Performance

Microsoft Dynamics CRM 2011 Data Load Performance and Scalability Case Study

White Paper

CONTRIBUTORS: Chris Brooks, Amir Jafri, Goro Hamamura , Ramy Yassa, Siri Harper, Shane Peckham, and Ami Pourteymour

Date: September 2012

Microsoft Dynamics

Feedback

To send comments or suggestions about this document, please click the following link and type your feedback in the message body:
http://go.microsoft.com/fwlink/?LinkID=265171.

Important: The subject-line information is used to route your feedback. If you remove or modify the subject line, we may be unable to process your feedback.

Table of Contents

Overview

Microsoft Dynamics CRM 2011 is designed to help enterprise organizations attain a 360-degree view of customers, achieve reliable user adoption, adapt quickly to business change, and accelerate project delivery and returns – all on a platform that provides enterprise levels of scalability and performance. This white paper provides details of bulk data load benchmark testing performed on behalf of a large financial services company.

The company's existing CRM solution stored customer data (including various related custody account information) in their host system, in which users during daily operations were frequently creating and updating customer records. To better accommodate business needs, the company wanted to set up a solution that would provide for management of their master customer data as well as data integration with their host system.

One of the first steps in the move to a new business solution involved migrating and synchronizing legacy customer data with the new system. To this end, the customer contacted Microsoft Services with a request to conduct benchmark testing of an on-premises deployment of Microsoft Dynamics CRM 2011. The testing was designed to ensure that the system could accommodate the need to create or update 10 million custody account records as part of a single batch job that would complete in a period of no longer than four hours.

To that end, the primary goals of the testing were to:

- Verify that bulk data load for both the initial data migration and for batch processing on fixed schedules could be achieved in a reasonable time.
- Estimate the time required to migrate 91.8 million customer records and their related custody entities to Microsoft Dynamics CRM 2011.
- Identify the hardware capacity required to meet the customer's data load throughput targets.

Results Summary

By using a parallel load process, a scaled out middle tier, and a highly-scalable database server, a high level of data load throughput was achieved, as is shown in the following table:

Insert Performance	Detail
Total Rows Loaded	10 Million
Total Time	2 hours 9 minutes
Inserts per second	1,288

Details of the hardware configuration against which testing was performed are shown in the following table:

Hardware	Detail
Virtual Web Server	8
Virtual Web Server Cores	32 (4 per server)
Database Server	32 cores, 256 GB RAM
Database Storage	SAN, across 21 disk spindles

Note: There are several ways to migrate data to Dynamics CRM, and the approach selected, combined with the tools used, available hardware, and configuration of software affect data load performance. This whitepaper provides an example of a data load scenario for a financial services customer to describe one approach and the results achieved.

Important: The results presented in this document reflect the scalability and performance of a specific on-premises implementation of Microsoft Dynamics CRM 2011 running in a particular test environment. Results will vary from implementation to implementation based on a variety of factors; no two enterprise implementations of Microsoft Dynamics CRM are the same.

Benchmark Environment

Software

The following software was used for the purpose of the benchmark effort:

- Microsoft Dynamics CRM 2011
- Microsoft Windows Server 2008 R2 Enterprise Edition
- Microsoft SQL Server 2008 R2 Enterprise Edition

