

Understanding Mainframe Economics

An IBM® Redbooks® Point-of-View publication

By **John J. Thomas**
IBM Senior Technical Staff Member

Highlights

The cost of running a workload can be lower on the mainframe than on other platforms you might be considering.

When considering the economics of the mainframe, simple benchmark test results can be misleading. Variability in workload demand and workload management are examples of factors that affect the cost of delivering a workload.

Typically, administrators achieve significant operational efficiencies on the mainframe. A variety of workloads can be virtualized and consolidated by using Linux on System z, which results in lower overall software costs when compared to other platforms.

Customized total cost of ownership studies are offered by IBM that can look at different line items of cost to help you determine whether the mainframe is the correct platform for your workloads.

Mainframes still make good economic sense

Some people say that “The mainframe is too expensive!” And, if you compare the price of an IBM System z® processor core to an x86 processor core, the x86 core does appear to be much cheaper. Given this difference, and considering that x86 technology has made steady improvements in performance, does the mainframe make economic sense anymore?

The question applies to both traditional, mission-critical (sometimes referred to as “legacy”) System of Record workloads and to today’s emerging workloads supporting the growing fields of cloud, social, mobile, and analytics. Is it cost-effective to deploy all these workloads on the mainframe?

To answer these questions, it is important to understand mainframe economics, particularly cost per workload. When you consider various real-world factors, you find that the mainframe can deliver both traditional and emerging workloads in an extremely cost-effective way.

Cost per workload

The most accurate way to compare the cost of different platforms is to look at the cost of delivering a given workload. After you have identified a specific workload to study, you can compare different deployment options.

What happens if you choose to do nothing and retain your current deployed platform for an existing workload? Do costs stay linear or do they go up as the application grows over time? What happens if you stay on the current platform but optimize the hardware and software stacks? What happens if you redeploy the workload on other platforms? Each option has its own workload delivery cost.

The key steps in identifying the cost of a workload are shown in Figure 1 on page 2. As the diagram shows, first you must determine the infrastructure needed to deliver the workload. Then you can compare the total cost of that infrastructure, taking into account various line items of cost.



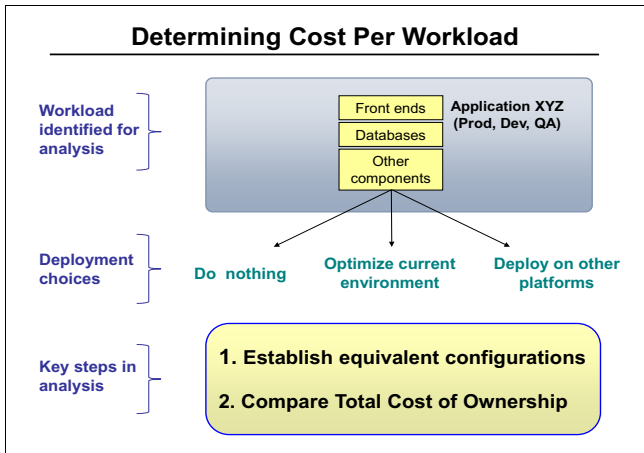


Figure 1 Determining cost per workload

Establishing equivalent configurations for workload delivery

There are different approaches to establishing equivalent configurations for workload delivery.

A *bottom-up* approach looks at low-level platform factors (clock speeds, cycles per instruction, the effect of I/O, and so on) and relies on low-level benchmark tests or simple spreadsheet calculations to determine equivalence. At the other end of the spectrum (see Figure 2) is a *top-down* approach in which you derive equivalence by observing the real world behavior of the same workload deployed on two different side-by-side platforms. This latter approach is ideal but often not practical. Few clients have the same workload deployed on different side-by-side platforms. When the correct low-level factors are selected to represent your real-world deployment, bottom-up estimates begin to approach top-down estimates in terms of accuracy.

