crystal decisions™

Crystal Enterprise

Scalability Testing

Overview

Crystal Enterprise (CE) is a complex system of services that provide specific functionality. There is no single correct architecture that is appropriate for all environments. This document is intended to provide an overview of CE Architecture, a general description of the primary services, and some guidelines regarding the relationships between those services.

This document is written for both a conceptual view of CE architecture and a detailed reference of CE services. However, this document is not intended to be used as the sole reference guide for conducting scalability testing and assumes a high level of familiarity with CE and related terms.

The information in this document applies to CE version 8.5.

IMPORTANT	No amount of documentation is a substitute for the experience of a skilled consultant. Crystal Decisions Professional Services should always be involved in any significant test or deployment. Professional Services can be reached at 1-800-877-2340.

Contents

Introduction

This document contains information on the following topics:

- CE Services
- Service Command Line Arguments
- CE Architecture
- CE Load Testing
- Common Scalability Testing Issues

Crystal Enterprise Services

Because CE implements a number of services, this section lists and describes the roles of those services relevant to scalability and volume testing.

APS – Automated Process Scheduler

The APS is responsible for user authentication, keeping track of system objects (including folders, report objects, etc), controlling user access to objects, and controlling system services (i.e. assigning reports for processing on a given server). The APS keeps persistent track of system objects in a System Database.

The APS is commonly a memory intensive service and is not processor intensive.

PS – Page Server

The Page Server has two key roles.

- During on-demand viewing the Page Server is responsible for getting the report object from the Input File Repository Server (FRS), establishing a connection with the database, querying the database for an appropriate set of data, formatting the result set, parsing the report into individual pages, and passing these individual pages to the viewer through the Cache Server. By parsing the report into individual pages, significantly less network bandwidth is used.
- While viewing an instance of a report, the Page Server is responsible for getting the report instance from the Output FRS, expanding the report, formatting the report, parsing the report into individual pages, and passing these individual pages to the viewer through the Cache Server. Again, by parsing the report into individual pages, significantly less network bandwidth is used.

NOTE	A report instance is a report with saved data that may be viewed an infinite number of times with only one database query being executed. Prime candidates for report instances are month end reports. For example, a month end report can be generated on the last day of the month and be ready the next morning for thousands of viewers.

The Page Server is a processor intensive service and uses memory according to the size of reports being processed. Large reports (that is reports with large

datasets and many graphics/charts) may use significant memory. As a general rule, before compression or after expansion, a report in memory is commonly between 15 and 20 times larger than the corresponding instance in the Output FRS. This is an important point to keep in mind when deciding on disk space.

JS – Job Server

When the APS requests a report from the Job Server, the Job Server is responsible for getting a report from the Input FRS, establishing a connection with the database, querying the database for an appropriate set of data, packaging the report object (report template) with a compressed set of data for the report, writing that back to the Output FRS, and notifying the APS of the location of the report instance.

The Job Server is a processor intensive service and uses memory according to the size of reports being processed. Large reports (that is reports with large datasets and many graphics/charts) may use significant memory. As a general rule, before compression or after expansion, a report in memory is commonly between 15 and 20 times larger than the corresponding instance in the Output FRS.

CS – Cache Server

The Cache Server is responsible for maintaining a set of the last viewed reports. Reports are retained on a LUFO (Last Used First Out) basis. A properly configured Cache Server can provide significant benefit. Multiple attempts to view the same on-demand report within a specified time frame can be handled with one database connection and the resulting report can be served to a significant number of users.

If you database is highly transactional, users need to see up to the second changes in the data, you may choose to reduce the benefit of the Cache Server service by reducing the default timeout of 20 minutes. This increases the frequency of database hits and therefore returns more up to the minute data.

Conversely, databases that are not highly transactional (those which are updated every few hours, nightly, or even less frequently) can leverage this service by increasing the default timeout and expanding the amount of disk space available to this service. This service is tied very closely to the Page Server.

The Cache Server can be memory intensive but is not processor intensive.

FRS – File Repository Server (Input and Output)

The FRS is the repository of reports – both objects and instances. Report objects are often referred to as report templates whereas instances are report objects with a snapshot of data from a specific time. A report instance may be a month end report where one copy of the report is generated on the last day of the month and the resulting report template and data are kept for an extended period of time. Viewing a report instance does not require a connection to the database (although one was created when the report was generated) and changes to the database that occur after report processing are not reflected in the report. It is common for a number of instances per report object to be generated. For example, there could be 24 instances of a month end report showing a two-year history.

