions and services functionality as well as how these relate to CPU utilization, memory and/or disk consumption. For the purposes of this guide, most system and performance testing was conducted on machines averaging at 2.5 GHz clock speed with 2GB Memory per CPU.

Use this guide to better understand how each applicable service behaves, including relative capacities and thresholds within the overall Crystal Reports Server system.

This guide can be used for both Windows and Linux environments. There are many variables (inherent CPU differences, hyper threading, operating system, network, report design, functionality, SDK, etc) that impact sizing and performance. While this guide cannot account for all of these variables, the concepts, formulas, configurations and observations can be used universally. The sizing guide is a living document and may be updated as new sizing and configuration information become available.

Step 1:   Load

What is Load?

Load defines the amount and types of use and activity that will interact with the Crystal Reports Server system. Load can be broken down into various types of user interactions and user types:

Potential users is the number of users that have the ability to logon to the system.

Concurrent active users is an estimate of the number of users who are concurrently logged on to the system and are actively interacting with the system (selecting folders, viewing reports, scheduling reports, etc.). There is a distinction between "concurrent active users" and "concurrent users" as concurrent active users are actively interacting with the system at various rates, rather than simply logged on but inactive.

Simultaneous requests is an estimate of the total number of simultaneous requests being made by concurrent active users to the system (logging on, selecting folders, viewing a report, scheduling, refreshing, etc.). The number of simultaneous requests is generally dependent on the usage type and rate.

Concurrent active users and simultaneous requests are the two load types that will be heavily weighted in determining the required resources and the appropriate configuration to support a reliable and well performing system. The number of potential users is useful in determining these three measurements.

Estimating Potential Users

This is the easiest number to calculate as this is the total population of users who have the ability to access the Crystal Reports Server environment.

Estimating Concurrent Active Users

When calculating the size and configuration of a deployment, it is important to determine the expected concurrent system usage.

In our experience, many customers find that their concurrency ratios are on average from 10% to 20% of their total potential user base (e.g., 500 total potential users = 50 to 100 concurrent active users). This can vary depending on the nature of the deployment, but is a reasonable rule of thumb for planning purposes. If your estimated concurrency rate is higher or lower, you can use that value to complete the rest of the calculations in this document.

A guideline for estimating concurrent active users: concurrent active users = 10% to 20% of total potential users

> *@ 10%: 1000 potential users = 100 estimated concurrent active users*

Estimating Simultaneous Requests

The quickest method for estimating the number of simultaneous requests is to calculate 10% of concurrent active users:

> *100 concurrent active users x 10% = 10 simultaneous requests*

Estimation Example

The following process is an example of one methodology that might be used to estimate this number in more detail. If we divide the users into sets of types of users, based on how they use the system, we can more accurately assume a number. The percentages used in this example can be adjusted to match the profile of a particular environment.

For the purposes of this calculation we will divide users into 4 types:

> **Heavy Users** – users who will be persistently logged onto the system and viewing reports.
>
> **Active Users** – users who will log into the system frequently throughout the day averaging one request every 4 seconds.
>
> **Moderate Users** – users who will log into the system from time to time throughout the day averaging one request every 8 seconds

Light Users – users who will log into the system infrequently to view reports with an estimate of one request every 16 seconds.

Divide concurrent active users into these 4 defined categories and then calculate the percentage of each type:

@ 10%: 1000 potential users = 100 concurrent active users

Heavy Users	15 concurrent active users	15%
Active Users	45 concurrent active users	45%
Moderate Users	25 concurrent active users	25%
Light Users	15 concurrent active users	15%
Total	100 concurrent active users	100%

Now that we have determined the percentage breakdown for each user type, we can estimate the total simultaneous requests based on the estimated user type request frequency (i.e. How we define what a "heavy concurrent active user" means in terms of request frequency or rate).

The following percentages are only assumptions that have been made relating to the rate of simultaneous use based on user group type:

- For every 100 Heavy concurrent active users assume 100 simultaneous requests could be made or a **100% rate**

- For every 100 Active concurrent active users assume that 25 simultaneous request could be made or a **25% rate**

- For every 100 Moderate concurrent active users assume that 12 simultaneous requests could be made or a **12% rate**

- For every 100 Light concurrent active users assume that 6 simultaneous requests could be made or a **6% rate**
  Formula to calculate simultaneous user requests:

    (((Concurrent active users × % of Heavy Users) / 100) * (1)) + (((Concurrent active users × % of Active Users) / 100) * (0.25)) +

    (((Concurrent active users × % of Moderate Users)/ 100) * (0.12)) + (((Concurrent active users × % of Light Users) / 100) * (0.06)) = Calculated Simultaneous Users (rounded up)

  - or –

    (((100 × 15) / 100) * (1))) + (((100 × 45) / 100) * (0.25)) + (((100 × 25) / 100) * (0.12)) + (((100 × 15) / 100) * (0.06)) = 31

Based on the assumption of 100 concurrent active users and the types of activities each user is likely to perform, we calculate that there will be an average of 31 simultaneous user requests.