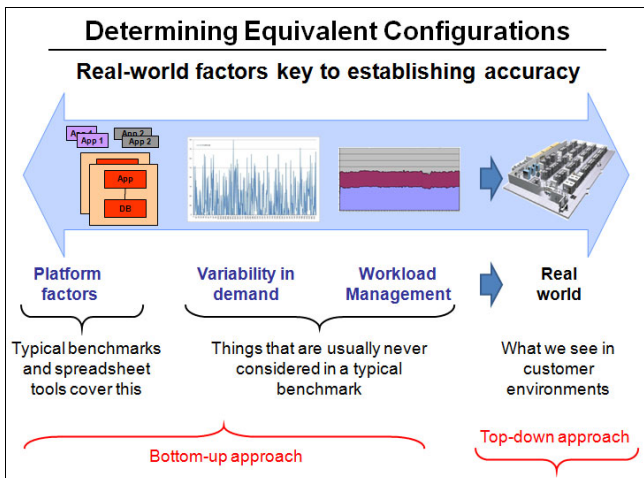


Figure 2 Establishing equivalent configurations

A simple, low-level benchmark test designed to stress an application will yield some data points, but the results can be misleading, because such tests often do not take into account many real-world aspects of delivering a workload. To determine platform equivalence for real-world scenarios, you must consider factors, such as provisioning capacity to meet peaks in demand, collocation with other applications, maintaining the required service level for high-priority workloads when lower priority workloads are present, and so on.

Each of these elements has an effect on the configuration needed to deliver a given workload.

Variable demand

Many benchmark tests are run in steady-state fashion. However, real workloads experience variance in demand, with highs and lows over the course of the day. System administrators provision enough capacity to meet the peaks, such as when a service level agreement (SLA) calls for "enough server capacity to meet 97.5% of incoming requests." The extra capacity needed to handle peaks in demand is referred to as *headroom*.

There are several interesting observations in this space. When you aggregate workloads with varying demand on a shared server, the amount of variability exhibited by the combined workloads is lower than when each workload is deployed on its own server. The more workloads you can aggregate, the smaller the overall variance in demand. In other words, the more workloads that are operating on a shared server, the lower the overall headroom requirements. Consequently, bigger servers with capacity to run more simultaneous workloads can be driven to higher average utilization levels without violating SLAs, therefore reducing the cost per workload.

Consider this example data from a data center (see Figure 3 on page 3). IBM collected CPU utilization data for several large and small servers, all running the same commercial application.

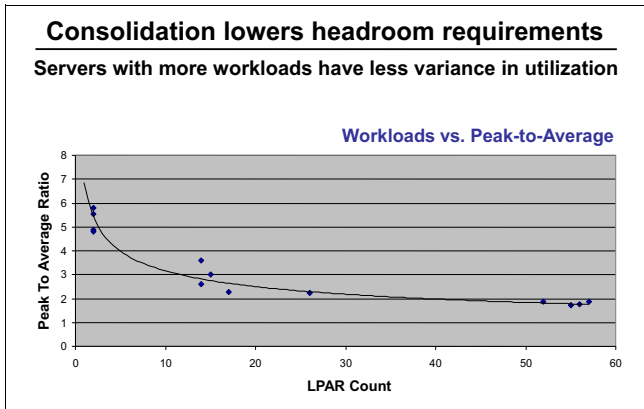


Figure 3 Effect of consolidation on headroom requirements

You can see the headroom requirements by looking at the peak-to-average CPU utilization ratio for each server. We plotted the headroom requirement for each server against the number of workloads it was running. The resulting plot shows that headroom requirements are lower on servers that run more workloads.

Put another way, when trying to establish the configuration needed to deliver a given workload, you have to take into account the variance in workload demand, what other workloads it can be pooled with, and the size of the server. Collecting actual CPU utilization data for the workload over time allows us to make reasonably accurate estimates on the variance and headroom requirements.

Although standard virtualization techniques can help the x86 platform improve average utilization levels, the mainframe excels at running many workloads together. This can significantly reduce headroom requirements and therefore lower overall resource requirements.