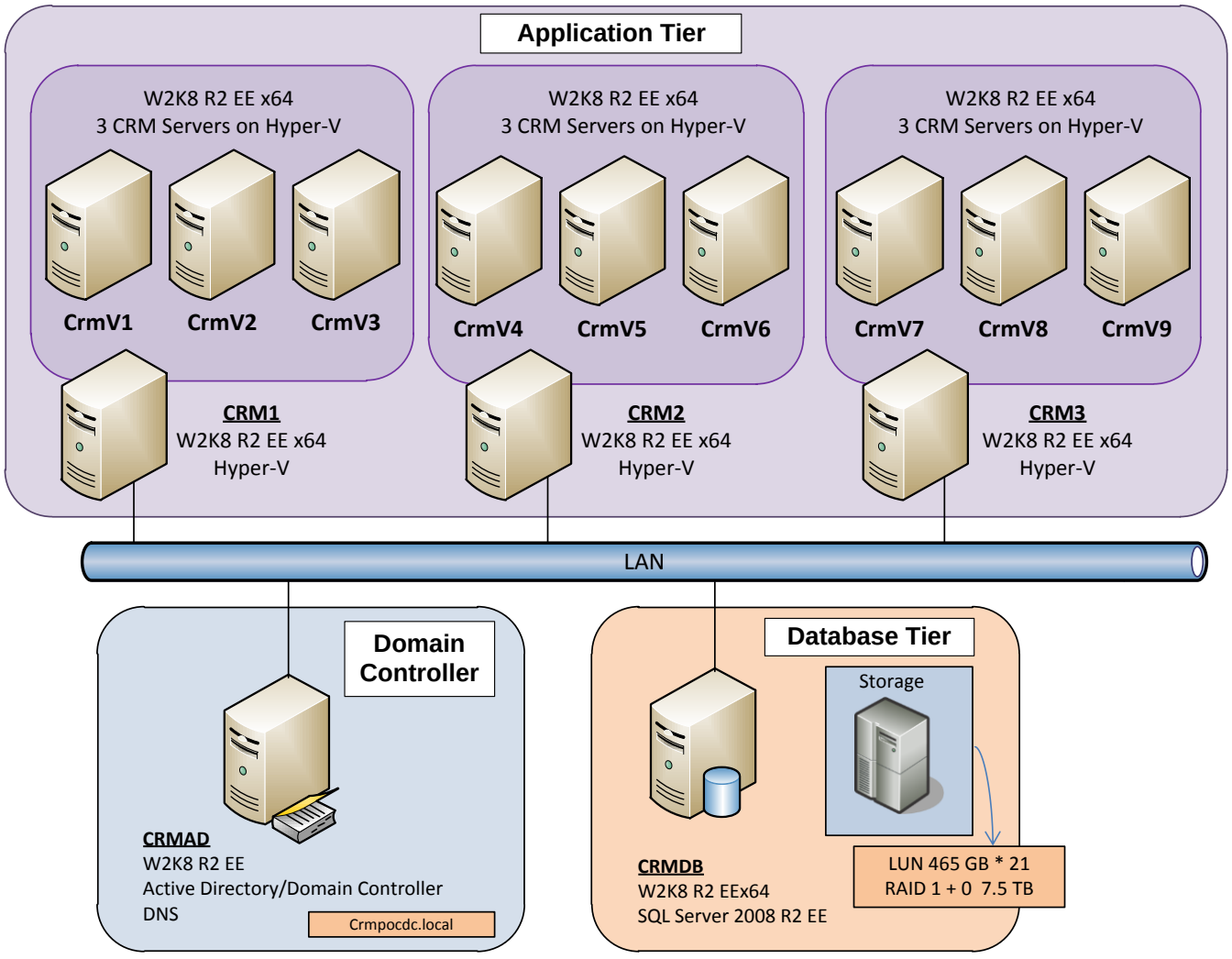
In addition, the data load application was developed by using Visual Studio 2010 Ultimate.

Hardware Environment

The application tier was deployed in a Web farm with nine application servers spread across three Hyper-V Servers. The database tier was deployed on a single server with an 8-socket, 32-core CPU and 256 GB of memory.

The hardware environment used for benchmark testing is shown in the following graphic:

Figure 1: Hardware environment



Details of the servers used in this environment are provided in the following table.