The FRS is neither memory nor processor intensive. However, if many objects and instances are created they may consume significant disk space. As users delete objects, the APS will manage the deletion of the underlying files. Instances may have criteria applied to them for automated deletion including deletion of reports over a fixed age for example, 24 months.

WCS – Web Component Server

The Web Component Server (WCS) is responsible for processing Crystal Server Pages (CSP). When a CSP page is requested the Web Server hands this request to the WCS for processing. The WCS passes authentication information such as user name, password, and authentication to the appropriate APS. This APS returns a token to the user to use for authentication. Then the WCS retrieves a list of folders and objects from the APS that the user has rights to see. It converts this information to HTML and passes this HTML back to the user through the Web Server. CE supports zero-client viewers that pass only HTML to the viewer saving them from being forced to download any ActiveX files or plug-ins.

The additional load of having the WCS convert all viewed report pages to HTML may cause the WCS to be a processor intensive service.

WS – Web Server

The Web Server should only be responsible for redirecting .csp pages to process on the WCS and then passing HTML returned from the WCS back to the user to be rendered in his/her browser. This process allows for Web Servers that support CE to have minimal additional load placed on them by the CE system. The only CE component that is required to be installed on the Web Server is called the Web Connector; it is responsible for the redirect of the .csp pages.

Crystal Enterprise Architecture

A logical diagram of the relationship between the CE services is drawn below. Note that these services may all exist on one server for small implementations or may be spread out over many servers with multiple copies of services being run within a given system. This particular diagram does not demonstrate any high availability options as only one copy of each service is listed.

For the purpose of having a highly available system, each service can be installed on more than one server so that if one server fails, another can accept the load. For the purpose of expanding volume, each service can be installed on more than one server or multiple copies of a single service can be installed on a given server (provided it has the memory and processing power to handle the additional load).

For detailed information on high availability, go to http://support.crystaldecisions.com/docs and search for the file name ce85_high_availability.pdf.

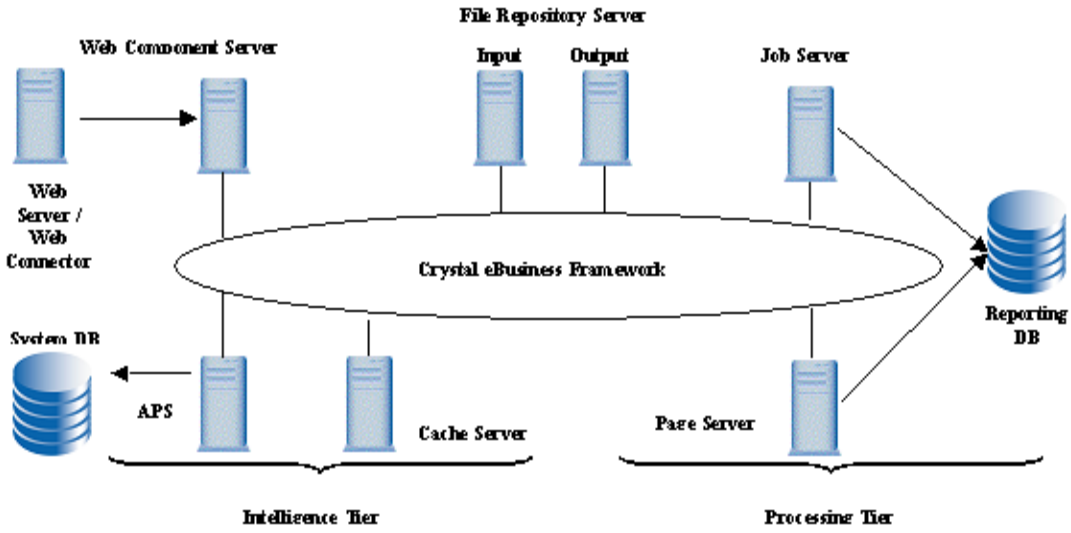

Figure 1

Although not shown in Figure 1, the Web Server is often placed inside a Demilitarized Zone (DMZ), that is, the Web Server is flanked on both sides by Firewalls. Communication to the Web Server from the outside world is done through a port in the Firewall and communication from the Web Server to the WCS is also carried through a (configurable) port.

For detailed information about configuring CE with Firewalls go to http://support.crystaldecisions.com/docs and search for the file name ce8_firewall_support.pdf or ce85_firewall_support.pdf.

Additional recommendations:

- Because of the close relationship between the Cache Servers and Web Component Servers, they should be located on the same server(s).
- Because the Page Server and Job Server commonly share the same database connections they often run on the same server(s). Report instances should be scheduled to run during times of reduced system load to allow for maximum throughput of on-demand reporting. In some environments, Job Servers are disabled during daytime operations to eliminate the possibility of inadvertent use. Report processing is consistently the heaviest load placed on the servers.
- The WCS requires significant processor time as it processes the interface for all CE related views (report output, directory listings, report object listings). To ensure available CPU, time it is often beneficial to place the WCS on a server separate from the Job Server and Page Server. Better performance may be obtained on two dual processor servers rather than on one quad processor server.

