Accelerating Microsoft® Office Excel® 2010 with Windows® HPC Server 2008 R2

Technical Overview

Published January 2010

Abstract

Microsoft® Office Excel® is a critical tool for business. As calculations and models become more complex, they require increasing amounts of time to execute on a desktop or portable computer. This white paper discusses different scenarios where Windows® HPC Server 2008 R2 can integrate with Office Excel 2010, which may help run Office Excel 2010 workbooks and critical user-defined functions (UDFs) significantly faster by enabling their execution in a Windows high-performance computing (HPC) cluster.

Dramatically reducing the calculation time of an Office Excel 2010 workbook gives business users and decision-makers more information in less time, enabling more thorough analysis, faster access to important information, and better-informed decisions. In addition to pure performance gains, running Office Excel 2010 workbooks and UDFs on Windows HPC Server 2008 R2 based clusters can provide benefits in flexibility, reliability, and efficient resource sharing.

CONTENTS

OVERVIEW

Microsoft® Office Excel® is a critical tool used by millions of users across all industry groups. With a wealth of statistical analysis functions, support for constructing complex analyses, and broad extensibility, Microsoft Office Excel 2010 is the tool of choice for analyzing business data.

As models grow larger and workbooks become more complex, the value of the information generated increases. However, more complex workbooks also require more time to calculate. For complex analyses, it is not uncommon for users to spend hours, days, or even weeks completing such complex workbooks. The problem this white paper addresses is how organizations can reduce the calculation time required for long-running workbooks to give users faster access to business-critical information.

One solution is to use Windows® HPC Server 2008 R2 to scale Office Excel 2010 calculations across multiple nodes in a Windows high-performance computing (HPC) cluster in parallel. This paper presents three methods for running Office Excel 2010 calculations in a Windows HPC Server 2008 R2 based cluster: running Office Excel 2010 workbooks in a cluster, running Office Excel 2010 user-defined functions (UDFs) in a cluster; and using Office Excel 2010 as a cluster service-oriented architecture (SOA) client.

Windows HPC Server 2008 R2 now enables running multiple instances of Office Excel 2010 in a Windows HPC cluster, where each instance is running an independent calculation or iteration of the same workbook with a different dataset or parameters. This solution allows near-linear performance increases for iterative spreadsheets, such as those running a Monte Carlo algorithm.

Running Office Excel 2010 UDFs in a cluster is a new ability of Office Excel 2010 for running complex or time-consuming UDFs—functions contained in Excel link libraries (XLLs)—in a Windows HPC Server 2008 R2–based cluster. If a workbook includes long-running UDFs, moving calculations to the cluster can result in significant performance improvements.

Using the Windows HPC Pack software development kit (SDK), Office Excel 2010 can function as a cluster SOA client to run complex and time-consuming calculations across a set of servers in a Windows HPC cluster. Any Microsoft .NET or Component Object Model (COM) application can use the Windows HPC Pack SDK: This paper uses Microsoft Visual Studio® Tools for Office (VSTO) to construct an Office Excel 2010 add-in that connects to the cluster as a cluster SOA client.

The sections that follow describe each solution and provide specific examples of how they can be used to improve the performance of long-running Office Excel 2010 workbooks.

HIGH-PERFORMANCE COMPUTING

High-performance computing uses a cluster of servers to split complex computations into smaller tasks that can be calculated in parallel on multiple servers. A Windows HPC cluster consists of several networked servers (see Figure 1).

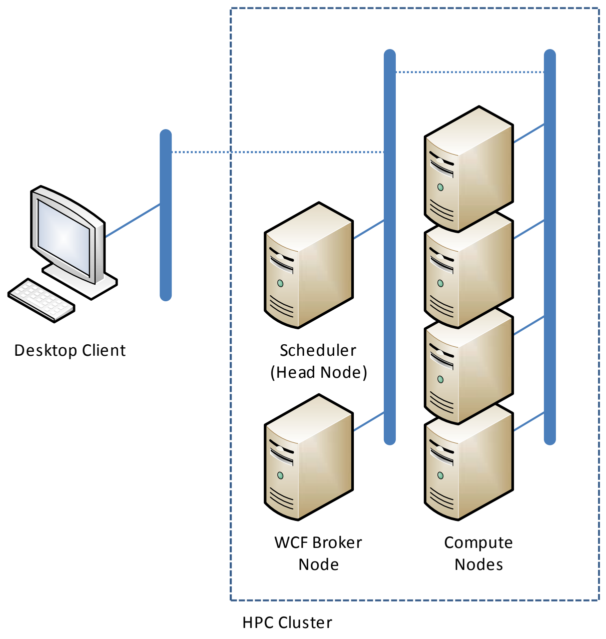


Figure 1. Windows HPC cluster topology

Each server in the cluster performs one or more specific roles. A Windows HPC cluster uses a Scheduler (hosted on a Head node*)* to manage computation tasks and Compute nodes (that is, servers) to perform the actual work.

In a typical cluster calculation, a client computer—such as a desktop computer running the Windows operating system—connects to the Scheduler over the network to submit a job. The Scheduler handles the work of managing the Compute nodes, locating available resources, and running the computation. As the Compute nodes complete units of work, the Scheduler sends calculation results to the client computer.

Windows HPC Server 2008 R2 also supports one or more Windows Communication Foundation (WCF) Broker nodes, which enable running WCF services in the cluster. WCF enables and simplifies building service-oriented applications, and Windows HPC Server 2008 R2 supports an SOA model that enables running WCF services in a Windows HPC cluster. Taken together, WCF and SOA provide a simple framework for building service applications to run in a cluster.

Using the SOA model, developers can build a cluster application as a library in any language that runs on the Microsoft .NET Framework, such as C# or Microsoft Visual Basic®.NET. WCF removes the complexity of managing and hosting services, so the WCF library need only provide calculation functions. In a Windows HPC cluster, the WCF Broker uses the cluster Scheduler to manage compute resources and perform the actual calculations.

