Compare z/OS and distributed terminology

Level: Introductory

Barbara Hampson (hampson@us.ibm.com), Application and Integration Middleware Software Manager, WebSphere Application Server FVT, IBM
Alan Beaubien (wbeaubie@us.ibm.com), IBM Software Group, Application and Integration Middleware Software WebSphere SWAT Team, IBM

09 Jun 2005

Get an overview of terminology differences between the distributed computing environment and mainframe environment. Today, software is becoming increasingly versatile across hardware and operating system boundaries, causing the boundaries to blur and overlap. Developers, testers, and technical support teams are no longer just distributed or mainframe people. They now need to understand and be fluent in concepts and terminology from both sides. Barbara Hampson and Alan Beaubien provide handy cheat sheets that help you compare terms.

Introduction

This article compares terminology differences between the distributed computing environment (mid-range computers running UNIX® and Windows®) and mainframe environment (large scale computers, such as zSeries® hardware). Ten to twenty years ago, these two environments were completely isolated from each other; those who worked in the mainframe environment rarely had the need or desire to cross over to the distributed environment, and vice versa. Today that is no longer true. Software is becoming increasingly versatile across hardware and operating system boundaries, causing the boundaries to frequently blur and overlap. Developers, testers, and technical support teams find themselves no longer just distributed or mainframe people. They're increasingly required to understand and be fluent in concepts and terminology from both sides.

One of the stumbling blocks in developing this fluency is to understand the terminologies of the two environments. If someone refers to a dataset in mainframe lingo, what does that mean in a distributed environment? This article demystifies and makes the connection between the distributed and mainframe terms. Whether you're in development, testing, support, sales, or consulting, if your work crosses these boundaries, you can use the cheat sheets to easily navigate the terminology barrier.

The intent here is not to provide an exhaustive comparison of the different terminologies, but to offer what we believe to be the key or major pieces of the terminology that you'll encounter regularly as you make your way through the mainframe and distributed computing environments.

z/OS operating system and software

z/OS® is the operating system for the IBM® z900 (zSeries 900) large mainframe servers. It's the next generation of the OS/390 operating system, which in turn was an upgraded and renamed version of the MVS (Multiple Virtual Storage) operating system. z/OS:

- Is highly scalable. Scalability, in simple terms, is the ability of a computer application or product (either hardware or software) to continue functioning well even after it's changed in size or volume, to meet user requirements.
- Offers advanced system security with RACF® (Resource Access Control Facility), a key component of the z/OS Security Server, and System Authorization Facility (SAF), a standard feature of the z/OS operating system. These two security systems let system administrators verify

and identify system users, protect system resources, and so on.

- Employs 64-bit z/Architecture as well. A 64-bit processor is a microprocessor with a word size of 64 bits, a requirement for data and memory intensive applications, as well as high-performance servers, such as the WebSphere® Application Server. z/OS supports Web- and Java-based applications, and the performance is benefited greatly by the 64-bit architecture employed in the z/OS operating system.

z/OS also features UNIX System Services (USS), which allows UNIX applications to run in the mainframe environment.

zSeries hardware

zSeries hardware is currently comprised of four different processor models, all based on z/Architecture. The processor models include the z800, z890, z900, and z990 series. Each model is designed with different customer needs in mind.

The design of the z800 model is for customers whose capacity requirements are less than those of the z900 model, while still providing the advantages of zSeries technology. The z800, in addition to supporting traditional z/OS workloads, can host hundreds of Linux™ images running commercial applications under the z/VM® operating system.

z900 servers are based on the z/Architecture, which enables applications to have large memory and can help eliminate bottlenecks associated with the lack of addressable memory. Like its smaller z800 counterpart, the z900 can also host hundreds of Linux images under the z/VM operating system. The z900 servers offer a variety of configurations, ranging from 1- to 16-way processor engines.

