



## Comparing Mainframe and Windows Server Transactions per kWh

White Paper

Published: June 2008

For the latest information, please see [www.microsoft.com/mainframe](http://www.microsoft.com/mainframe)

### Abstract

With the increasing focus on making data centers as green as possible, this paper shows how a well-managed group of PC servers running the Windows Server® operating system provides much better power consumption than z9 mainframes and provides a viable alternative to the newest z10 systems mainframes running the same workload.

Using transaction rates based on the IBM Redbook *A Performance Study of Web Access to CICS* and power consumption data published by IBM, we establish “transactions per kWh (kilowatt-hour)” figures and compare those figures to figures derived using the same transactions that are running on Windows Server–based systems from Fujitsu Computer Systems.

The results weigh heavily in favor of moving off mainframes, which deliver anywhere from 225 to 800 *thousand* transactions per kWh compared to Windows Server–based configurations, which deliver 2.3 to 7.7 *million* transactions per kWh.

Shifting transactions from mainframes to Windows Server–based machines with similar usage management regimes can save over 70 percent in power consumption alone—moving into figures over 90 percent if mainframe Web enablement strategies have already been implemented.

FUJITSU

THE POSSIBILITIES ARE INFINITE

Microsoft®

# Table of Contents

- Executive Summary.....3**
- Objective: Estimate Energy Consumption for Similar Uses .....4**
- Platforms Compared .....5**
- Measuring Performance.....6**
  - The Trader Application..... 6
  - CICS Equivalent for the Windows Server Environment ..... 7
  - Techniques of Web Enablement..... 7
- Comparing Transactions per kWh .....9**
- Windows Server Energy and Carbon Footprint Savings.....11**
- Conclusion: Windows the Greener Option .....14**
- Appendix A. Platform Comparison and Conversion Factors .....15**
  - Details of Platforms Compared ..... 15
  - Conversion Factors..... 15
    - Air Conditioner (Heat Pump) Efficiency Units..... 15
    - British Thermal Units (Btu) to Kilowatt hours (kWh)..... 15
    - Watts (W) to Joules (J) ..... 15
    - Metric Tons (“Tonnes”) to Pounds (lbs) or Short Tons ..... 15
- Appendix B. Calculation Details .....16**
  - Mainframe Trader Results ..... 16
    - Redbook Trader Results ..... 16
    - Extending Trader Results to a Current Mainframe Model ..... 16
  - Windows Server–Based Trader Results ..... 17
  - Calculating Energy Use..... 19
    - Energy Input Plus Cooling ..... 19
    - Heat Pump (Air Conditioning) Efficiency..... 20
    - Mainframe Energy Consumption..... 21
    - Windows Server–Based PRIMERGY Energy Consumption..... 21
  - Combining the Results ..... 21

## Executive Summary

This paper demonstrates that Fujitsu PRIMERGY servers can run between 3 and 24 times as many transactions per kilowatt hour (kWh) as pre-z10 mainframes running the same business transactions. The transaction range depends on whether the mainframe CICS transactions have been Web-enabled, and which technique of Web enablement has been employed. Cost figures are also used to show that moving transactions of this type from the mainframe to Windows Server®-based PRIMERGY servers can save between 19 and 80 carbon credits per year.

These results, summarized in Figure 1, are obtained as follows:

1. Results published by IBM for a sample application called Trader are extrapolated to a recent mainframe using IBM's Large System Performance Reference (LSPR) tables. The published results compare performance for several Web enablement strategies for CICS COBOL applications.
2. The same program, Trader, is run on Fujitsu PRIMERGY servers running the Windows Server 2008 operating system, and using Fujitsu modernize-as-you-migrate products (NetCOBOL, NeoKicks, NeoData) that provide an equivalent runtime environment for the CICS COBOL applications in the Windows Server environment.
3. A key assumption is that the Windows Server environment is managed with the same care as the mainframe environment to ensure that processing power is fully utilized.
4. Published energy consumption figures for both platforms are combined with the performance results to derive transactions-per-kWh figures.
5. Total transaction rates and overall power consumption are also combined and compared to derive potential carbon credit savings.

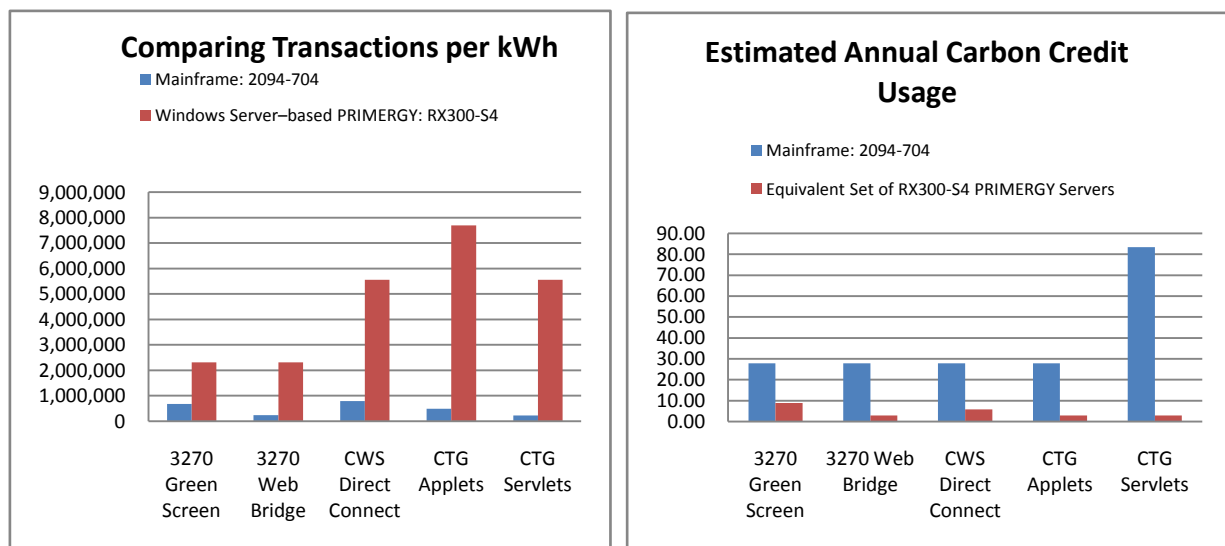


Figure 1. Comparisons of transactions per kWh and carbon credit usage, using various Web enablement techniques.