The complete list of HPC features is too long to provide here. However, here are a few key benefits that make Windows HPC Server 2008 R2 an excellent platform for running Office Excel 2010 calculations:

- **Reliability.** A complex Excel workbook that calculates in hours or days on a client computer is vulnerable to accidents, power outages, and hardware failures. In an HPC cluster, if an individual compute node fails, calculations can be redirected to other nodes.
- **Flexibility.** Compute nodes can be added to a Windows HPC cluster at any time to increase available computing power. As users, calculations, and utilization increases, the cluster can be scaled as needed to meet the needs of an organization. Also, resources available to a single user can grow and shrink with priority using the job scheduler that comes with Windows HPC Server 2008 R2.
- **Efficient resource sharing.** With a Windows HPC cluster installed on an organization’s network, all authorized users in the organization can take advantage of the cluster. A single cluster can support multiple users and Office Excel 2010 instances while providing a central point for resource management and, overall, more optimized resource sharing across the organization.

GETTING THE BEST PERFORMANCE

The key to any HPC solution is parallelization. *Parallelization* refers to splitting a complex computation into component parts that can be run simultaneously (or *in parallel*) on multiple servers.

To run an Office Excel 2010 workbook in a Windows HPC cluster, some part of the calculation must be able to run in parallel. The actual performance improvement in the application will depend on how much of the calculation can be parallelized, the overhead involved in the cluster calculation, and the size of the cluster and its available resources. Not all long-running Office Excel 2010 workbook calculations can be exported to the cluster. Some Office Excel 2010 functions, such as creating a PivotTable or sorting a large data set, cannot be effectively parallelized and are therefore not suitable for calculation in the cluster.

An important first step in designing a cluster solution is identifying why a particular workbook calculation is long running. The examples in the sections that follow highlight common characteristics of long-running workbooks that lend themselves to cluster calculation.

UNDERSTANDING INDUSTRY CHALLENGES

Scenario 1: Insurance Reserve Requirements

In the insurance industry, firms must calculate reserve requirements on an ongoing basis to satisfy regulatory requirements and to understand their financial position. Reserve requirements calculation involves a thorough analysis of outstanding policies that a firm issues or owns and includes complex tax, asset, and actuarial factors that must be calculated for each policy.

Insurance firms often perform this calculation in Excel. Excel provides an excellent framework for constructing complex policy analyses, managing data, and creating reports. Many non-Microsoft tools have been developed to support the industry and help construct reserve requirements models in Excel.

However, as the number of policies increases and firms incorporate more jurisdictional and regulatory factors into the calculation, the time required to perform the calculation increases. It is not uncommon for complex reserve requirements calculations involving tens of thousands of individual policies to take days or even weeks to finish.

Scenario 2: Financial Portfolio Modeling

Organizations throughout the financial services industry maintain portfolios of assets and need to understand the value of these assets on an ongoing basis. This scrutiny might be required to satisfy internal controls or outside regulatory requirements or simply to understand the financial position of the firm at any given moment.

Excel is a commonly used tool to model financial portfolios. With a wealth of statistical analysis functions as well as support for complex calculations, Excel can be used to generate comprehensive ongoing portfolio valuations. However, as portfolios become larger and more complex and the assets themselves require more complex mathematical analysis, modeling individual or aggregate portfolios of assets can become time-consuming. Calculating a model containing thousands of individual assets and securities may require minutes or even hours on a typical client computer.

The time required to calculate a portfolio model has a direct impact on a firm’s ability to manage assets, understand its financial position, and manage overall risk. As the time required to calculate the value of an individual or aggregate portfolio increases, the firm loses the ability to make critical business decisions and react to market conditions.

THREE SOLUTIONS FOR MOVING EXCEL CALCULATIONS TO THE CLUSTER

This section presents three solutions for moving long-running Excel calculations to a Windows HPC cluster.

Running an Office Excel 2010 Workbook in a Windows HPC Cluster

Windows HPC Server 2008 R2 now enables running multiple instances of Office Excel 2010 in a Windows HPC cluster, where each instance is running an independent calculation or iteration of the same workbook with a different dataset. Many complex and long-running workbooks run *iteratively*—that is, they perform a single calculation many times over different sets of input data. These workbooks might include intensive mathematical calculations contained in multiple worksheets, or they might contain complex Microsoft Visual Basic for Applications (VBA) functions.

When a workbook runs iteratively, the best option for parallelizing the calculation is to run the entire workbook in the cluster. In this model, individual calculations need not be split into component parts, but the overall calculation—generating the results from many individual calculations—can be run in parallel.

Every application that benefits from this solution has three parts: the *workbook,* a *service,* and a *client.* Office Excel 2010 and Windows HPC Server 2008 R2 must be installed on each cluster server. Microsoft Office Excel 2007 or Office Excel 2010 must be installed on the client computer. The client computer can run the Windows XP with Service Pack 3, Windows Vista®, or Windows 7 operating system. The next sections describe these components and show how they fit together.

The Workbook

The *workbook* refers to a standard Excel workbook. This solution runs multiple instances of Office Excel 2010 on cluster servers, meaning that it supports workbooks that use VBA or XLL add-ins or an Excel Add-in (XLA) as well as external resources (provided these resources are accessible from the servers).

In some cases, workbooks may need to be modified to work with this solution. When Office Excel 2010 runs on the server, it does not support user interaction. Windows HPC Server 2008 R2 includes a comprehensive pop-up manager that can handle occasional dialog boxes and pop-up messages, but it is not designed to support interactive Office Excel 2010 features: Users cannot create a PivotTable when running on the server, for example, because doing so requires user interaction.