Step 2:   Services

Step 2 will help you estimate the number of Crystal Reports Server services required to support the load (user activity) estimated in Step 1. This section of the sizing and configuration guide outlines the specific nature of each Crystal Reporst Server service as determined through various internal testing and benchmarking activities.

The Crystal Reports Server suite consists of many core services. It is during this step that you will determine which services are required and how many are needed to provide optimal performance.

Crystal Reports Server 2008 architecture differs from previous versions. Server Intelligence Agent (SIA) is introduced to simplify the management of other services in the web-based Central Management Console (CMC). The SIA is configurable by the Windows client Central Configuration Manager (CCM). **Adaptive Job Server and Adaptive Processing Server** are host servers of publishing and program services.

Core Services

Server Intelligence Agent (SIA)

The SIA maintains server status according to the settings you specify in the CMC. It processes the CMC's requests to start, stop, monitor, and manage all servers on the node, and it also monitors potential problems and automatically restarts servers that have shut down unexpectedly. The SIA ensures optimal performance by continually monitoring server status information, which is stored in the CMS database. When you change a server's settings or add a new server in the CMC, the CMS notifies the SIA, and the SIA performs the task.

The SIA is automatically configured during installation, but you can change these settings through the CCM.

Memory Requirement:

Only RAM memory consideration is required to maintain the execution of SIA. The recommended RAM memory is 350MB.

Central Management Server (CMS)

The principle component of the Crystal Reports Server system is the Central Management Server . Its primary responsibilities include, but are not limited to:

   Management of users & user groups

   Management of object security

   Central system configuration

   Management of scheduled tasks

   Management of historical tasks. (e.g., success or failed instances)

   Management of system objects (i.e., Info Objects)

   License management

   System database management

   Name service

Only one CMS is permitted per Crystal Reports Server environment. This service is expected to support the maximum number of concurrent active users currently available by license.

Processor Requirement:

1 CPU for the CMS service is sufficient to accommodate up to the current Crystal Reports Server license limitations (250 CAL + 250 NUL).

Memory Requirement:

Memory usage is controlled (roughly) by the number of objects stored in the object cache. This is controlled by the Windows registry key called "MaximumObjectsToKeepInMemory" and specifies the maximum number

of objects that the CMS stores in its memory cache. Increasing the number of objects reduces the number of database calls required and greatly improves CMS performance. However, placing too many objects in memory may result in the CMS having too little memory remaining to process queries. The upper limit is 100,000 and the default setting is 10,000.

For CMS System Database memory requirements please refer to Step 4: System Database Tuning.

Crystal Reports Cache Server

The Cache Server stores report pages generated by the Page Server. By storing report pages in cache, the Page Server and/or database server does not need to be accessed each time the report is requested. The Cache Server is used for the viewing of Crystal Reports only, and when the user's preference is set to any viewer other than Interactive DHTML.

When a report is requested for viewing, the system will first check the Cache Server to see if there are any current available cache pages generated for that report. If there are pages, the Cache Server will send available report pages to the Web Application Server. If there are no cached pages, the Cache Server will request a Page Server to generate these pages.

The threshold that determines the number of Cache Server services required is the "maximum simultaneous processing threads" per Cache Server service. The number of "simultaneous processing threads" is equal to the number of Estimating Simultaneous Requests. A guideline for the maximum simultaneous processing threads per Cache Server service is 400

400 "maximum simultaneous processing threads" per Cache Server service

Processor Requirement:

1 CPU is required for every estimated 200 maximum simultaneous processing threads
Example 1:

If an environment must service 200 simultaneous Cache Server requests, based on the 400 maximum per Cache Server service one would require only one Cache Server service. And based on the 200 "maximum simultaneous processing threads" per processor, one would require a single processor.