Table 1: Server configuration

Server	OS	Role	Model	Model
CRMAD	W2K8R2 EE x64	DC	HP Proliant DL380G6 Xeon X5560 2.80GHz 2QC	16Core 36GB RAM
CRMDB	W2K8R2 EE x64	SQL Server 2008 R2 EE	HP Proliant DK785 G5 Opteron 8356 2.3G 8QC	32Core 256GB RAM
CRM1	W2K8R2 EE x64	Hyper-V + SCVMM	HP Proliant DL380G6 Xeon X5560 2.80GHz 2QC	Host: 16Core 32GB RAM
CrmV1	Same as above	CRM Server 1	Virtual	Guest: 4Core 4GB RAM
CrmV2	Same as above	CRM Server 2	Virtual	Guest: 4Core 4GB RAM
CrmV3	Same as above	CRM Server 3	Virtual	Guest: 4Core 4GB RAM
CRM2	W2K8R2 EE x64	Hyper-V	HP Proliant DL380G6 Xeon X5560 2.80GHz 2QC	Host: 16Core 32GB RAM
CrmV4	Same as above	CRM Server 4	Virtual	Guest: 4Core 4GB RAM
CrmV5	Same as above	CRM Server 5	Virtual	Guest: 4Core 4GB RAM
CrmV6	Same as above	CRM Server 6	Virtual	Guest: 4Core 4GB RAM
CRM3	W2K8R2 EE x64	Hyper-V	HP Proliant DL380G6 Xeon X5560 2.80GHz 2QC	Host: 16Core 32GB RAM
CrmV7	Same as above	CRM Server 7	Virtual	Guest: 4Core 4GB RAM
CrmV8	Same as above	CRM Server 8	Virtual	Guest: 4Core 4GB RAM
CrmV9	Same as above	CRM Server 9	Virtual	Guest: 4Core 4GB RAM

Database Configuration

Data on the SQL Server was stored in *file groups*, which are primarily designed to group together the files that store a database object. Distributing a database object over multiple files can improve performance, especially if the files are stored on different disk drives.

To optimize performance during this benchmark effort, the CRM database was divided into six (6) database files stored on two different logical disks. The CRM database was also configured with a fixed size at 3 terabytes to ensure that an automatic expand operation did not occur during the data load process.

Database file locations are shown in the following table.