Service Command Line Parameters

The following reference of CE command line settings should only be manipulated by a trained CE consultant.

Crystal Management Console settings:

- **Maximum Simultaneous Processing Threads** on the page server should be set to no more than 50.
- The sum of all **Maximum Simultaneous Processing Threads** on Cache Servers must equal the sum of all **Maximum Simultaneous Processing Threads** on the Page Servers.

Crystal Configuration Manager Command line settings:

These command line settings are case-sensitive so if a setting is misspelled or case incorrect, that setting will be ignored.

Cache Server Settings:

-desiredsessperrpt 1

This setting relates to memory management. This number must be the same on both the Cache Server and Page Server.

-maxDBResultRecords 0

Page Server Settings:

-maxDBResultRecords 0

This setting forces CE to get all records and override any settings that may cause it to return less than the full recordset. Setting this number to a value greater than zero causes the service to return only that many records. For example, **maxDBResultRecords 500** causes the service to only return a maximum recordset of 500 records.

Job Server Settings:

-maxDBResultRecords 0

Web Component Server Settings:

-defaultSessionTimeout X

Where X is the number of minutes that you want the session to last (then time out). Setting this number to greater than 20 has no impact as 20 is the largest possible setting.

APS Settings:

-ndbqthreads 10

This setting defines the number of threads the APS will use when writing updates to the system DB.

-maxobjectsincache 100000

This setting defines how much information is kept in memory by the APS. The more available memory the APS has, the more information can be kept in memory. Similarly, an APS that shares memory with a database server, Page Server, Job Server, Cache Server, or other memory intensive application will have less available memory and therefore this number should be reduced. Considering the dynamic nature of the considerations that determine how much memory is available to the APS, the best way to determine if this number is effective is to observe available memory on the APS when under load and reduce this number if memory use stays consistently high.

Common Scalability Testing Issues

Load testing script settings

Trapping and excessively using a single APS Logon Token.

This token is generated when a user is initially authenticated by the APS. The token is routinely created with an expiry date and limited number of available uses. When either the number of uses has been consumed or the date has expired, the token is no longer valid. A script that traps and excessively uses this token will incorrectly exhibit a high number of logon or viewing errors.

Trapping and reusing Object/Instance IDs.

CE IDs are routinely created and destroyed by the APS. IDs should be queried for as part of the browsing process and not stored long term. Incorrect use of the IDs will result in an excessive number of errors when trying to view this specific object.

Phantom servers listed in registry

In the registry on all servers, the following key should be reviewed:

HKEY_LOCAL_MACHINE\SOFTWARE\Crystal Decisions\8.5\Enterprise\APSClusterMembers

Sometimes additional servers are listed in error. For example, there should be only two APSs in a cluster but five servers are listed in the registry key. Having such phantom servers listed can hinder CE performance. This may occur as servers move around or if every component is installed on every CE Server while only specific services are intended to run on these servers.

Page Server thread timeout setting

If this timeout is set either too low or too high, issues can arise. When timeout is set too high, the threads can run out and when it is set too low, they can expire

before the database has had a chance to return a result set. This situation can be difficult to identify, as the responsiveness of the database will impact this scenario.

Load Runner specific settings:

Other issues occur when load-testing software is not configured correctly to simulate actual system use. This can lead to incorrect results.

Tools such as Load Runner are often used to load test CE. The following figures outline suggested Load Runner settings based on the settings specified in the Service Command Line Parameters sections of this document.

IMPORTANT	These settings are suggestions only. If you are not comfortable performing these tests yourself or are not satisfied with results you are receiving through load testing, please contact Crystal Decisions Professional Services at 1-800-877-2340 for assistance. Crystal Decisions Technical Support does not support load testing.