When a workbook is used in this scenario, it is important to identify the input values and the output or result of the calculation. The input values might be cells within a worksheet in which the user enters a value, or they might be parameters to a VBA function. The output might be a second set of cells within a worksheet or the result of a VBA function. Identifying the workbook input and output values is important when developing the service, which will run on the cluster servers and execute the workbook calculation.

The Service

The *service* is a WCF service that controls the execution of the Office Excel 2010 workbooks on the cluster servers. The service starts Office Excel 2010, calculates a workbook, and returns results.

Windows HPC Server 2008 R2 offers an SOA model that enables running WCF services in the cluster. In a Windows HPC cluster, the WCF Broker node handles managing and hosting the service library. The Scheduler handles assigning and managing compute resources. From the standpoint of the developer, using WCF removes all the complexity of hosting and managing the service. The developer need only build the calculation functions.

When running Office Excel 2010 workbooks in a Windows HPC cluster, the WCF service manages the Office Excel 2010 process used to calculate the workbook. It provides a single method, which calculates the workbook and returns the calculation results.

Once the service has started Office Excel 2010 on a server, it can interact with the workbook in either of two ways to provides access to the Excel object model (similar to using Excel automation on a client computer): Using the Excel object model, the service can read from and write to the workbook, and it can trigger calculation of the workbook. For example, in a simple workbook, the service might write values into some input cells, recalculate the workbook, and then read the values of some output cells.

The Excel object model also supports calling VBA methods within the workbook. If the workbook uses a VBA method to run a calculation, the WCF service can call the VBA method with input data and return the result when the calculation is complete.

The Client

The *client* is a program that controls the overall calculation. It is designed to work with the WCF service, and it tells the service which workbook to calculate along with the parameters or options to use, and it receives results when the calculation is complete (see Figure 2).

The client typically runs on a user's client computer. Client programs can be written in any language supported by Microsoft .NET or COM, and the client can be a Windows application or a command-line program. (In the example given later in this paper, the client program is built into Office Excel 2010 using VSTO and C#.)

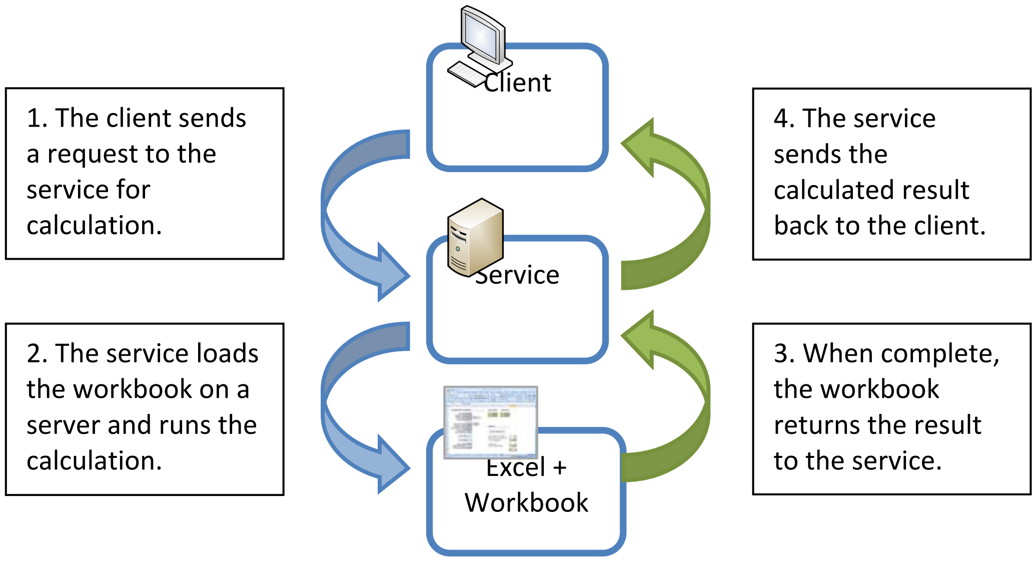

Client
Service
Excel + Workbook
1. The client sends a request to the service for calculation.
2. The service loads the workbook on a server and runs the calculation.
3. When complete, the workbook returns the result to the service.
4. The service sends the calculated result back to the client.

Figure 2. Application flow for a single service call of running workbooks on a cluster

<u>Strengths and Weaknesses of Running Office Excel 2010 in a Windows HPC Cluster</u>

Running Office Excel 2010 in a Windows HPC cluster has the following strengths and weaknesses:

- **Run multiple Office Excel 2010 instances on cluster servers.** Using this approach, users can run multiple Office Excel 2010 instances on the cluster, including support for VBA, XLL, and XLA. Although it does not support some user interface (UI) features of Office Excel 2010, many workbooks run in the cluster without modification, and most workbooks—including those that access remote databases or look up data in other workbooks—can be modified to run with this approach.
- **Run workbooks from Office Excel 2010, from a Windows program, or from a command-line program.** Using this approach, users can run a workbook calculation from programs that support Microsoft .NET or COM. For example, it is possible to build a solution in Office Excel 2010 using VSTO, from a stand-alone Windows program, or from a command-line program. However, some additional and customized UI activity is required in the client program, which may require changes to workflow and user processes and additional developments.

- **Pop-up window management.** Office Excel 2010 was designed as a desktop program. As a result, it incorporates user interaction, often in the form of pop-up messages and dialog boxes. If a long-running workbook displays a pop-up dialog box, the calculation will stop, costing processing time. Such stoppages can be difficult to debug if Office Excel 2010 is on a remote server.

  Windows HPC Server 2008 R2 integrates a pop-up manager that logs pop-up window events and can take custom actions to deal with them—anything from clicking **OK** in a dialog box to canceling the calculation and notifying the user.