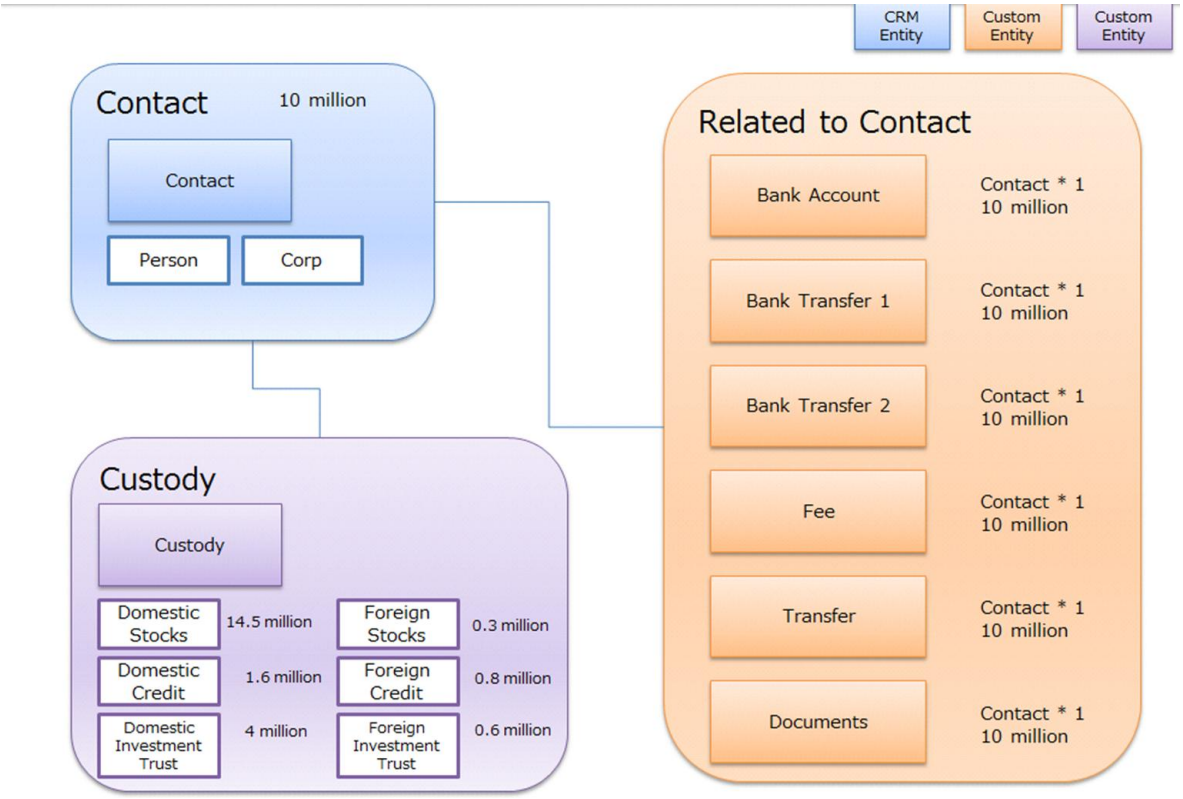
Table 2: Database file locations

Number	HDD	RAID	Drive	Capacity	Description	DB Size
1	Local	RAID 10	D:	173 GB	SQL Server Install Drive master, model, msdb DB File	
2	Storage	RAID 10	J:	2 TB	Database data file default location CRM DB Files 500GB * 3	1.5 TB
3	Storage	RAID 10	K:	1.57 TB	CRM DB Files 500GB * 3	1.5 TB
4	Storage	RAID 10	L:	2 TB	Database Log file default location CRM Log Files	Max 1 TB
5	Storage	RAID 10	M:	2 TB	Tempdb 10GB * 32 (matches the number of CPU cores)	320 GB

Microsoft Dynamics CRM Configuration

Each server was deployed with all CRM server roles installed with default settings. In addition, to accommodate customer requirements, Dynamics CRM was customized with the addition of seven new entities, all of which were related to the standard Contact entity. The relationship of these entities to each other is shown in the following graphic:

Figure 2: Entity relationships



The white boxes shown within Contact and Custody entities above indicate different types of records within an entity, which require separate data mappings and therefore also several data loads per entity. For example, a data load to Domestic Stocks would require different field mappings or different source files than for Foreign Stocks, and they must be imported separately into Custody entity.

Note: No complex customizations, plug-ins, or workflows were implemented for this benchmark.

Entity Definition

The following table lists the entities that were used in the benchmark and the number of fields per attribute type for each entity.

Table 3: Entity definitions

Entity	Description	# of records (million)	# of Attributes	# of each attribute type						
				Text	Optionset	money	Int.	Datetime	Two options	Lookup
Contact	Customer (2 types: Consumer, Corporate)	10	222	59	27	1	5	37	93	0
Custody	Customer's custody account by product (6 product lines)	21.8	100	39	15	0	21	11	13	1
Bank Account	Customer's main bank account	10	13	5	5	0	0	2	0	1
Bank Transfer 1	Customer's bank transfer account 1	10	21	10	9	0	0	1	0	1
Bank Transfer 2	Customer's bank transfer account 2	10	12	4	7	0	0	0	0	1
Fee	Fee types based on product, rank, and contact channel	10	18	4	7	5	0	1	0	1
Transfer	Account transfer type	10	7	3	2	0	0	0	1	1
Document	Holds document types per product	10	10	3	3	0	0	2	1	1

Overview of Data Load Volume

Based on customer requirements, a minimum of 10 million records per entity were loaded into Dynamics CRM in this benchmark. Total size of the source data was 31 GB. Entities involved were Contact (standard CRM entity) and seven custom entities as shown in the table below.