## Objective: Estimate Energy Consumption for Similar Uses

For several years Fujitsu and Microsoft have been helping companies save considerable amounts of money and gain application agility by migrating mainframe applications to the Microsoft® .NET Framework. With organizations becoming increasingly conscious of their energy consumption, there is a clear need to determine how this move affects an organization's energy footprint.

For this paper we focus on the move that both Fujitsu and Microsoft recommend: Take legacy applications from well-managed mainframes and use our tools to move them to the .NET Framework running on equally well-managed servers. A key part of this message is that the same standards that are applied to managing the mainframe environment should be applied to managing the Windows Server environment. Although it should be obvious, it often needs to be stated that if you want the same level of security or the same utilization of resources, you need to manage them in the same, or similar, ways.

For an energy consumption comparison, this means that we need to look at Windows Server-based machines that are sized to match the workload in the same way that mainframes are sized—namely, ensuring that the available processing power is fully utilized. This is also a reasonable cost management strategy because Windows Server-based systems can often be quickly and easily expanded through a scaled-out architecture when usage increases, so it is not necessary to purchase excess capacity far in advance of actual need.

Our approach in this paper has therefore been the following:

- Obtain performance figures for a typical mainframe legacy online (CICS) application, along with various Web enablement forms of that application.
- Extrapolate these figures to a recent (z9) mainframe for which energy consumption figures have been published<sup>1</sup>, assuming the mainframe is 100-percent utilized.
- Obtain performance figures for the same application, and its Web enablement equivalents, in the Windows Server environment. We drove the Windows Server-based system to between 84-percent utilization and 97-percent utilization, which was sufficient to roughly parallel the mainframe performance (and gave an advantage to the mainframe in the energy comparison, so that our comparison would be on the conservative side).
- Use energy consumption and heat dissipation figures from both platforms to calculate transactions-per-kWh figures for each application form, and then compare the two platforms.

---

<sup>1</sup> Stahl, Elizabeth, Michael Buechele, and John P. Rankin, "A Power Benchmarking Study on the IBM System z9: Applying Energy Efficiency Metrics to Performance," IBM, October 4, 2007.

[www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101110](http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101110)

## Platforms Compared

Because every computer has a different processing power and energy consumption rate, this paper does not attempt to give an exhaustive analysis by comparing many mainframes to many Windows Server–based machines. Rather it aims to give a snapshot of how the two platforms compare by taking figures using a fairly recent energy-efficient mainframe and a current energy-efficient Windows Server–based machine. The paper fully documents the calculations that were performed; we will also provide the source code upon request<sup>2</sup>. Readers can perform the same tests and calculations on platforms of their choice.

We chose the mainframe model that IBM used in energy consumption tests for the white paper “A Power Benchmarking Study on the IBM System z9: Applying Energy Efficiency Metrics to Performance” (see footnote 1, page 4). Because one of the goals of the IBM paper was to help make the case for the mainframe as an energy-efficient platform, we assume that the model used in that paper, an IBM System z9 EC 2094-704 processor, was one of the more energy-efficient mainframes available at that time, and thus provides a fair target for our comparison. Most mainframes that are being used by organizations considering the move to the .NET Framework will likely be older and less energy-efficient than the 2094-704.

The server model chosen for use with Windows Server was the Fujitsu PRIMERGY RX300-S4 rack server with two Quad-Core Intel Xeon X5460 processors. This is one of the PRIMERGY models currently available at Fujitsu’s TRIOLE Integration Center, a center set up to help partners test cross-platform porting or interoperability with Fujitsu hardware. Using a current PRIMERGY server is a reasonable choice for this comparison, because anyone who is now deciding to move from the mainframe to a Windows Server environment will be purchasing their servers in the future.

For the software on the PRIMERGY server, we used Windows Server 2008 and appropriate components of the Fujitsu “modernize-as-you-migrate” suite of products. This product suite includes NetCOBOL for .NET, NeoKicks, and NeoData, which were used in these tests, and NeoBatch and NeoSort, which were not used in these tests. The suite provided a good base for comparing performance with the mainframe. Much of the mainframe application code remained unchanged, so we were able to compare like with like. By opening the application to new technologies, such as converting BMS maps to Microsoft ASP.NET pages and giving access to the .NET Framework, we could compare performance when using the various Web enablement strategies offered by CICS. We could also take advantage of software that offers a serious modernization route for legacy mainframe applications.

---

<sup>2</sup> To receive a copy of the Trader source code that was used in the tests performed for this paper, send a request to [netcobol@netcobol.com](mailto:netcobol@netcobol.com).

## Measuring Performance

In comparing power consumption per transaction on the mainframe and on Windows Server–based machines, one challenge is to find an application that has been, or can be, benchmarked on both platforms. Neither IBM nor Fujitsu is currently publishing figures based on the commonly recognized benchmark TPC-C<sup>3</sup> for the platforms we compare. There is also no other benchmark that both companies have used when publishing figures for the selected platforms.