- **Running workbooks in the cluster may require resource management.** In some cases, running Office Excel 2010 in the cluster may require specific resource management. If a user is accessing a database, for example, he or she must ensure that the database is accessible from cluster servers. If the workbook calls any non-Microsoft add-ins, the user must make sure that these add-ins are available on the servers, as well.
- **Cluster calculation involves some overhead.** Running a workbook in a cluster incurs some overhead. A fixed amount of time is involved in connecting to the job scheduler and starting Office Excel 2010 on each server. Also, there is variable overhead in running the calculation on each compute core, including network latency and additional processing as data records are passed to each process and results are collected.

When to Run Office Excel 2010 Workbooks in a Windows HPC Cluster

Run Office Excel 2010 workbooks in a Windows HPC cluster when:

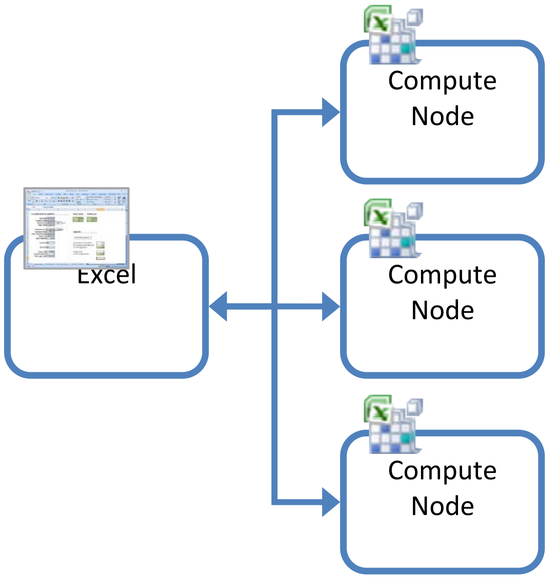
- There is a long-running workbook that calculates iteratively (that is, runs the same calculation many times over different sets of input data).
- The workbook uses VBA, macros, or complex calculations involving multiple worksheets.
- Excel users—not software developers—create, maintain, and modify their workbooks and Office Excel 2010 applications.
- A user would like to run a long-running Office Excel 2010 calculation from a separate application or script.

Running Office Excel 2010 UDFs in a Windows HPC Cluster

UDFs are a well-established mechanism for extending Excel. Functions contained in XLLs can be called from spreadsheet cells like any function in the standard Excel library.

Beginning with the 2007 Microsoft Office system, Excel provided support for multi-threaded recalculation (MTR)–enabled UDFs. Office Excel 2007 was the first Excel release to support multi-threaded calculation on client computers with more than one processing core. MTR-enabled UDFs can be run in parallel on multiple processing cores, resulting in better workbook performance.

Office Excel 2010 extends this model to the cluster by enabling Office Excel 2010 UDFs to run in a Windows HPC cluster (see Figure 3). In the cluster, UDFs work much like traditional UDFs, except that the calculation is performed by one or more servers. The key benefit is parallelization. If a workbook contains calls to long-running UDFs, multiple servers can be used to evaluate functions simultaneously. As far as users are concerned, there is no difference between a client computer function and a function running in the cluster—except better performance.



The benefit of running UDFs in a Windows HPC cluster is in *parallelization*.

When Excel recalculates, it can send a large number of UDFs to the cluster for processing at the same time, speeding up the overall calculation.

This process is transparent to the user. From the user standpoint, there is no difference in functionality—just better performance.

Figure 3. Running UDFs in a Windows HPC cluster

Strengths and Weaknesses of Running UDFs in a Windows HPC Cluster

Running UDFs in a Windows HPC cluster has the following strengths and weaknesses:

- **Existing XLLs require only minor modification.** UDFs in XLL files must be rebuilt to support calculation in the cluster. However, only a small change to the code is required.

- **UDFs run both on client computers and in the cluster.** The user has complete control over whether code executes on the client computer or in the cluster and can change this setting at any time.
- **No changes to the Excel user workflow.** To the user, functions running in the cluster are identical to functions running on the client computer. Existing spreadsheets do not need to be modified, and users can insert functions as they would normally with no change to process or workflow.
- **Only native code (XLL) functions are supported.** If an organization's UDFs are written in VBA, consider using the approach presented earlier to run the workbook entirely in a cluster. If the UDFs are written in a Microsoft .NET Framework language like C# or Visual Basic.NET, consider using Office Excel 2010 as a cluster SOA client.
- **Code must be recompiled.** To support running in a Windows HPC cluster, the XLL library must be rebuilt. Doing so requires that the source code and tools be available to recompile the library.
- **UDFs running in a Windows HPC cluster do not support the Excel application programming interface (API).** Because UDFs running in the cluster are not running within Excel, they do not have complete access to the Excel workbook or application. This means that Excel API functions are unavailable. For example, UDFs running in a Windows HPC cluster cannot resolve workbook references.

  However, many UDFs will work in the cluster without changes, and many more UDFs can be modified to meet these limitations. Provided that the UDF performs a single, atomic calculation—that is, it takes some parameters as inputs and returns a value as an output—there is a good chance it can be converted run in a Windows HPC cluster. In contrast, if a UDF needs to look up values in a spreadsheet or uses array or range references, it may not be a candidate for running in the cluster.
- **Cluster calculation involves some overhead.** Running a UDF in the cluster includes some network latency, because data is sent through the network to the cluster and results are returned to Excel over the network. If a UDF already executes quickly on a client computer, there may not be a benefit to running the calculation in the cluster.

When to Run UDFs in a Windows HPC Cluster

Run UDFs in a Windows HPC cluster when:

- There is a long-running workbook and much of the calculation time is consumed by UDFs in the spreadsheet.
- Functions are self-contained: They take arguments and return a value but do not otherwise look up data or make changes to the spreadsheet.

- There is an existing XLL containing UDFs that can be recompiled to support running in a Windows HPC cluster; or, a user has native code (C or C++) that he or she wants to call from Excel, and the user can build an XLL to provide UDFs.