Table 4: Source files sizes

No	Entity Name	Type of record within entity	Columns	Max Length	Data Counts	Data Size
1	Contact	Contact Information	172	1,184	10 million	11.8 GB
2	Bank Account	Contact Bank Account	12	99	10 million	1 GB
3	Bank Transfer 1	Contact 1 Bank Transfer	20	225	10 million	2.3 GB
4	Bank Transfer 2	Contact 2 Bank Transfer	11	54	10 million	0.5 GB
5	Fee		17	112	10 million	1.1 GB
6	Transfer		6	28	10 million	0.3 GB
7	Documents received history	History of documents received	9	56	10 million	0.6 GB

No	Entity Name	Type of record within entity	Columns	Max Length	Data Counts	Data Size
8	Custody	Domestic Stocks	99	637	14.5 million	9.2 GB
		Domestic Credit			1.6 million	1 GB
		Domestic Investment Trust			4 million	2.6 GB
		Foreign Stocks			0.3 million	0.2 GB
		Foreign Credit			0.8 million	0.5 GB
		Foreign Investment Trust			0.6 million	0.4 GB

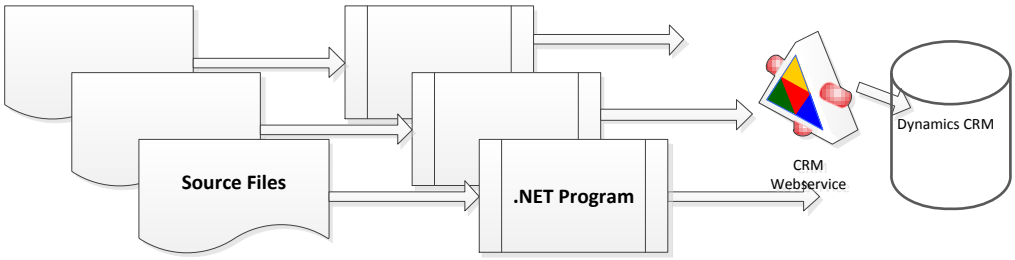
Tuning and Optimizations

Microsoft SQL Profiler was used to identify queries that were executed frequently during data loads, and this information was then analyzed by SQL Tuning Advisor. Custom indexes were implemented based on the Tuning Advisor results. There were no changes to the out-of-box indexes created by Dynamics CRM.

Data Load Methodology

The method selected and used to perform the data load process is fully supported as it leverages Dynamics CRM Web Services. A custom .Net application was written to split the source data into smaller chunks, and then multiple processes were run to load the data into Dynamics CRM. The .Net program was deployed on each of the nine Dynamics CRM Application server so that the data load could be run on an entity by entity basis. ETL tools such as SQL Server Integration Services could be used to implement a similar approach.

Figure 3: Parallel Data load process per entity



Details of the data load process are presented in the following sections.

Flat File Processing

First the data was divided into n flat files (n = total number of records divided by number of processes/server * number of CRM Application servers).

These are the steps for flat file (CSV) generation:

- Flat File Parsing
  - Generate first row in the source file to contain column headings
  - Enclose embedded double quotes in fields
  - Data load application parses column by column. Missing delimiters in flat files causes multiple columns to be read as a single column

- Preparing Data Files for Import
  - Put records for each entity in separate files
  - Add GUIDs to records that are related to one another to avoid GUID retrieval during the data load
  - Make sure that the first line of the data file contains column headings of the corresponding entity whose data is contained in the file
  - Make sure data exists for all business-required fields
  - Make sure that the first column heading is not the name of a record type

Uploading of Data into Dynamics CRM from Flat Files

The following steps were performed in preparation for uploading data into Microsoft Dynamics CRM from flat files:

- All columns were correctly mapped (in the flat file header) with respective Dynamics CRM attributes
- All picklist values were properly mapped
- Users who would be record owners were created in Dynamics CRM
- Business Units were created in Dynamics CRM to accommodate incoming records
- All picklists and picklist values were created first, followed by the transaction data.

Uploading Data in Parallel

The custom application was executed with ten (10) processes on each server. The application itself was single-threaded, but ten instances of the application were running simultaneously on each of the CRM servers. The source files were loaded into separate data load processes and then run in parallel to ensure faster processing. This process occurred on a per entity basis, so that when the data for one entity was successfully loaded - the process for the next entity was manually started. In reality this should be automated by Job Scheduler.