Fortunately IBM has created a CICS COBOL application called Trader for the purpose of reviewing the performance of various Web-enabling options in the CICS system. IBM published the results of this comparison in the Redbook *A Performance Study of Web Access to CICS*<sup>4</sup>. The Trader source code was also made available, so we have been able to run tests on the PRIMERGY server that are identical to those reported in the Redbook. By using IBM’s Large System Performance Reference (LSPR) tables<sup>5</sup> we can extrapolate the results published in the Redbook to the mainframe selected for this paper, and obtain a reasonable comparison of mainframe and PRIMERGY transaction processing. This constitutes a like-to-like, or apples-to-apples, comparison.

The Trader application, Web enablement options, and the results from driving transaction workloads in older and recent mainframe platforms and on Windows Server platforms are described in the following sections.

### The Trader Application

The Trader application is a simple version of a stock-trading application. It allows authenticated users to buy and sell shares and to obtain real-time quotes on their current holdings. The Redbook *A Performance Study of Web Access to CICS* contains a full description of the Trader structure and flow. The Redbook defines a sequence of operations that represent a typical Trader “transaction” that was used for testing. The transaction involves:

- Nine pairs of BMS maps received and sent.
- A BMS RECEIVE with a 3270 SEND.
- Numerous disk I/O operations to support the transaction requests.
- A limited amount of business logic to support the transaction requests.

---

<sup>3</sup> TPC-C is one of the standard benchmarks of the Transaction Processing Performance Council: [www.tpc.org/information/benchmarks.asp](http://www.tpc.org/information/benchmarks.asp).

<sup>4</sup> Wakelin, Phil, Graham Rawson, Per Fremstad, Dave Scott, and Andy Abbey, *A Performance Study of Web Access to CICS*, IBM Redbooks, February, 2000.

<sup>5</sup> LSPR workload tables are comparisons of performance of mainframe computers; the tables are published by IBM at [www-03.ibm.com/servers/eserver/zseries/lspr/index.html](http://www-03.ibm.com/servers/eserver/zseries/lspr/index.html). The LSPR workloads represent various usage profiles, to help users compare performance using a profile that is closest to their typical application mix. For example, some users may run only batch applications, and others may run online applications almost exclusively.

By combining several screen interactions, variable data access, and business logic, the defined Trader transaction provides a reasonable emulation of typical real-world online business transactions.

### **CICS Equivalent for the Windows Server Environment**

One of the reasons for choosing Trader is that it is compatible with the NeoKicks product, which takes CICS COBOL applications to the .NET Framework. Although by design NeoKicks converts CICS BMS maps to Microsoft ASP.NET Web pages, it nevertheless provides a good platform for comparing the Windows Server–based environment to the mainframe environment for several reasons:

- NeoKicks leaves the COBOL code as it is. (Like CICS on the mainframe, it converts EXEC CICS statements to invocations of NeoKicks system routines, leaving the rest of the code unchanged.)
- From the application user’s perspective, the initially converted interface looks and behaves just like the 3270 green screen (even though it is actually a Web page).
- For the migrated CICS application and interface, NeoKicks supports further enhancements that can parallel the Web enablement strategies recommended by IBM.
- NeoKicks has already been used by many companies to bring millions of lines of CICS COBOL code to the .NET Framework, demonstrating that it is a viable option for legacy mainframe applications.

Keeping with the modernize-as-you-migrate philosophy, we also used NeoData to migrate VSAM data to Microsoft SQL Server® database software. NeoData integrates with NeoKicks, so that although READ and WRITE statements remain unchanged, they access the data in SQL Server instead of in VSAM files. Using NeoData also lets users of Trader create ad-hoc reports using SQL queries in SQL Server, significantly modernizing the application (though generating ad-hoc reports was not part of the test). Thus the business function of the Trader application remains the same, with little or no change to the COBOL code—which means that we are comparing transactions with exactly the same function, nearly identical application code, and a realistic means of moving from one platform to the other.

### **Techniques of Web Enablement**

The Redbook *A Performance Study of Web Access to CICS* analyzes the performance of a number of different ways of enabling the Trader application for the Web. These techniques range from using the CICS 3270 Web Bridge, which requires almost no change to the Trader code, to rewriting major parts of Trader in Java in order to use the CICS Transaction Gateway. The processing power that is necessary to support these different Web enablement techniques can vary from requiring a little less than the CICS green screen version to requiring three times as much.

Because these Web enablement techniques have been available for several years, and because organizations have experienced business pressures to enable their CICS applications for Web use, in this paper we assume that a number of readers have already modified applications to take advantage of one or more of these techniques. We have therefore identified Windows Server equivalents for a number of the techniques, so that you can see how performance and energy consumption compare for the various models (see Table 1).

**Table 1. Correspondence Between Web Enablement Techniques for Mainframe and Windows Server–Based Environments**

Web Enablement Technique	Windows Server–Based Equivalent	Comments
<b>3270 Green Screen (no Web enablement)</b>	NeoKicks green screen look-alike	This is the most basic NeoKicks option. It provides the same look and feel, even though the NeoKicks application is Web-enabled.
<b>3270 Web Bridge</b>	NeoKicks green screen look-alike	This option can Web-enable the Trader application while preserving much of the look and feel of the original Trader 3270 application.
<b>CICS Web Support (CWS) Direct Connect</b>	NeoKicks converted application, with new user interface written in C#	We wrote some new user interface pages in C# calling NeoKicks transactions that mimic the functionality of CWS direct connect that is described in the Redbook.
<b>CICS Transaction Gateway (CTG) Applet</b>	NeoKicks converted application, with separate Windows® program calling Web services	We wrote a separate Windows program in C# that calls a simple Web service for each call to NeoKicks, to run the business logic, which emulates the CTG applet model.
<b>CTG Servlet (rewrite Trader to use a Java servlet using the CTG external call interface [CTG ECI] to invoke the Trader business logic module)</b>	NeoKicks converted application, with new user interface written in C#	The code that we created for the CWS direct connect comparison was the closest match, in .NET, to the CTG servlet model.