1 CPU machine with one Cache Server service

Example 2:

If an environment must service 800 simultaneous Cache Server requests, based on the 400 maximum per Cache Server service one would require at least two Cache Server services. And based on the 200 "maximum simultaneous processing threads" per processor, one would require at least 4 available processors.

1 quad machine with 2 Cache Server services

Memory Requirement:

Estimate 1MB per simultaneous processing thread + 17MB base

Disk Requirements:

For the Cache Server service, sufficient hard drive disk space should be available for the generation of cache files as well as in the temp directory for the creation of temp files (C:\Program Files\Business Objects\BusinessObjects Enterprise 11\Data\machinename.cacheserver\). The location of Cache Server cache files can be specified from the Central Management Console. The highest volume of cache files will likely be the "encapsulated page file" or ".epf." A single .epf file represents one page of a Crystal report and the size of the individual file is variable dependent on the complexity of the report page (e.g., a single .epf file from the sample "Statement of Account" report is 88kb in size). A sufficient amount of disk space should be made available to support all potential cache pages generated at a given period.

File Repository Servers (FRS)

There is an Input and an Output File Repository Server in every Crystal Reports Server implementation.

The Input File Repository Server manages objects (Crystal Reports, program objects, Microsoft Excel files, Microsoft Word files, Microsoft PowerPoint files, Adobe Acrobat PDFs, rich text format files, text files, hyperlinks, object packages) that have been published to the system by administrators or end users (using the Publishing Wizard, the Central Management Console, the Import Wizard, or a Business Objects designer component such as Crystal Reports).

The Output FRS maintains all the instances that have been produced from reports, programs, and object packages that have been scheduled.

Repository location

NO TE	To optimize system performance on the File Repository Servers, network settings on Windows 2000 Server could be set to "Maximize Throughput for File Sharing". This will give a higher priority to file sharing applications.

One of each - an Input and and Output File Repository Server - is required. In larger deployments, there may be multiple Input and Output File Repository Servers, for redundancy. In this case, all Input File Repository Servers must share the same directory. Likewise, all Output File Repository Servers must share a directory.

Processor Requirements

The File Repository Servers require higher I/O resources (faster disk, network) and fewer CPU resources. When estimating the number of CPUs in the Crystal Reports Server system, the File Repository Servers are not considered.

Memory Requirements

The File Repository Servers will have little impact on system memory.

Disk Requirements

Enough disk space must be available to store files.   Typically the Output FRS will require more disk space than the Input FRS. The Output FRS maintains all the instances (with saved data) that have been produced from reports, programs, and object packages that have been scheduled, and as such will require proportionately more disk space. For both the Input and Output FRS, the amount of space required will vary from system to system, however, knowing the average file size and multiplying this by the number of projected instances will assist in estimating total disk needs.

Adaptive Job Servers

Event Server

The Event Server manages file-based events. When you set up a file-based event within Crystal Reports Server, the Event Server monitors the directory that you specified. When the appropriate file appears in the monitored directory, the Event Server triggers your file-based event; the Event Server notifies the CMS that the file-based event has occurred. The CMS then starts any jobs that are dependent upon your file-based event.

Processor and Memory Requirements:

The Event Server under normal enterprise usage is not a processing or memory intensive server and as such will not be weighted in the sizing process. If Event Server functionality is required, allocate this service into the system but do not estimate any additional CPUs for this service.

Processing Tier

Crystal Reports Page Server

The Page Server is primarily responsible for responding to page requests by processing Crystal reports and generating Encapsulated Page Format (EPF) pages. A single .epf file represents one page of a Crystal report. The Page Server retrieves data for the report from the latest instance or directly from the database (depending on the user's request and user's security level).

Specifically, the Page Server responds to page requests made by the Cache Server. The Page Server and Cache Server interact closely, so cached EPF (encapsulated page files) pages are reused as frequently as possible, and new pages are generated as soon as they are required. The InfoView portal takes advantage of this behavior by ensuring that the majority of report-viewing requests are made to the Cache Server and Page Server.

The Crystal Reports Page Server creates Page Server sub-processes. Each sub-process loads the Crystal Reports engine and then instantiates threads or print jobs as needed. With the Page Server, if an individual print job were to fail for any reason, only those threads contained in the Page Server sub-process would be affected. All other sub-processes within the Page Server service would be unaffected. In addition, individual sub-processes are shut down after so many requests and a new sub-process is started, if required, so as to maximize resource management.