For this environment, ten processes per server were identified as the ideal number of processes for best performance. Having greater than ten processes was tested but resulted in a CPU performance bottleneck on the CRM servers.

Note: For production data loads, it is recommended to load data into a staging database prior to data loading into Dynamics CRM. By using a staging database, you can convert the data into Dynamics CRM format and ensure data validity (such as checking for valid Parent/Child relationships and updating necessary keys as well as ensuring validity of data and checking of data lengths) for to help ensure a smooth data load process.

Benchmark Results

Data was loaded into the custody entity for these performance benchmark results. The data load job contained 100 fields of various data types as shown in table 3.

Note: Except for system configuration entities and entities storing data common to all of the Business Units in the system, all of the entities used in the test were user-owned to indicate that the data was owned by Business users working with the system every day and not by system users.

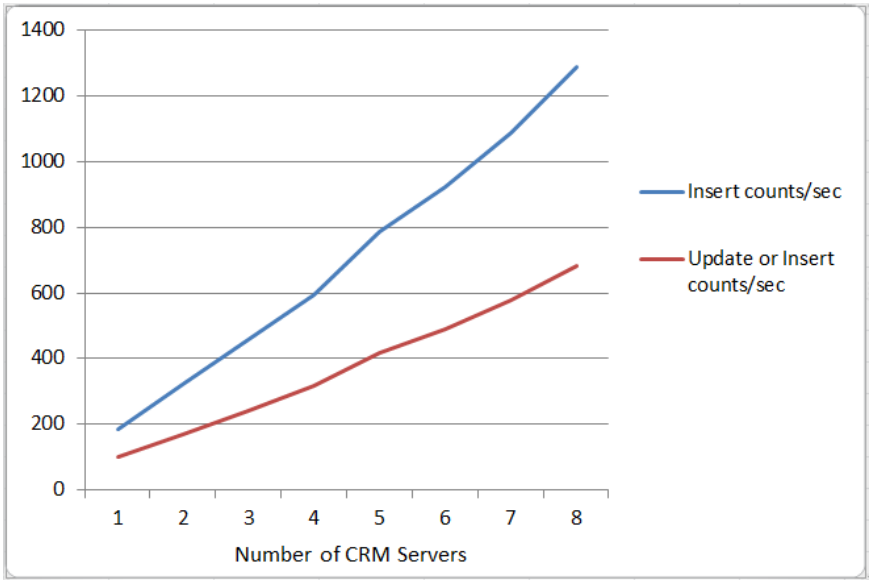
The table below shows measurement results record counts and processing time during data loading. Ten million records were inserted in the first measurement, and 7 million records were inserted with 3 million updates for the next measurement.

Table 5: Record counts/sec

Number of CRM Servers	1	2	3	4	5	6	7	8
Total Insert time	14 hrs 50 min	8 hrs 38 min	6 hrs 5 min	4 hrs 40 min	3 hrs 32 min	3 hrs 1 min	2 hrs 33 min	2 hrs 9 min
Insert counts/sec	187	322	457	596	785	923	1,088	1,288
Total Update/Insert time	27 hrs 54 min	16 hrs 16 min	11 hrs 28 min	8 hrs 47 min	6 hrs 39 min	5 hrs 39 min	4 hrs 48 min	4 hrs 4 min
Update or Insert counts/sec	100	171	242	316	418	491	578	683

The graphic below shows the data in the table above plotted in chart format:

Figure 4: Record counts/sec



Conclusion

The results of this benchmark illustrate how a Microsoft Dynamics CRM 2011 implementation can scale to support performance on large data loads. By scaling out and adding more CRM application servers and scaling up the database server (with processors, memory and IO capacity), data processing capacity can improve linearly.

Customers may be able to achieve even higher levels of performance and scalability via customization and a finer level of optimization.

SQL Server is the crucial component for Dynamics CRM, proper sizing and performance tuning of the SQL Server/database is very critical for performance. This is not limited to data loads, but also for daily operations and use.