## Comparing Transactions per kWh

The results of measuring transaction rates for the different Web enablement techniques are summarized in Table 2 alongside the energy consumption rates for the two platforms. From these figures it is a simple calculation to establish the number of transactions that would be performed in the time it takes the computer to consume one kilowatt hour:

$$\text{Transactions per kWh} = (\text{Transactions per second} / \text{Energy used per second}) \times \text{Energy in a kWh}$$

The final column of Table 2 shows the resulting figures. Figure 2 illustrates the results graphically.

You can see that the actual transaction-per-second rates vary significantly across the Web enablement techniques, but the difference in energy consumption between the two platforms results in a huge difference in the amount of business work performed when one kilowatt hour of energy is expended. Whereas the mainframe figures are in the hundreds of thousands, the PRIMERGY figures are in the millions.

Appendix B explains in detail how the transactions per second and energy consumption rates were obtained, including the extrapolations from the older to the newer mainframe model.

**Table 2. Energy Consumption per Transaction for 2094-704 Mainframe and Windows Server–Based PRIMERGY RX300-S4**

Web Enablement Technique	Computer	Transactions per Second	Rate of Energy Consumption (Watts)	Transactions per Kilowatt Hour (kWh)
<b>3270 Green Screen (no Web enablement)</b>	Mainframe	984	5,221	678,455
	PRIMERGY	354	551	2,314,939
<b>3270 Web Bridge</b>	Mainframe	341	5,221	234,810
	PRIMERGY	354	551	2,314,939
<b>CICS Web Support (CWS) Direct Connect</b>	Mainframe	1155	5,221	796,524
	PRIMERGY	850	551	5,555,593
<b>CICS Transaction Gateway (CTG) Applet</b>	Mainframe	717	5,221	494,616
	PRIMERGY	1177	551	7,696,852
<b>CICS Transaction Gateway (CTG) Servlet</b>	Mainframe	327	5,221	225,487
	PRIMERGY	850	551	5,555,593

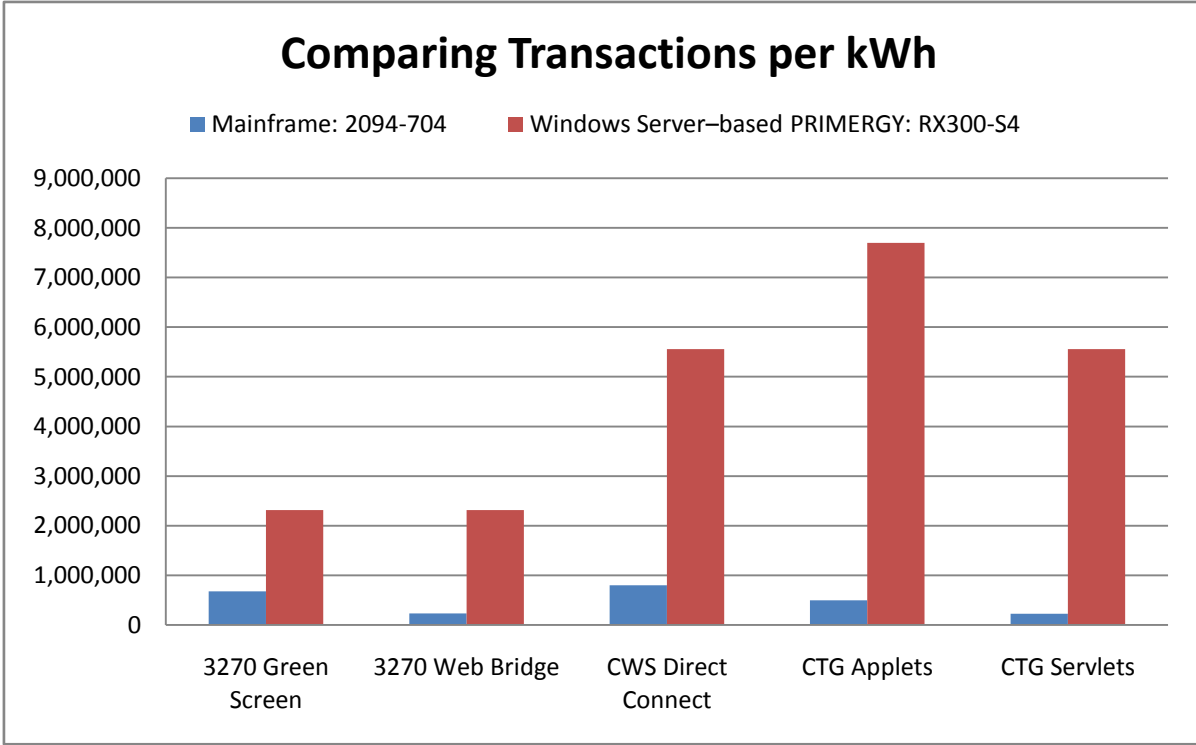


Figure 2. Transaction rates per kilowatt hour for 2094-704 mainframe and Windows Server-based PRIMERGY RX300-S4.

## Windows Server Energy and Carbon Footprint Savings

A key concern today is how big our carbon footprint is—the quantity of greenhouse gases that we cause to be emitted.

Looking at the carbon footprint requires a different approach from the transaction-per-kWh analysis performed thus far in this paper. We must assume that the computers will be purchased, switched on, and kept running regardless of the actual workload supported at a particular time. As confirmed in the IBM paper “A Power Benchmarking Study on the IBM System z9,” the power consumed by computers does not vary greatly between a low-usage state and a high-usage state. Therefore to compare carbon footprints, we simply need to ensure that we are providing systems with roughly the same computing power and compare their power consumption figures converted to carbon footprint units. (After these tests were performed, Fujitsu introduced PRIMERGY servers that provide users with the ability to select between maximum performance and minimum power consumption modes, so it is possible to lower the PRIMERGY server power consumption when workloads are low or not time-critical. We do not take account of this energy-saving ability in the following calculations.)