Definitions used in this section:

- **Page Server Service** – service that manages sub-processes

- **Page Server Service Sub Process** – process responsible for managing report jobs

- **Report Job** – thread responsible for generating report pages requested by report viewers

- **Maximum Simultaneous Report Jobs** – the total number of report jobs that can be contained in a Page Server service

For the Page Server service, the number of Page Server sub-processes and the total Maximum Simultaneous Report Jobs are determined by the following default algorithm:

Max Simultaneous Report Jobs (threads) = (# CPU's) * (25)   [minimum of 50 on a single CPU]

Max # of Sub Processes = (Max Simultaneous Report Jobs) / 10 report jobs per sub-process (rounded up)

Max # of Processes = (Max # of Sub Processes) + 1 Parent Process (The maximum report jobs per sub-process is set at 10)

By default, the above algorithm is used to determine the Maximum Number of Simultaneous Report Jobs on a particular machine. The algorithm has been purposely tuned conservatively to favor reliability (lower number of Simultaneous Report Jobs per CPU) so as to work optimally in most reporting environments and configurations. The default can be overridden in the CMC (under Page Server properties) by changing the radio button from "Unlimited Jobs" to "Jobs limited to:"

The "Jobs limited to:" option gives the administrator the ability to increase or decrease the Maximum Number of Simultaneous Report Jobs that can run on a single Page Server service (parent process). Based on our internal testing, broad guidelines for this setting are outlined below within this section.

IMPORTANT	In Crystal Reports Server, only one Page Server service is required to run on the single machine. The Crystal Reports Server Page Server creates and stops new Page Server sub-processes on an as-needed basis. The Page Server detects the number of processors on the machine and will scale accordingly.
	For this reason, it is important to **not** install, and is **not** necessary to install more than one Page Server if the default setting of "Unlimited" is kept.

This Page Server design improves performance, reliability and manageability and will dynamically adapt to different loads (by creating and stopping Page Server sub processes as needed) within its available resources.

When using the default setting of "Unlimited", the Crystal Reports Page Server's **Maximum Number of Simultaneous Report Jobs*** maximum will be calculated as 25 x Number of CPUs, with a minimum of 50 (25 by calculation, but minimum is always 50)

> 1 CPU - Maximum Simultaneous Report Jobs = 50
>
> 2 CPUs - Maximum Simultaneous Report Jobs = 50
>
> 4 CPUs - Maximum Simultaneous Report Jobs = 100
>
> 8 CPUs - Maximum Simultaneous Report Jobs = 200
>
> * Maximum Number of Simultaneous Report Jobs is equal to the amount of user Estimating Simultaneous Requests

Shared Page Server machine

There may be limited cases where it is advisable to change the default setting from unlimited so as to throttle back the maximum number of Page Server Jobs created on a single machine.

The Page Server service can be changed from the default setting of "Unlimited" to a recommended range of 25 - 75 "Maximum Simultaneous Report Jobs" per available CPU. The setting of 25 - 75 can be adjusted higher or lower depending on the environment (e.g. report complexity, size, etc.) Values below 25 per CPU may be appropriate since other processes such as CMS, Job Server, etc are present on the machine and are sharing the same CPUs.

On-Demand (Live Data) vs. Saved Data Viewing (Pre-Scheduled Instance)

Live Data

On-demand reporting gives users real-time access to live data, straight from the database server. Use live data to keep users up-to-date on constantly changing data, so they can access information that's accurate to the second. For instance, if the managers of a large distribution center need to keep track of inventory shipped on a continual basis, then live reporting is the way to give them the information they need.

Before providing live data for all your reports, however, consider whether or not you want all of your users hitting the database server on a continual basis. If the data isn't rapidly or constantly changing, then all those requests to the database do little more than increase network traffic and consume server resources. In such cases, you may prefer to schedule reports on a recurrent basis so that users can always view recent data (report instances) without hitting the database server.

Saved Data

To reduce the amount of network traffic and the number of hits on your database servers, you can schedule reports to be run at specified times. When the report has been run, users can view that report instance as needed, without triggering additional hits on the database.

Report instances are useful for dealing with data that isn't continually updated. When users navigate through report instances, and drill down for details on columns or charts, they don't access the database server directly; instead, they access the saved data. Consequently, reports with saved data not only minimize data transfer over the network, but also lighten the database server's workload.

For example, if your sales database is updated once a day, you can run the report on a similar schedule. Sales representatives then always have access to current sales data, but they are not hitting the database every time they open a report.

CPU utilization and memory consumption will be relatively comparable between Live Data viewing and Saved data viewing, however; viewing Save Data reports on average will decrease viewing response times and increase throughput and system efficiency.