Using Excel as a Cluster SOA Client

Windows HPC Server 2008 R2 provides an SDK for building applications that run in the cluster. Any language that runs in the Microsoft .NET Framework (like C# and Visual Basic.NET) or any language that supports COM (like C and C++) can use the Windows HPC Pack SDK to build cluster-aware applications.

Excel supports a number of frameworks for extensibility and development. Using the Windows HPC Pack SDK, users can build Excel applications that use the cluster for computation-intensive workbook tasks.

Strengths and Weaknesses of Using Excel as a Cluster SOA Client

Using Excel as a cluster SOA client has the following strengths and weaknesses:

- **Flexibility.** Many frameworks for extending Excel exist, and they offer tremendous flexibility in the applications and functions users can develop. Using these extension frameworks and the Windows HPC Pack SDK, users can construct a wide range of applications that use Excel as a cluster SOA client. Most services that run in the cluster can be integrated with Office Excel 2010.
- **Starting from scratch.** Although this option offers a lot of flexibility, that flexibility comes with some development cost. Building applications that connect to the cluster requires building an Office Excel 2010 interface and a cluster service. Doing so is not difficult for seasoned developers, but it may be outside the skill set of business users.

When to Use Excel as a Cluster SOA Client

Use Excel as a cluster SOA client when:

- A user has an existing application or cluster service he or she wants to integrate with Office Excel 2010.
- A user would like to build a complex or intensive calculation into Office Excel 2010 and can develop a cluster application or service.

EXAMPLES

This section presents example scenarios in which the various methods described in this paper are used to solve real-world problems. These example scenarios are based on actual customer implementations, but some details have been changed, and multiple actual implementations were combined to construct the examples.

Running an Office Excel 2010 Workbook in a Windows HPC Cluster: Insurance Reserve Requirements

An insurance reserve requirements calculation uses a complex Excel workbook. The workbook calculates more than 150,000 records but takes more than a week to finish.

The solution uses Windows HPC Server 2008 R2 and Office Excel 2010 to calculate the workbook in the cluster. Running the workbook in a 128-core windows HPC cluster, the calculation runs in just over two hours.

Overview

A large insurance firm uses Excel to calculate reserve requirements. This calculation uses a complex workbook incorporating tax, financial, and actuarial analyses for each policy. The workbook is designed to calculate with more than 150,000 individual policy records and generate aggregate and summary reports. Running on a client computer, the total calculation takes more than one week. The firm currently calculates the workbook on a monthly basis but would like to be able to get the same data weekly or even daily.

The workbook runs as a VBA application. To start the calculation, a user clicks a button on one of the worksheets. The VBA code first loads tax, financial, and actuarial data from a Microsoft SQL Server® database and creates worksheet tables containing the information. Next, the VBA code opens a text file containing the records to calculate. For each record, the VBA code sets various workbook cell values and recalculates the workbook. Calculation of a single record takes between 2 and 5 seconds.

When the calculation of a single record is complete, the VBA code reads values from the workbook representing the results of the calculation. These values are used to update summary data and are written to a report file. When the application has calculated the complete set of more than 150,000 records, the VBA code writes summary data to the report file.

Selecting a Solution

In this scenario, there is no single long-running function resulting in long overall calculation time. Rather, the calculation of each record is complex, incorporating a number of VBA functions and calculations in multiple worksheets. The workbook is calculated many times over a large set of input data, but individual records are calculated as a single unit, suggesting that the overall process can be parallelized by calculating individual records independently.

If a calculation is iterative over a large set of data and requires that the entire workbook be calculated for each data record, then the best solution may be to run multiple instances of the entire workbook in the cluster.

Structure of the Solution

As described earlier, every solution to run workbooks on a cluster has three parts: the workbook, which will be calculated in the cluster; a service, which is responsible for running Office Excel 2010 on the cluster servers; and a client program, which controls the calculation and provides a UI.

For this solution, the client is built into Office Excel 2010 using VSTO. With VSTO, developers can build a client program that uses Office Excel as the UI.

The service is a WCF service library that uses the SOA model in the Windows HPC cluster. WCF simplifies the design and development of the service, so the library only needs to include a single function to calculate the workbook. The service is designed to be asynchronous, meaning that individual calculations can be run simultaneously. (WCF automatically makes service functions asynchronous, so little extra effort is required from the developer.)

The workbook was originally designed to read records from a text file, calculate each record in a loop, and then write summary results back to the spreadsheet. The basic application flow of the original workbook is shown in Figure 4.

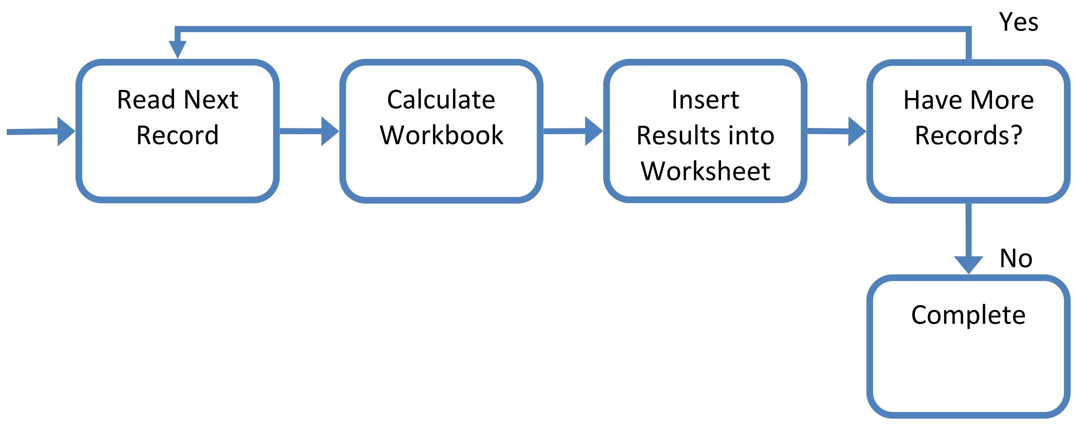


Figure 4. The application flow of the original Excel workbook

This flow occurs in a VBA routine. On the client computer, this makes sense, because the workbook can only calculate one record at a time. In the cluster, however, each server calculates a single record at a time. To support this, the developer isolates the calculation as a separate VBA method, which he or she can call with input data and that returns the result of the calculation, as shown in Figure 5.