Mixed workloads with differing priorities

Servers must support both high-priority and low-priority workloads when sharing resources, such as in on-premises cloud environments where there are often multiple tenants with different priorities. So, what is the desired behavior when mixing workloads with different priorities? For one thing, lower-priority workloads need to yield resources to higher-priority workloads when required, but they must be allowed to consume unused resources when available. In addition, the performance of the higher-priority workload must not degrade when lower-priority workloads are added to the same platform.

In tests that compared System z mainframes to virtualized x86 systems, IBM observed the effect of differences in workload management. As shown in Figure 4, high priority workloads on System z (IBM z/OS® or z/VM®) did not degrade when lower priority workloads were added. The higher priority workloads ran with no loss in throughput or increase in response time. In comparison, higher priority workloads running on standard x86 virtualized environments did exhibit degradation when lower priority workloads were added.

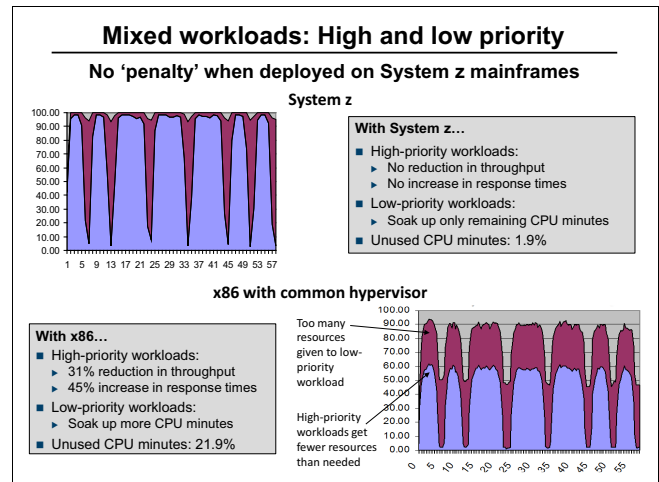


Figure 4 Effect of mixed workloads on system performance

With common x86 virtualization technology, the standard way to maintain required high-priority service levels is to segregate the workloads onto different servers. So, most x86 virtualized deployments put the Dev (Development), Test, and QA (Quality Assurance) environments onto servers that are separate from Production. This wastes any spare capacity on the machines, because no other workloads can be added to those servers without affecting SLAs. The resulting x86 configuration tends to be more expensive due to the increase in processor cores, associated software licenses, and management requirements.

These are just two of the factors that can influence efforts to establish equivalent configurations. Additional factors will affect the configurations needed for specific workloads, including I/O requirements (both for the workloads and for surrounding applications, such as batch), memory, collocation, and so on.

A detailed analysis of the workload in its context (that is, actual data from the workload coupled with an understanding of its surrounding workloads) helps to establish accurate equivalent configurations.

Total cost of ownership

With equivalent configurations established, we can compare their total cost of ownership (TCO).

The key is to identify all the line items of cost. Server hardware is just one line item and that the total cost of delivering a workload is greatly affected by other line items, such as software licenses (both up-front and yearly support and subscription charges), storage, network, labor, power and cooling, and so on. Then, there is the cost of supporting additional environments (Dev, Test, and QA) and providing for disaster recovery. Time factors can drive cost, too, along with organic business growth and planned business changes, both of which affect capacity. Periodic changes, such as technology refreshes, have an impact on cost over time. Even nonfunctional requirements, such as availability, security, and resiliency, can drive additional costs.

Studies show high *core compression* (a reduction in the number of processor cores needed) when adding workloads to the mainframe and high *core expansion* (*core proliferation*) when taking workloads off the mainframe.

For example, consolidating workloads using Linux on System z usually yields lower costs than a scale-out x86 deployment. Core compression on System z is the key, because with fewer processor cores needed to run the same workload, fewer middleware licenses are required. Depending on the vendor and software (such as application servers and databases that are priced on a per-processor basis), the reduction in the overall cost of a solution can be substantial.

Conversely, offloading traditional workloads from the mainframe to scale-out platforms leads to dramatic increases in the number of processor cores needed, driving up spending on software licenses, management software, labor, network, power and cooling resources, and facilities.