Crystal Page Server Data Sharing

The "Oldest On-Demand Data Given To a Client (in minutes):" setting controls how long the Page Server uses previously processed data to meet requests. If the Page Server receives a request that can be met using data that was generated to meet a previous request and the time elapsed since that data was generated is less than the value set here, then the Page Server will reuse this data to meet the subsequent request.

Reusing data in this way significantly improves system performance when multiple users need the same information. When setting the value of the "oldest processed data given to a client" consider how important it is that your users receive up-to-date data. If it is very important that all users receive fresh data (perhaps because important data changes very frequently) you may need to disallow this kind of data reuse by setting the value to 0. The default is always set to 0 meaning that all users will, by default, receive fresh data.

If Data Sharing can be used in a system, this can decrease the number of CPUs required to process (view) a report.

When to use Crystal Reports Job Server vs. Crystal Reports Page Server

The Crystal Report Page Server is designed to process a large set of smaller reports whereas the Crystal Reports Job Server is designed to process a smaller set of very large reports. Smaller reports are less complex and contain a smaller set of data. They are suitable for a large group of users to view as "on demand" reports (Live Data). Larger complex reports that must retrieve and process a very large set of data should be scheduled (Saved Data)

Processor Requirements

For sizing estimates based on number of simultaneous jobs per CPU it is recommended to use a range starting from the default 25 to a recommended maximum of 75
        1 Processor = 25 - 75 Maximum Simultaneous Report Jobs

Memory Requirements

Depending on the design of a report, the number of records retrieved from the database, memory requirements may vary. When a report is viewed and loaded into memory the report is decompressed and expanded up to as much as 40 times the original report file size (with saved data/retrieved records).

Example (minimum memory requirements on Page Server machine):
        500KB Report File Size (contains saved data) = 500KB * 40 (decompression ratio) = 20MB

        25 Reports * 20MB = 500MB of minimum memory required

IMPORTANT	It is NOT recommended that the /3GB boot.ini switch be used to increase the size of the user process address space from 2 GB to 3GB (therefore reducing the size of the system space from 2GB to 1GB) on a Page Server machine.
	When the Page Server is rendering reports a high degree of GDI work (Graphical Device Interface, a Windows standard for representing graphical objects and transmitting them to output devices) is being performed and most GDI memory is allocated in the kernel address space, which is at a real premium with /3GB

Physical Address Extension support (PAE)

Using the /PAE switch may increase memory access. The /PAE switch changes the addressing mode to allow the O/S to access more than 4GB of RAM. By using this switch, each process is still limited to 2GB of user addressable space, but now the system can have more of these large processes running at once.

Disk Requirements

For the Crystal Reports Page Server service, sufficient hard drive disk space should be available in the temp directory for the creation of temp files during report processing (C:\Program Files\Business Objects\BusinessObjects Enterprise 11\Data\machinename.pageserver\). The data from the database server is stored in these files until it can be saved and compressed in the report.

Hard drive access speed to the temp directory may have an impact on the speed at which a report processes

Optimize disk performances on: C:\Program Files\Business Objects\BusinessObjects Enterprise 11.5\Data\procSched\<machinename>.pageserver

 or the equivalent directory path on a Unix system

Crystal Reports Job Server

The Crystal Reports Job Server processes report files (.rpt) as requested by the CMS and generates report instances (versions of the report that contain saved data). To generate a report instance, the Job Server communicates with the database to retrieve the current data. Multiple Job Servers can facilitate larger scale deployments.

The Job Server has the ability to process both report files and packaged objects. Object packages are simply collections of Crystal Reports Server objects (e.g. report files) that can be grouped together and managed by the Crystal Reports Server system as a single object.

When you are scheduling reports, the reports are run through the Job Server. The Job Server service launches a Job Server child process so each report is run in its own process. A single Job Server service with 1 available CPU can optimally support 5 maximum jobs (processes). Also, a single Job Server service can be used to support up to 20 jobs across 4 CPUs. The default settings for maximum jobs will vary based on the complexity and size of reports, but the conservative setting of five simultaneous jobs per available CPU should handle any kind of report.

Processor Requirements

1 available CPU can optimally support 5 maximum jobs (processes) – higher or lower depending on report complexity and size

For batch schedule reporting, the number of CPUs required to support the Crystal Reports Job Server service is dependant on the number of concurrent Jobs processing, the complexity and size of reports, and the time window available to execute reports.