A common measure used for carbon footprints is the carbon credit, which is equivalent to 1 metric ton of CO<sub>2</sub>. (A metric ton or “tonne” is 1,000 kilograms [kg] or 2,205 pounds [lbs].)

The Greenhouse Gas Protocol Initiative (GHG)<sup>6</sup> developed the protocol that is now used by most companies in estimating their carbon footprints. Statistics published by the GHG show that carbon dioxide emissions in the United States can vary from 0.30 to 0.91 kg per kilowatt hour (kWh), depending on the region (as defined by the non-profit North American Electric Reliability Corporation, or NERC). A simple average across these regions is:

$$0.63 \text{ kg per kWh} = 0.00063 \text{ carbon credits per kWh}$$

Carbon emission statistics published by the U.S. Government<sup>7</sup> suggest that use of 1 kWh of electricity in the United States causes 1.341 pounds of carbon dioxide to be emitted into the atmosphere. When we convert this figure to carbon credits, 1 kWh is equivalent to 0.00061 carbon credits.

The comparison of carbon footprint units is shown in Table 3. The figures in this table assume linear scaling of the Trader transactions that are running under NeoKicks on the PRIMERGY server. This assumption is supported by our tests of NeoKicks on ASP.NET, and the relatively simple form of the Trader transactions. Because the U.S. Government figure for carbon emission is published as a nationwide average, we use it in Table 3; it is likely a more accurate average than the simple average taken from the GHG figures.

---

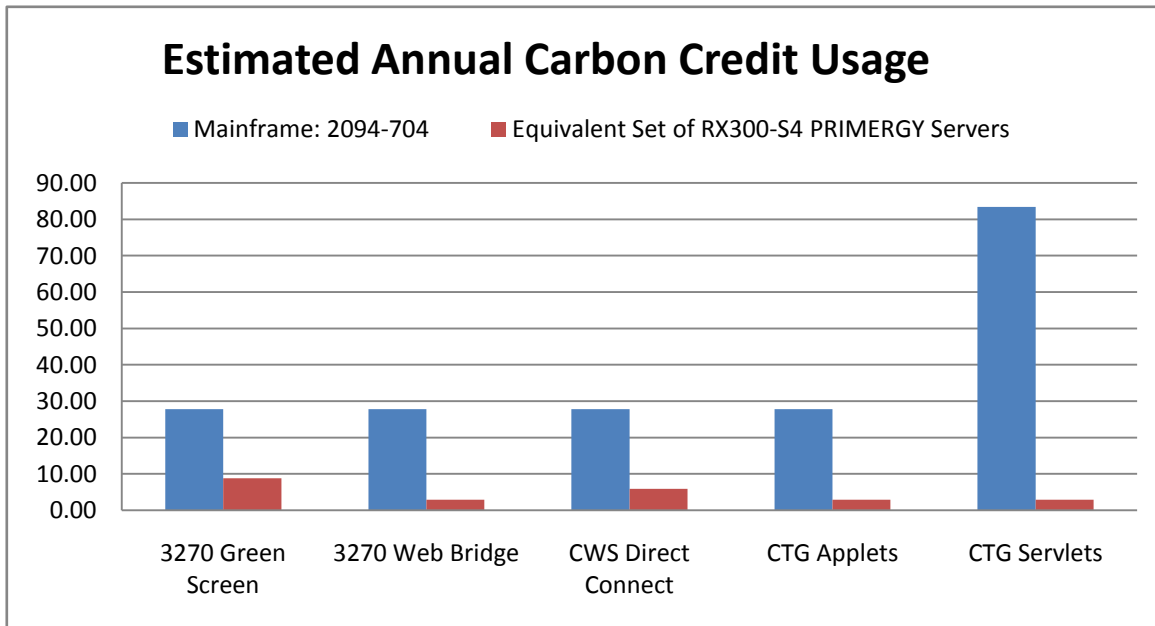
<sup>6</sup> A partnership between the World Resources Institute and the World Business Council for Sustainable Development, which works with businesses, governments, and environmental groups around the world to build credible and effective programs for tackling climate change. Data obtained from their spreadsheet on “Indirect CO<sub>2</sub> emissions from purchased electricity” is downloadable (after registration) from [www.ghgprotocol.org](http://www.ghgprotocol.org).

<sup>7</sup> Updated State-level Greenhouse Gas Emission Coefficients for Electricity Generation 1998-2000, Energy Information Administration, Office of Integrated Analysis and Forecasting, U.S. Department of Energy, April 2002.

**Table 3. Calculating Carbon Footprint Savings of Moving to a Windows Server Environment**

Web Enablement Technique	Transaction Rates		Number of Mainframes to Number of Windows Server-based machines		Carbon Credits per Year		Savings (carbon credits = tonnes of CO <sub>2</sub> )
	2094-704	RX300-S4	2094-704	RX300-S4	2094-704	RX300-S4	
3270 Green Screen (no Web enablement)	984	354	1	3	27.81	8.80	19.01
3270 Web Bridge	341	354	1	1	27.81	2.93	24.88
CICS Web Support (CWS) Direct Connect	1155	850	1	2	27.81	5.87	21.95
CICS Transaction Gateway (CTG) Applet	717	1177	1	1	27.81	2.93	24.88
CICS Transaction Gateway (CTG) Servlet	327	850	3	1	83.44	2.93	80.51

Look at the first row in Table 3. This represents the mainframe’s optimal situation—requiring that we use three RX300-S4 servers to replace the 2094-704, in order to compare a 3270 green screen application on the mainframe to a Web-enabled equivalent in the Windows Server environment. Even in this case, we can still save 19 tonnes of carbon emissions a year by switching to a Windows Server environment. With the popular Web enablement strategies of using Java applets or servlets, the savings can increase to as much as 80 tonnes per year. Figure 3 shows the results for each platform in a graphical format.