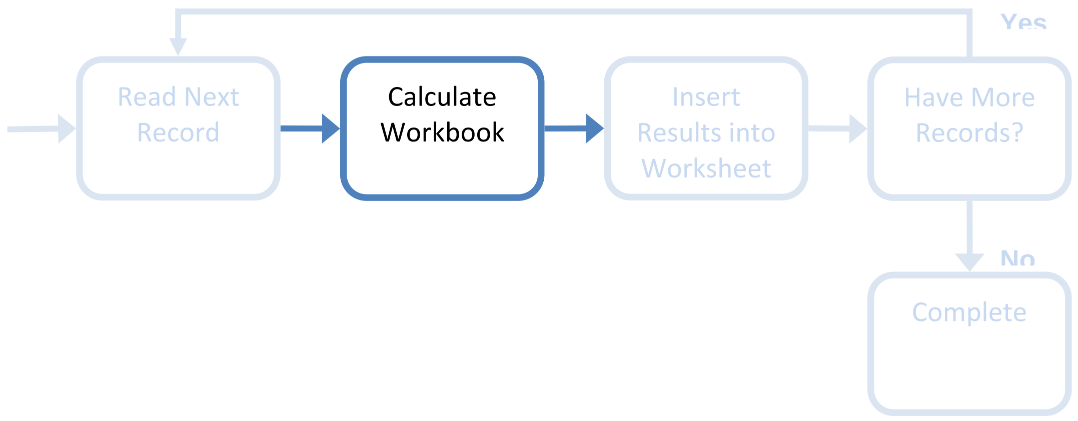


Figure 5. Isolating the calculation as a separate VBA function

This modification changes only the application flow, not the calculation itself. The same VBA code is used to perform the actual calculation, which is important in this solution, because the firm may periodically change the calculation routine, and those changes must be incorporated in the cluster solution. Because this solution supports running the workbook into the cluster and calling VBA functions, all of the business logic can be retained in the VBA code, which is in a new workbook that runs in the cluster.

The rest of the calculation—reading data records and managing results—is handled by the client program. From the standpoint of the user, the application works the same in the cluster as it did on the client computer. Instead of being controlled by VBA, however, the client program—a VSTO add-in—controls the calculation.

All preliminary calculation steps are run on the client computer using Office Excel 2010, as they were in the original application. This process includes loading database tables for tax, financial, and actuarial data into the spreadsheet. The client program does this by calling VBA routines in the workbook.

Next, the client program saves a temporary copy of the workbook, ensuring that the calculation uses the updated data tables and captures any changes the user has made to the workbook. The temporary copy is stored where it will be accessible to the cluster servers.

The client program then reads the input data file. For each record in the file, the client program sends a request to the cluster service. Once all requests are sent, the client waits for calculations to finish. While the application is running, the client program presents visual feedback to the user—a progress bar and the current status of the calculation.

On each server, when the service receives a request, it calls the VBA method in the workbook and run the calculation. When an individual calculation is complete, the service returns the results of the calculation to the client program and waits for the next request.

As results are returned, the client program inserts them into the spreadsheet. Once all records have been calculated, the workbook includes the results for each record as if the calculation had been run on the client computer. The client then takes care of some accounting tasks, saving a copy of the processed workbook and writing a report log.

The application flow in the cluster solution is shown in Figure 6.

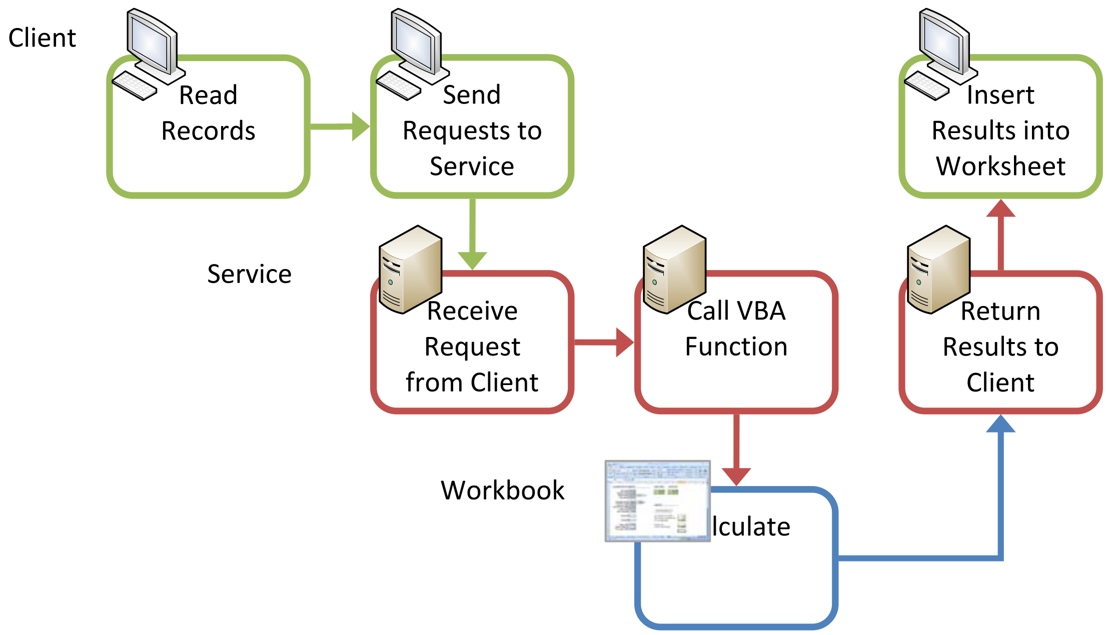


Figure 6. The application flow in the cluster solution

Performance

The 128-core Windows HPC cluster consists of 16 servers, each with 8 processor cores. With a 128-core Windows HPC cluster, the calculation runs in just over two hours (compared with more than one week running on a client computer). Running workbooks in the cluster provides the option of running the service on a per-node or per-core basis. Running per-node, users can run at most one instance of Office Excel 2010 for each cluster server. Running per-core, users can run one instance of Office Excel 2010 for each core on each server.