(# of reports) X (Average Report Process Time)

_____   = number of required CPUs

 (Time Window) X (# of simultaneous jobs per CPU)
          Example:

          How many CPUs are required to process 1000 reports taking 10 minutes each on average, that
          must be processed in a four-hour (240 min) time window?

(1000 reports) X (10 minutes)

_____   = 8.4 CPUs (round up to 9 CPUs - i.e. 2 Quad and 1 Single box)

(240 minutes) X (5 simultaneous jobs per CPU)

Memory Requirements

Depending on the design of a report and the number of records retrieved from the database, memory
requirements may vary. When a report is executing and loaded into memory, the report is decompressed
and expanded up to as much as 40 times the original report file size (with saved data/retrieved records).

Example (minimum memory requirements on Crystal Report Job Server machine):

          2MB Report File Size (contains saved data) = 2MB * 40 (decompression ratio) = 80MB

          5 Reports * 80MB = 400MB minimum memory required

IMPORTANT	It is NOT recommended that the /3GB boot.ini switch be used to increase the size of the user process address space from 2 GB to 3GB (therefore reducing the size of the system space from 2GB to 1GB) on a Page Server machine.
	When the Page Server is rendering reports a high degree of GDI work (Graphical Device Interface, a Windows standard for representing graphical objects and transmitting them to output devices) is being performed and most GDI memory is allocated in the kernel address space, which is at a real premium with /3GB

Physical Address Extension support (PAE)

Using the /PAE switch may increase memory access. The /PAE switch changes the addressing mode to
allow the O/S to access more than 4GB of RAM. By using this switch, each process is still limited to 2GB of
user addressable space, but now the system can have more of these large processes running at once.

Disk Requirements

For the Crystal Reports Job Server service, sufficient hard drive disk space should be available in the temp
directory for the creation of temp files during report processing (C:\Program Files\Business
Objects\BusinessObjects Enterprise 11\Data\procSched\machinename.reportjobserver\). The data from the
database server is stored in these files until it can be saved and compressed in the report.

Hard drive access speed to the temp directory may have an impact on the speed at which a report
processes.

Report Application Server

The Report Application Server (RAS) is very similar to the Page Server. It, too, is primarily responsible for responding to page requests by processing reports and generating EPF pages. However, the RAS uses an internal caching mechanism that involves no interaction with the Cache Server. The RAS also provides the API services to enable the creation and modification of Crystal via custom code. Additionally, RAS is used at the time of viewing or submitting schedule requests for reports containing dynamic prompts and cascading lists of values.

The default Maximum Simultaneous Report Jobs is set at 75 for each RAS service. This value, as with the Crystal Reports Page Server, may be adjusted according to anticipated load and available hardware resources. The guideline of 25 to 75 (default) Simultaneous Report Jobs per CPU is recommended. The ideal setting for your reporting environment however, is highly dependent upon your hardware configuration, your database software, and your reporting requirements. Additionally, a recommended guideline is to run 1 Report Application Server per CPU.

Example:

For each available CPU, run 1 RAS service with a setting of 25 to 75 Maximum Simultaneous Report Jobs (use 25 for Optimal Performance)

Processor Requirements

1 Processor = 25 to 75 Maximum Simultaneous Processing Jobs (use 25 for Optimal Performance)

Memory Requirements

Depending on the design of a report, the number of records retrieved from the database, memory requirements may vary. When a report is viewed and loaded into memory the report is decompressed and expanded up to as much as 40 times the original report file size (with saved data/retrieved records).

Example (minimum memory requirements on RAS machine):

500KB Report File Size (contains saved data) = 500KB * 40 (decompression ratio) = 20MB

25 Reports * 20MB = 500MB minimum memory required

IMPORTANT	It is NOT recommended that the /3GB boot.ini switch be used to increase the size of the user process address space from 2 GB to 3GB (therefore reducing the size of the system space from 2GB to 1GB) on a Page Server machine.
	When the Page Server is rendering reports a high degree of GDI work (Graphical Device Interface, a Windows standard for representing graphical objects and transmitting them to output devices) is being performed and most GDI memory is allocated in the kernel address space, which is at a real premium with /3GB

Physical Address Extension support (PAE)

Using the /PAE switch may increase memory access. The /PAE switch changes the addressing mode to allow the O/S to access more than 4GB of RAM. By using this switch, each process is still limited to 2GB of user addressable space, but now the system can have more of these large processes running at once.