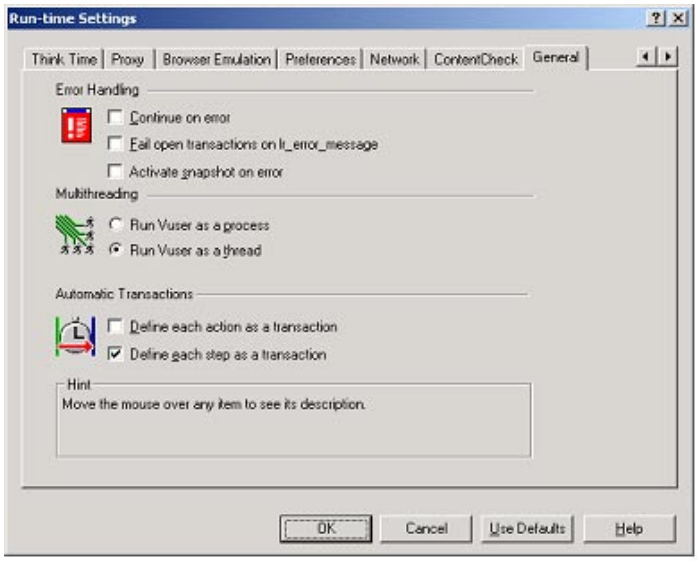

Figure 2 - Define each step as a transaction.

Browser Emulation:

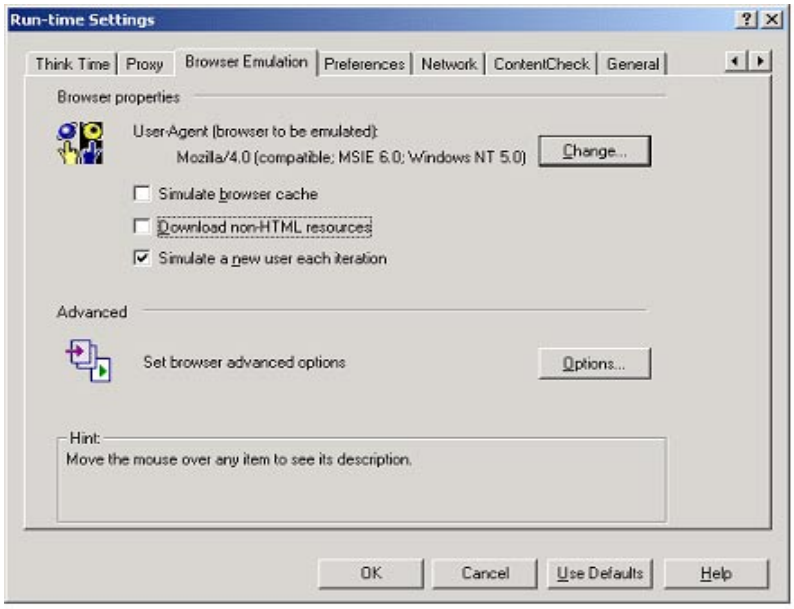

Figure 3 - Simulate a new user each iteration

IMPORTANT	The **Download non-HTML resources** setting cannot be saved with Script. It must be set during Scenario Testing.

Click the **Options** button from Figure 3 to get the following dialog box.

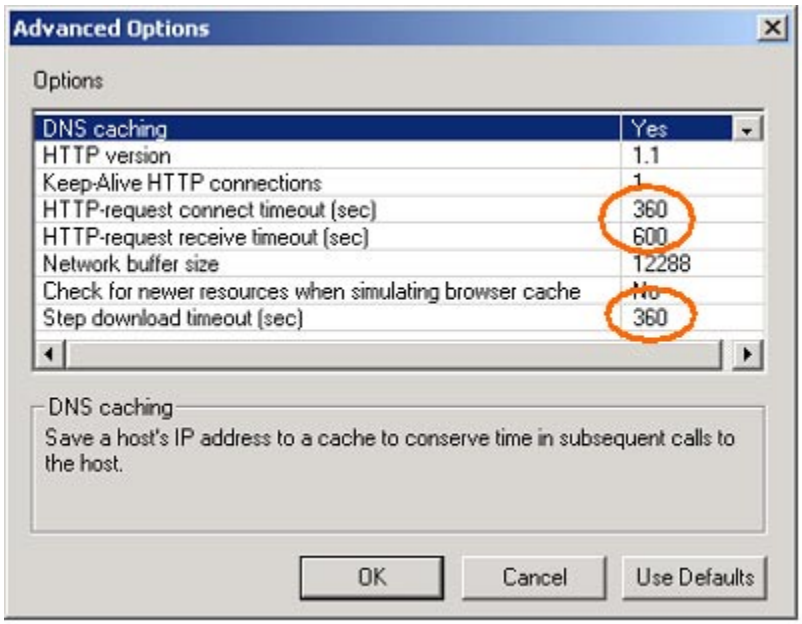

Figure 4

Contacting Crystal Decisions Professional Services

Find Regional Phone Numbers at:

http://www.crystaldecisions.com/contact/
by looking under the heading **Quick Reference Phone List**.

Get Service Offering Details at:

http://www.crystaldecisions.com/services/consulting/