IBM has introduced many pricing and licensing models in support of new workloads. These range from specialty processors for certain types of workloads to special licensing models and solution-edition pricing geared toward specific workloads, such as cloud and mobile. In addition, using accelerators can deliver some workloads at a much lower cost. For example, the IBM DB2® Analytics Accelerator dramatically lowers the cost of analytics on the mainframe.

Technology refresh cycles can also affect cost. Most IT equipment is refreshed on 2 - 7 year intervals (often every 3 - 4 years), and each time that this happens,

distributed servers are repurchased (or re-leased), typically with some additional capacity. In contrast, with a growing mainframe, clients typically only have to purchase the additional (incremental) capacity; existing capacity is often carried over to the new hardware. And in some cases, the effect of technology refresh cycles can be even greater. In many non-mainframe deployments, the old and new systems must coexist for months while the refresh is in progress, requiring additional space, power, licenses, and so on, until the work is finished.

There are many other major line items, including costs for disaster recovery, labor efficiency, system management tools, and so on. Adding up all of these line items gives us the TCO for a given platform.

What's next: How IBM can help

When considering the economics of the mainframe, remember that mainframes are built to support economies of scale (see Figure 5). This means that the incremental cost of adding workloads to an existing mainframe can often be substantially lower than the incremental cost in a linear x86 scale-out model.

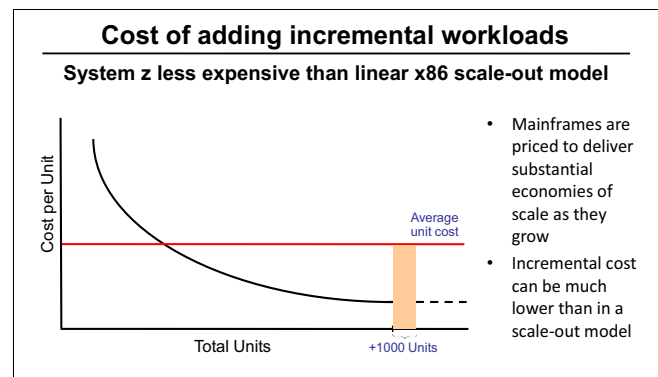


Figure 5 Added cost of incremental workloads on mainframe

A prime design point of the mainframe is to support many workloads running together, driving high levels of system utilization. When utilized well, the mainframe can deliver many workloads at the lowest cost per unit of work.

Establishing equivalent configurations based on real-world workload aspects and then calculating the total cost of ownership gives us an accurate comparison of platforms. Studies that take into account various aspects of the workload, and all of the relevant line items of cost, repeatedly demonstrate that the mainframe delivers both traditional and emerging workloads at the lowest cost per workload.

IBM offers Total Cost of Ownership studies that are customized to the client. This is a no-charge service that analyzes the client workloads and produces a detailed cost comparison for different deployment options. Contact IBM to see how you can obtain this service to quantify mainframe economics.

Resources for more information

For more information about the concepts highlighted here, see the following resources:

- ▶ IBM System z mainframes:
<http://www-03.ibm.com/systems/z/?lnk=mprSY-sysz>
- ▶ IBM System z mainframe software:
<http://www-01.ibm.com/software/os/systemz/>
- ▶ Advantages of IBM System z and enterprise computing:
<http://www-03.ibm.com/systems/z/advantages/index.html>
- ▶ Linux on System z:
<http://www-03.ibm.com/systems/z/os/linux/>
- ▶ IBM zEnterprise@ innovations:
<http://www-03.ibm.com/systems/z/hardware/features/index.html>
- ▶ IBM IT economic assessments:
<http://www-01.ibm.com/software/info/eagletco/>

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to: *IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.


This document, REDP-5127-00, was created or updated on August 14, 2014.



Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DB2®
IBM®
Redbooks®
Redbooks (logo) 
System z®
z/OS®
z/VM®
zEnterprise®

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.