List of Values Job Server

The List of Values (LOV) Job Server processes scheduled list-of-value objects. These are objects that contain the values of specific fields in a Business View. Lists of values are use to implement dynamic prompts and cascading lists of values within Crystal Reports. List-of-value objects do not appear in CMC or InfoView. For more information, see the *Business Views Administrator's Guide*.

The List of Values Job Server behaves similarly to the Report Job Server in that it retrieves the scheduled objects from the Input File Repository Server (FRS) and saves the instance it generates to the Output FRS. There is never more than one instance of a list-of-values object. On demand list of value objects are processed by the Report Application Server.

Processor Requirements

The number of CPUs required to support the LOV service is dependant on the number of concurrent jobs processing. By default the "Maximum Jobs Allowed:" is set to 5. For most environments this setting will not have to be changed and a single CPU is sufficient.

Memory Requirements

This list of values objects are "Group by" Crystal Reports. The file size of the list of values objects (stored in the output FRS) can be used to calculate memory requirements. When a report is processed and loaded into memory the report is decompressed and expanded up to as much as 40 times the original report file size (with saved data/retrieved records).

Example (minimum memory requirements LOV machine):

100KB Report File Size (contains saved data) = 100KB * 40 (decompression ratio) = 4MB

5 LOV Reports * 4MB = 20MB minimum memory required

Application Tier

Web Application Server

Depending on how the system is being utilized, the web application server (IIS, Apache Tomcat, BEA WebLogic, IBM WebSphere) can manage differing number of concurrent user sessions and simultaneous requests. The main functions of the web application server within the Crystal Reports Server system are:

Processing the .NET/Java script

Translating the Encapsulated Page Files (page on demand) to DHTML pages.

Communicating with Cache Server for report view requests

Managing session state information for the users

Processor Requirements

NOTE	Processor Requirement guidance is generalized and relative requirements may change based on the individual characteristics of different web application server types. To better understand specific web application server characteristics, please consult with the specific vendor.

One web application server is estimated to manage approximately 400 concurrent user sessions (user session = 1 logged on user) per processor. And, generally, a service can efficiently manage 100 simultaneous requests (request = e.g., a user clicking on a folder). Under normal circumstances, it is improbable that all concurrent users would make a request simultaneously; therefore, the following numbers allow for and differentiate between "concurrent user sessions" and "simultaneous requests." Because the service deals with two thresholds (Maximum number of Concurrent User Sessions and Maximum number of Simultaneous Requests), it is important to consider both when determining the required hardware.

This can be illustrated in the following examples:

Example 1

1 single available processor (1 processor) with 1 web application server service running could efficiently service 400 concurrent user sessions and can handle 100 simultaneous user requests.

Example 2

1 dual processor machine (2 processors) with 2 web application server services running could efficiently service the maximum permitted 500 concurrent user sessions and handle 200 simultaneous user requests.

Rule of Thumb

1 web application server = 400 concurrent user sessions (user session = 1 logged on user).

1 web application server = 100 simultaneous requests per processor (request = e.g., a user clicking on a folder)

As a base guideline, it is recommended to estimate 1 web application server per 100 simultaneous requests, however each viewer type will have its own characteristics that will impact the capacity for concurrent users or simultaneous requests and as such this number can be higher and lower.
Viewing in ActiveX Viewer

1 web application server = 100 simultaneous requests per processor (request = e.g. a user interacting with the report)

Viewing in HTML or .Net Webform Viewer

1 web application server = 50 simultaneous requests per processor (request = e.g. a user interacting with the report)

LiveOffice – Special Settings

Depending on the usage that is made of LiveOffice, the settings of the given services should be applied as described in the previous sections. For instances:

Where LiveOffice is used with Crystal Reports, one should follow the section on RAS.

Recommendations

Add one Report Application Server instance for every 50 to 75 LiveOffice/Crystal users.

The Report Application Server connection timeout should be optimally adjusted in order to prevent unused sessions remaining open too long a time. When the timeout value is too large the max connection may be overflowed. The timeout value, however, should be long enough to avoid active sessions being interrupted.

In a LiveOffice document, using a filter rather than an interactive prompt will provide better performance due to fewer interactions with the database.

Step 3:   Single Machine Configuration

Crystal Reports Server is licensed as a single server reporting solution. This basic configuration separates a Crystal Reports Server from from your web and database server, and installs all Crystal Reports Server services on a single machine. This grants the services their own set of processing resources, which they do not have to share with database and web server processes.