**Figure 3. Estimated annual carbon credit usage (tonnes of CO<sub>2</sub>) for 2094-704 mainframe and Windows Server-based PRIMERGY RX300-S4 server.**

The actual savings can vary radically, depending on the NERC electricity region from which you receive your power (see Table 4). At the low end, you can save 8 tonnes of emissions per year. At the high end, the figure climbs to 120 tonnes.

**Table 4. Carbon Credits with Minimum, Maximum, and Average Emissions (Using GHG Figures for NERC Regions)**

Web Enablement Type	Carbon Credits per Year					
	Minimum		Maximum		Average	
	2094-704	RX300-S4	2094-704	RX300-S4	2094-704	RX300-S4
3270 Green Screen (no Web enablement)	13.92	5.87	41.72	17.60	28.98	12.23
3270 Web Bridge	13.92	2.94	41.72	8.80	28.98	6.11
CICS Web Support (CWS) Direct Connect	13.92	2.94	41.72	8.80	28.98	6.11
CICS Transaction Gateway (CTG) Applet	13.92	1.47	41.72	4.40	28.98	3.06
CICS Transaction Gateway (CTG) Servlet	41.77	1.47	125.17	4.40	86.93	3.06

## **Conclusion: Windows the Greener Option**

In this paper we have presented a snapshot view, comparing a recent mainframe model operating efficiently at maximum capacity with a modern PRIMERGY server operating efficiently at around 90-percent capacity. The Windows Server platform can in fact be between 3 and 24 times more energy-efficient than the mainframe.

The important factor in achieving such gains is that the same care must be taken in managing the Windows Server environment as is taken in managing the mainframe environment. This is something that can be planned and managed as a requirement of a modernize-as-you-migrate project.

These results may appear surprising to some, yet they are totally in accord with the experience of Fujitsu and Microsoft. Both companies have seen every customer who has migrated from the mainframe impressed by the performance delivered by a Windows Server–based platform. Both companies know that today’s Windows Server environments are at the leading edge of power efficiency.

## Appendix A. Platform Comparison and Conversion Factors

### Details of Platforms Compared

	Mainframe environment: IBM 2094-704	Windows Server environment: PRIMERGY RX300-S3
<b>Hardware</b>	<ul style="list-style-type: none"><li>• System z9, model S08 with a single cargo I/O cage</li><li>• 2094-704 indicates 4 cores being used</li></ul>	<ul style="list-style-type: none"><li>• CPU: Two Xeon X5460 3.16 GHz</li><li>• Memory: Eight 2-GB – FBD 667 PC2-5300F ECC</li><li>• HDD: 2 * 73-GB SAS (hot-plug)</li><li>• 1 LPe 11002 Emulex Fibre Channel 4 Gb/s Dual Channel HBA</li><li>• Integrated remote management controller</li></ul>
<b>Software</b>	<ul style="list-style-type: none"><li>• Operating system: z/OS 1.8</li><li>• NetCOBOL for .NET v3.1</li></ul>	<ul style="list-style-type: none"><li>• Operating system: Windows Server 2008 Standard Edition (32-bit)</li><li>• NeoKicks v2.0.13.0</li><li>• NeoData v1.0.10.0</li><li>• SQL Server 2008 February CTP (32-bit)</li></ul>

### Conversion Factors

#### Air Conditioner (Heat Pump) Efficiency Units

1 SEER = 1.111 EER

1 COP = 3.792 SEER = 3.413 EER

#### British Thermal Units (Btu) to Kilowatt hours (kWh)

1 Btu = 0.000293 kWh

#### Watts (W) to Joules (J)

1 W = 1 Js<sup>-1</sup>

#### Metric Tons (“Tonnes”) to Pounds (lbs) or Short Tons

1 tonne = 1000 kg = 2205 lbs = 1.103 short tons

## Appendix B. Calculation Details

### Mainframe Trader Results

#### Redbook Trader Results

The mainframe used in the IBM Redbook comparison was an IBM 9672-R55 that contained five processors.<sup>8</sup> Web client simulations were provided by connecting IBM AIX operating systems to the mainframe (see *A Performance Study of Web Access to CICS* for details).

The mainframe Trader test results are shown in Table 5.

**Table 5. Trader Test Results on the 9672-R55 Mainframe, by Web Enablement Technique (in Milliseconds [ms])**

Web Enablement Technique	CPU ms for 10 Trader Transactions	Trader Transactions per Second
<b>3270 Green Screen (no Web enablement)</b>	452	111
<b>3270 Web Bridge</b>	1306	38
<b>CICS Web Support (CWS) Direct Connect</b>	385	130
<b>CICS Transaction Gateway (CTG) Applet</b>	620	81
<b>CICS Transaction Gateway (CTG) Servlet</b>	1360	37

#### Extending Trader Results to a Current Mainframe Model

The mainframe we selected for our power consumption review is the IBM 2094-704, a considerably newer and more powerful model than the 9672-R55 used for the Redbook tests.

We used the LSPR workload tables to obtain a conversion factor that compares the power of the more modern mainframe to the power of the older mainframe. In the various tables, the two workload figures that are closest to the Trader type of application are CICS/DB2 (CICS online transactions using the IBM DB2 database) and OLTP-W (a Web-enabled online workload). Using the CICS/DB2 and OLTP-W workload figures, we obtain the figures that are shown in Table 6 below. The differences between the various workload figures are small, so we could also have used the more general TI-Mix workload (a transaction-intensive workload created by a weighted combination of other workloads) and obtained much the same result.