For this solution, the firm elects to run the workbooks on a per-core basis based on an analysis of the specific workbook. Office Excel 2010 supports multi-threaded workbook calculation, which on the client computer can improve calculation performance by utilizing more than one processor core. However, in the cluster, where multiple instances of Office Excel 2010 are running, it may be more efficient to run the calculation in single-threaded mode but run more than one instance of Office Excel 2010 on each server. By restricting the calculation to a single processor, the individual calculations are slightly slower, but the aggregate calculation is much faster, because each core runs a separate instance of Office Excel 2010.

Running a workbook on a cluster also incurs some overhead. A fixed amount of time is involved in connecting to the job scheduler and starting Office Excel 2010 on each server. There is some additional variable overhead in running the calculation on each compute core, including network latency and additional processing as data records are passed to each process and collect results.

Running Office Excel 2010 UDFs in a Windows HPC Cluster: Derivatives Portfolio Calculation

A large financial services firm uses Excel to price derivative securities. The Excel calculation uses UDFs in an XLL to calculate the prices of individual securities.

Running on a client computer, a market sweep analysis evaluating several thousand individual derivative contracts finishes in approximately 45 minutes. A single portfolio of several hundred contracts finishes in about 5 minutes.

The solution converts the existing UDFs to run in a Windows HPC cluster. Running in a 128-core Windows HPC cluster, the market sweep analysis finishes in less than 1 minute. The individual portfolio is calculated in about 12 seconds.

Overview

A large financial services firm is engaged in pricing derivate securities, both for its own account and for customer accounts. Individual portfolios consist of several hundred individual derivatives contracts. Periodically, the firm performs a market sweep analysis, which consists of calculating prices for several thousand outstanding derivatives contracts.

To support this analysis, the firm has developed several XLLs containing UDFs that perform the price calculations. The specifics of each function depend on the particular contracts, but each function takes a number of parameters and returns a result.

The calculation functions use a binomial or lattice model to arrive at a price for a given security. At the best level of precision, each calculation can take up to 500 milliseconds (1/2 second) to finish.

To calculate a workbook, the user refreshes dynamic market data using Dynamic Data Exchange (DDE). As these values are updated, Excel automatically recalculates the workbook. During the calculation, the user-defined pricing functions are called, and results are returned.

Selecting a Solution

An analysis of the workbooks and functions that the firm uses indicates that the pricing functions consume most of the calculation time. The Excel calculation in a typical workbook is straightforward: When individual prices have been found, the workbook calculates aggregate values and summary statistics.

If the time required to calculate a long-running workbook is consumed by a large number of UDFs, running these functions in the cluster can be the best solution to the problem.

In this case, the UDFs meet the criteria for running in the cluster. Each function is entirely self-contained—that is, each function takes some arguments and returns a specific value but does not look up values or otherwise interact with the workbook. The firm has access to source code for the extension libraries and can rebuild them to support running in the cluster. Therefore, the solution selected in this case is to run the UDFs in a Windows HPC cluster.

Structure of the UDF Solution

The existing XLLs were modified to support calculation in the cluster—a modification that requires only a small change to the code. When the code was updated, the libraries were rebuilt and deployed to users. The functions are also deployed in the cluster using the Windows HPC Cluster Connector provided with the Windows HPC Pack SDK client utilities.

Running on a client computer, the updated functions are identical to the previous versions. However, users of Office Excel 2010 now have the option of running the functions in the Windows HPC cluster.

To run the functions in the cluster, the user enables cluster calculation in the Office Excel 2010 **Settings** dialog box. No changes to the workbooks, user workflow, or process are required. When the workbook is recalculated, Office Excel 2010 sends the UDFs to the cluster for calculation. Running in a 128-core

Windows HPC cluster, the workbook can now calculate up to 128 UDFs simultaneously, resulting in better overall performance.

Performance

Running in a 128-core Windows HPC cluster, the market sweep analysis finishes in less than 1 minute. The individual portfolio is calculated in about 12 seconds.

The UDFs themselves do not run faster in the cluster than they do on the client computer: The performance improvement comes from parallelization. In a 128-core Windows HPC cluster, Excel can calculate 128 functions simultaneously.

Cluster SOA Client Solution: Data Analysis

A financial analyst examines large data sets and develops ratio correlation matrices. Each set of data requires ratio tables and cross-correlation matrices for up to 1,000 observed prices of 100 assets. Using Excel on a client computer, the process can take 30 minutes or more.

This solution uses an Excel add-in built with VSTO as a cluster SOA client to run the calculation in a Windows HPC cluster. Using the cluster solution, the same data set is calculated in less than one minute. The VSTO add-in also automates the process, minimizing the work that the analyst must do and helping to reduce accidental data error.

Overview

A financial analyst examines large data sets and develops ratio correlation matrices. A given set of data might contain 1,000 prices for each of 100 individual assets. The ratio correlation analysis requires first creating price ratio tables representing the movement of individual asset prices over time. For 1,000 prices, the ratio data will be a table of 499,500 rows. The analyst creates ratio tables for each of the 100 assets, and then uses the Excel data analysis tools to generate a correlation matrix. When the correlation matrix has been created, other analysts within the firm use the Excel workbook for portfolio analysis.