Step 4:   System Database Tuning

System Database Backend Performance

To provide optimal overall performance the network and the database system need to be adequately sized and configured. Tuning a database system for peak performance is a complex task and involves hundreds of database parameters. Different tuning methodologies have been developed and comprehensive performance tuning guides are available.

Connection between Crystal Reports Server and the database system

For data storage and retrieval, Crystal Reports Server exchanges messages with the database system over the network. The data caches ensure minimal network traffic between Crystal Reports Server and the database system. Nevertheless good network performance is critical, for example some operations cannot use Crystal Reports Server's data caches.

The database client software uses the underlying network system. There is no tuning required for the database client software.

Make sure network response times are adequate and network throughput is sufficient.

Performance Criteria

Meeting the following 5 criteria helps to avoid the most common performance bottlenecks.

1. The database system's cache hit rates are over 90%.
2. The optimizer statistics are not older than 24 hours.
3. Lock granularity is row locking.
4. There are no lock escalations.
5. There are no log write waits. The average disk write queue length to the disk drives which contain the database log files is smaller than 5.

1. 90% or higher cache hit rates…

give us

- Fewer physical disk reads and writes
- Fewer SQL compiler executions
- Improved disk subsystem performance

is important to avoid

- long disk read or disk write queues
- unnecessary SQL statement compilations or long SQL statement compilation times

is achieved by

- providing sufficient physical memory
- configuration of sufficient cache sizes

2. up-to-date optimizer statistics…

give us

- improved query performance as the Query Optimizer chooses the Query Execution Plan with the minimal cost

- good selectivity estimates of predicates in SQL expressions specially for expressions including columns with unequal or skewed distribution of column values

is important to avoid

- long response times and low throughput
- long execution times for queries specially for queries predicate has high selectivity
- full table scans
- lock escalations
- lock waits or deadlocks
- significant differences between the estimated number of rows for each operator in the query plan from the actual number of rows

is achieved by

- implementing a procedure to periodically update the statistics or to update the statistics after frequent changes to the database.
- creating all statistics (table, column, index) and the statistics are not stale (underlying object has not been modified significantly … more than 10% of the rows)
- setting the sample size (percentage of data that got analyzed to gather statistical information) to a value which is sufficient for the skewed data distribution
- setting the number of histogram buckets to a value which is sufficient for the skewed data distribution

3. lock granularity is row locking…

give us

- increased concurrent execution of transactions
- reduced transaction processing time due to less lock waits

is important to avoid

- lock waits or deadlocks

is achieved by

- setting the lock granularity of your database to row level.

4. no lock escalations…

gives us

- increased concurrent execution of transactions

is important to avoid

- lock waits or deadlocks

are achieved by

- configuration of a sufficient size of the lock list
- setting the lock escalation threshold to an appropriate value
- keeping the optimizer statistics up-to-date

5. no log write waits…

gives us

- increased concurrent execution of transactions
- reduced transaction processing time due to no I/O waits

is important to avoid

- lock waits or deadlocks

is achieved by

- providing an I/O subsystem with sufficient throughput
- providing a Disk Controller Cache and setting the Disk Controller cache to 100% write cache

Calculating the Database File Size

Estimate

Each row stores one BOE XI object. The average size of a BOE XI object is 1024 Byte.

*(Size Data + Size Indexes) * (1 + (1 – page fill factor))*

*= ((Number Rows * 1024) + Number Indexes * (32 * number of rows * 2)) * 1.3*

*= 1.3 * number rows * (1024 + number indexes * 64)*

= 1.3 * number rows * 2624

Number of Objects	Database Size	Unit
10,000	33,312	KB
100,000	333,125	KB
1,000,000	3,331,250	KB
10,000,000	33,312,500	KB

Sybase (per one lob one data page is allocated)

*= ((number rows * 844) + (number rows * page size) + Number Indexes * (32 * number of rows * 2)) * 1.3*

= 1.3 * number rows * (2624 + page size)

Number of Objects	Database Size (page size 4K)	Unit
10,000	87,360	KB
100,000	873,600	KB
1,000,000	8,736,000	KB
10,000,000	87,360,000	KB

Related Content

- IBM DB2 8.1 Manuals "Administration Guide: Performance"
- IBM Redbook "DB2 UDB ESE V8 non-DPF Performance Guide for High Performance OLTP and BI"
- Oracle Manual "Performance Tuning Guide and Reference"
- Oracle Manual "Supplied PL/SQL Packages and Types Reference"

Copyright