---

<sup>8</sup> Although comments in the Redbook imply that CICS might reach certain limits before the mainframe is fully loaded, and the claims of linear scaling are demonstrated using only narrow workload ranges, we have assumed that linear scaling with CICS scales is perfect across all five processors. Therefore we assume that the fully loaded 9672-R55 can support five times the number of transactions that a fully loaded single processor supports. The Trader transactions-per-second figures should therefore represent an absolute upper limit on the maximum attainable by the 9672-R55.



**Table 6. LSPR Table Ratios for Test (Redbook) and Target (Current) Mainframes**

Mainframe Model	Ratio	Relative to Model
9672-R55 (Redbook)	1.05	2064-1C1
2094-704 (current)	4.95	2084-301
Because our two computers (left column) are compared to two different base models (right column), we need a third figure that compares those two base models to each other:		
2064-1C1	0.53	2084-301

From these figures we calculate the following:

2094-704 is ( 4.95 / 0.53 = 9.34 ) times as powerful as the 2064-1C1.

2094-704 is ( 9.34 / 1.05 = 8.89 ) times as powerful as the 9672-R55.

We then use the 8.89 conversion factor to extrapolate the results from Table 5 to the 2094-704, as shown in Table 7.

**Table 7. Trader Test Results, Extrapolated to the 2094-704 Mainframe**

Web Enablement Technique	CPU ms for 10 Trader Transactions	Trader Transactions per Second	
		9672-R55	2094-704 (extrapolated)
3270 Green Screen (no Web enablement)	452	111	984
3270 Web Bridge	1306	38	341
CICS Web Support (CWS) Direct Connect	385	130	1155
CICS Transaction Gateway (CTG) Applet	620	81	717
CICS Transaction Gateway (CTG) Servlet	1360	37	327

### Windows Server-Based Trader Results

For the PRIMERGY server tests, the Microsoft Visual Studio® 2008 Web test recorder was used to drive the transactions and record the results. These tests ran on a CELSIUS R650 workstation with two Quad-Core XEON E5420 2.5 gigahertz (GHz) processors. This workstation was connected to the PRIMERGY RX300-S4 server that ran NeoKicks via a gigabit network link. This setup allowed us to drive the RX300-S4 server to sustained workloads of between 84 and 97 percent of its capacity.

The results that we obtained by running the equivalent Trader versions on the PRIMERGY RX300-S4 server, as specified in

Table 1, are shown in Table 8 in transactions per second (TpS).

**Table 8. Trader Test Results on Windows Server–Based PRIMERGY RX300-S4 Server**

Windows/NeoKicks Web Enablement Type Equivalent to:	Recorded Result	On PRIMERGY RX300-S4		
		Rate	Trader TpS	CPU Utilization
3270 Green Screen (no Web enablement)	3541	Requests per second	354	90%
3270 Web Bridge	3541	Requests per second	354	90%
CICS Web Support (CWS) Direct Connect	4249	Requests per second	850*	84%
CICS Transaction Gateway (CTG) Applet	3532	ECI calls per second	1177*	97%
CICS Transaction Gateway (CTG) Servlet	4249	Requests per second	850*	84%

\*Whereas the original Trader application makes 10 requests per business transaction, the recoded Direct Connect and CTG servlet versions make 5 requests per business transaction, and the CTG applet makes only three ECI calls per business transaction—hence the different conversion factors seen in Table 8 between requests per second and Trader transactions per second.

## Calculating Energy Use

To measure energy consumption in computers, we must take into account the energy necessary to run the computers, and the energy that goes into cooling the computers. From calculations base on these two factors, we can derive comparative figures for the mainframe and Windows Server environments.

### Energy Input Plus Cooling

The energy consumed in running a computer consists of two factors:

- **Input power:** The electrical energy required to keep the computer running—that is, the electrical input to the computer.
- **Cooling power:** The energy required to run a cooling system to maintain the computer at a reasonable operating temperature.

We obtained the input power from “active power” figures published by IBM and from the iRMC log on the PRIMERGY server, which records actual power consumed. (There are different ways of calculating the power of alternating current; “active power” is the one that matches the usual understanding of power consumed.)

The energy required for cooling depends on the outside air temperature and humidity. It’s going to be cheaper to keep something cool in Antarctica than it will be on the equator. However, rather than attempt to provide a range of temperature-dependent figures, we calculate the energy required for

cooling by taking the published heat generated by the computers and using an average heat pump efficiency figure to derive the energy required to keep the computer at a steady temperature.

### **Heat Pump (Air Conditioning) Efficiency**

Air conditioning units are often rated using EER (Energy Efficiency Ratio) or SEER (Seasonal Energy Efficiency Ratio), ratings that are defined by the U.S. Energy Information Administration. A SEER of 13 is the minimum rating for U.S. residential air conditioning systems that were manufactured in 2005. Larger cooling systems are more often rated using EER and do not have a regulated minimum, but rebates available from utility companies suggest that a rating of EER 10 or greater is a reasonable value today.<sup>9</sup>

EER and SEER are numbers derived from British Thermal Units (Btu). We have chosen to use the more intuitive thermodynamic quantity COP (Coefficient of Performance), which is not dependent on a particular unit of measurement. The COP of a heat pump is determined by dividing the energy output of the heat pump in watts (W) by the electrical energy in watts that is needed to run the heat pump. Using the COP, we take the heat that is generated by a computer and divide by the COP for a reasonably efficient air conditioning system to determine the electrical energy that is required to move that amount of heat out of the building.