The process is time-consuming, because the ratio tables must be created for each data set as it is received—a manual process that can be error-prone. The correlation analysis can be performed automatically with the Excel tools, but this analysis is slow for large data sets. The workbooks become large when the ratio tables have been created, but the tables are not used after the correlation analysis has been performed, and the tables are usually discarded to reduce the size of the workbooks. The analyst would like to be able to perform the calculation more quickly and to automate the process to make it less manually intensive and error-prone.

<u>Selecting a Solution</u>

Because individual ratio tables and correlation pairs are calculated independently, the calculation can be parallelized in the cluster. The actual calculation is relatively straightforward: Both the ratio calculation and the correlation analysis are simple mathematical operations. Complexity arises because the model deals with a large number of individual assets simultaneously and because the generated ratio tables are very large.

For purposes of performance, it makes sense to move the calculation out of Office Excel 2010. The calculation does not use any VBA code or macros, and it is not dependent on calculations in other worksheets or outside resources. The intermediate data is not used, so it should be more efficient to use Office Excel 2010 only for the input and output data.

In this scenario, the calculation could be moved to the cluster in any of several ways. For example, the analyst could create a UDF to perform a single calculation on a pair of columns. However, this approach would still require that the user construct the data tables in Office Excel 2010. The ideal solution will handle the calculation tasks as well as automate the process.

The analyst opts to use Excel as a cluster SOA client. He creates an Excel add-in using VSTO and runs the calculation as a cluster service. VSTO supports automating the process within Excel, reducing manual effort and helping to minimizing the potential for error. The calculation is developed as a C# library that can be run in the cluster. The VSTO add-in connects to the cluster to execute the calculation in parallel.

<u>Structure of the Cluster SOA Client Solution</u>

The solution is built in two parts: a calculation library, which runs in the cluster, and an Office Excel 2010 add-in, which manages the calculation from the client computer. The add-in provides a UI, manages collecting the data from the workbook, and generates the correlation matrix when the calculation is complete.

The calculation library is built as a WCF service designed to run on Windows HPC Server 2008 R2. Using WCF greatly simplifies the design and implementation of the calculation service. The service library developed for this solution includes a single function that performs the ratio and correlation calculation on a pair of price sets. In the cluster, the WCF Broker will handle managing and hosting the service. The Scheduler will handle assigning and managing compute resources.

The service library provides an interface that is used to generate a client library (WCF provides tools for generating this client library automatically). The VSTO add-in uses the client library to connect to the service running in the cluster and call the calculation function.

To use the new cluster calculation service, the analyst clicks a button on the Office Excel 2010 Ribbon. The VSTO add-in first collects data from the workbook, then connects to the cluster service and for each pair of price sets—up to a total of 10,000 pairs—and sends a request to the cluster service by calling the interface function in the client library. The function runs asynchronously, but the add-in displays a progress bar while the calculation is running.

As calculation requests finish, results are returned to the add-in and stored in memory. When all the requests have finished, the add-in creates a new worksheet within the workbook and inserts the calculation results as a correlation matrix.

Performance

Using the cluster SOA client solution with the Windows HPC cluster, the calculation runs much faster—from 30 minutes on the client computer using standard Office Excel 2010 tools to less than 1 minute in the cluster.

The VSTO add-in also automates the process, so that the analyst can calculate the data and generate the correlation matrix as a new worksheet with a single button click. This both speeds up the process and helps to reduce the opportunity for error.

SUMMARY

With the release of Windows HPC Server 2008 R2 and the Office Excel 2010, it is now possible to extend complex and long-running Excel calculations to a Windows HPC cluster. This paper described three methods for doing so: running Office Excel 2010 workbooks in a Windows HPC cluster, running Office Excel 2010 UDFs in a Windows HPC cluster, and using Office Excel 2010 as a cluster SOA client.

Windows HPC Server 2008 R2 now enables running multiple instances of Office Excel 2010 in a Windows HPC cluster, where each instance is running an independent calculation or iteration of the same workbook with a different data set. Using this approach, long-running and iterative workbooks can be calculated in parallel in the cluster, resulting in better overall performance.

Running Office Excel 2010 UDFs in a Windows HPC cluster is a new feature of Office Excel 2010 that provide a platform for running complex or time-consuming UDFs in parallel in a Windows HPC cluster. If a long-running workbook includes multiple UDFs, moving calculation to the cluster can result in significant overall performance improvement.

Windows HPC Server 2008 R2 provides an SDK for building and developing applications that run in the cluster. Using the Windows HPC Pack SDK, users can employ Office Excel 2010 as a cluster SOA client to run complex and time-consuming calculations across a set of servers in a Windows HPC cluster.

Each of these solutions has strengths and weaknesses, and which solution an organization selects depends on that organization's situation or calculation problem. Whatever solution is selected, executing some part of the calculation in a Windows HPC cluster will result in better performance—sometimes dramatically better performance—over calculating the same workbook on a client computer.

Executing Office Excel 2010 calculations in the cluster offers further benefits in flexibility, reliability, and efficient resource sharing—benefits that will accrue to any solution you select.

ADDITIONAL RESOURCES

For more information on the various Microsoft technologies and platforms discussed in this paper, please see the following links:

Windows HPC Server 2008 R2

http://www.microsoft.com/hpc

2010 Microsoft Office system

http://www.microsoft.com/office

Windows HPC TechCenter

http://technet.microsoft.com/en-us/library/ee783547(WS.10).aspx

Visual Studio Tools for Office

http://msdn.microsoft.com/en-us/vsto/default.aspx

Windows Communications Foundation

http://msdn.microsoft.com/en-us/netframework/aa663324.aspx

Understanding Service-Oriented Architecture

http://msdn.microsoft.com/en-us/library/aa480021.aspx