Here, approximate formulas for converting between these different quantities are derived from Pacific Gas and Electric figures<sup>10</sup>, based on lower elevation portions of California (an area for which formulas were provided, and which probably represents a reasonable average for the United States):

$$\text{SEER} = \text{EER} \div 0.9$$

$$\text{SEER} = \text{COP} \times 3.792$$

$$\text{EER} = \text{COP} \times 3.413$$

Or, rearranging the latter two equations:

$$\text{COP} = \text{EER} / 3.413$$

$$\text{COP} = \text{SEER} / 3.792$$

Thus the COP for reasonably efficient air conditioning systems would be:

$$\text{EER of 10: Efficient COP} = 10 / 3.413 = 2.93$$

Or

$$\text{SEER of 13: Efficient COP} = 13 / 3.792 = 3.43$$

---

<sup>9</sup> Two examples of utility company rebates for efficient cooling systems are the Tampa Electric Commercial Cooling Program ([www.tampaelectric.com/business/saveenergy/coolingrebate](http://www.tampaelectric.com/business/saveenergy/coolingrebate)) and the Platte River Power Authority Cooling Rebate Program ([www.prpa.org/productservices/crppoverview.htm](http://www.prpa.org/productservices/crppoverview.htm)).

<sup>10</sup> [www.pge.com/includes/docs/pdfs/mybusiness/energysavingsrebates/incentivesbyindustry/education/spc/2004/manualsforms/spc\\_cooling\\_units.pdf](http://www.pge.com/includes/docs/pdfs/mybusiness/energysavingsrebates/incentivesbyindustry/education/spc/2004/manualsforms/spc_cooling_units.pdf)

For simplicity we will use a COP of 3: the energy that is required for a computer to remove a given heat output is one-third (1 / 3) of the heat output.

### **Mainframe Energy Consumption**

From figures presented in the IBM paper “A Power Benchmarking Study on the IBM System z9” we can assume that a busy 2094-704 does the following:

- Uses 3,916 W.
- Dissipates 13,361 Btu per hour, or 3.92 kilowatt hours per hour (kWh/hr).

These figures convert to heat dissipating at a rate of 3915 W. Allowing for rounding errors, the heat dissipated is essentially the same as the input energy. This is not really surprising, because energy must be conserved, and the only external energy output by the computer is heat (plus a small amount of sound).

Thus for the mainframe the total energy required is equivalent to:

$$\begin{aligned} & \text{Total energy input} + \text{Total heat dissipated} / 3 \\ & = 3,916 + 3,915/3 \\ & = 5,221 \text{ W} \end{aligned}$$

### **Windows Server-Based PRIMERGY Energy Consumption**

The Fujitsu PRIMERGY iRMC utility recorded that the server consumed energy at an average rate of 413W during the tests. The Fujitsu Power Configuration Tool<sup>11</sup> shows us that a busy PRIMERGY RX300-S4 rack server dissipates heat at a rate of 1203 kJ/hr = 0.413 kWh/h = 334 W

Thus for the PRIMERGY server the total energy required is equivalent to:

$$\begin{aligned} & \text{Total energy input} + \text{Total heat dissipated} / 3 \\ & = 413 + 413/3 \\ & = 551 \text{ W} \end{aligned}$$

### **Combining the Results**

Table 2 on page 8 shows the energy figures and transaction rates being combined to give the transactions per kilowatt hour figures. The calculation is:

$$\text{Transactions per kWh} = (\text{Transactions per second} / \text{Energy used per second}) \times \text{Energy in a kWh}$$

---

<sup>11</sup> A spreadsheet created from power consumption tests conducted by Fujitsu-Siemens; designed to provide power consumption and heat dissipation rates for specifiable PRIMERGY server configurations to PRIMERGY users and prospective users.

For example, for the PRIMERGY figures for the 3270 green screen equivalent, Table 8 shows that we had 354 transactions per second, the above section gives the energy rate to be 551 W (1W = 1 Joule/sec), and a kilowatt hour contains 3,600,000 Joules, so:

Transactions per kWh = (354 TpS /551 Joule/sec) x 3,600,000 Joules/hour

Transactions per kWh = 2,314,939

## Appendix B. Answering Other Mainframe Migration Questions

If you are running your applications on a mainframe today, there are probably many questions that come to mind about whether migrating to a Windows platform can really work for you. The following papers and resources will help you answer those questions.

### White Papers on Mainframe Modernization

For white papers on a wide range of mainframe modernization topics such as managing security, disaster recovery, performance, handling online or batch transactions, can be found at:

<http://www.microsoft.com/windowsserver/mainframe/papers.mspx>

### Mainframe Modernization Products and Services

For information on companies that offer products and services in the mainframe modernization space, check out the Mainframe Migration Alliance (MMA) at:

<http://mainframemigration.org/>

### Answers to Your Unique Mainframe Modernization Questions

If you have specific questions not answered by the above resources, you can request help from Microsoft or Fujitsu by contacting them at:

Microsoft: [mframe@microsoft.com](mailto:mframe@microsoft.com)

Fujitsu: Send an email to [cobol@netcobol.com](mailto:cobol@netcobol.com)

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This research note is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2008 Microsoft Corporation. All rights reserved.

Microsoft, SQL Server, Visual Studio, Windows, and Windows Server are trademarks of the Microsoft group of companies. All other trademarks and product names are property of their respective owners.

Fujitsu, the Fujitsu logo, TRIOLE and NetCOBOL, are trademarks or registered trademarks of Fujitsu Limited in the United States and other countries.

NeoKicks, NeoBatch, NeoSort and NeoData are trademarks or registered trademarks of Fujitsu Computers Systems Corporation in the United States and other countries.

PRIMERGY is a trademark or registered trademark of Fujitsu Siemens Computers GmbH in the United States and other countries.

Intel Xeon is a trademark of Intel Corporation in the United States and other countries.