The z890 and z990 servers are the newest and most powerful zSeries machines to date, as well as the most technologically advanced. Offering such unique features as capacity on demand, the z990 provides greater horsepower (offering as many as 32 processing units), increased capacity, and greater scalability. The z890 and z990 models also offer zAAP (zSeries Application Assist Processor) technology, a specialized processing unit designed specifically to enhance productivity at a lower cost in the z/OS Java execution environment.

All zSeries servers have the capability of automatically directing resources to priority work through Intelligent Resource Director (IRD). The zSeries IRD combines the strengths of three key technologies: Workload Manager (WLM), logical partitioning (LPAR) and Parallel Sysplex clustering.

IRD technology allows zSeries servers to dynamically reallocate system resources to a specific application or applications, based on workload demands experienced by the system at any given moment. The LPAR technology provides for the creation of 15 or more separate LPARs, meaning you can have 15 or more separate systems running on the same physical machine. Parallel Sysplex clustering allows for high speed, read/write data sharing, and resource sharing among all z/OS images in a cluster. A Parallel Sysplex cluster can have as many as 32 separate z/OS images.

Distributed operating systems and software

Distributed systems or distributed environment describes a group of mid-range computers running software that communicates together so as to appear to the end user as a single local machine. Distributed systems platforms encompass Linux, UNIX (a family of operating systems), and Windows, all multi-user, multi-tasking, time-sharing operating systems.

Middleware is the layer of software on distributed systems that provides standard services such as:

- Naming (association of entities and identifiers)

- Directory services (maintaining the association of attributes to entities)
- Persistence
- Concurrency control
- Event distribution
- Authorization
- Security

Middleware helps tie together the distributed and mainframe legacy systems. It lets you migrate mainframe applications that could not otherwise interoperate or serve remote requests.

Distributed processing simply refers to computer systems that run applications using more than one computer or processor. Local Area Networks (LANs) are an example of distributed processing. In a LAN, a server normally contains applications and data storage that are shared among many computers; a single program can run simultaneously at different locations within the local network, accessing data as needed from a central location. Parallel processing is another example of distributed processing. In this case, applications are run on a single computer, but on multiple processors (CPUs) within the computer. The processors can cooperatively communicate with each other, or they can run applications completely independent of each other with one processor managing and distributing the work among the remaining processors.

Distributed hardware

Distributed hardware comes in many flavors -- Intel processors, pSeries® workstations, Solaris SPARC workstations, expandable hardware like the IBM eServer™ BladeCenter, and so on. At one time, distributed operating systems were closely linked to their hardware. For example, Intel processors only ran Windows operating systems. With the advent of Linux, the lines have blurred. Today Linux runs on traditional UNIX machines (such as Solaris workstations), Intel machines, and even on the mainframe zSeries machines.

Most users know distributed machines as workstations or a computer used to run applications that require a moderate amount of computing power and relatively high quality graphics capabilities. The majority of workstations are single-user computers, typically linked together to form a LAN, although they can also be used as stand-alone systems. While the terms workstation and PC are sometimes used interchangeably, workstations are generally a higher power machine, with better multi-tasking abilities and more processor speed than a PC.

How the two worlds are becoming more integrated

Especially in the WebSphere family of products, the distinction between the distributed environment and the mainframe world is shrinking all the time. WebSphere is the middleware that helps provide a layer of isolation for the application developer from the underlying platform due to the virtue of the J2EE (Java 2 Platform, Enterprise Edition) programming model. With WebSphere Application Server Version 5.0, the distributed and the mainframe versions of that software started showing numerous similarities, especially in the Web-based administrative console. And some of the code base was common between the different platforms. With WebSphere Application Server Version 6.x, even more of the software code base is common between the platforms. When Version 7.x is released, the code base is expected to be virtually the same.

Further evidence of the blurring distinction between the different systems is that z/OS features USS as a standard subsystem of the operating system. With each new release of the z/OS operating system, USS becomes more prominent.

In modern mainframe environments that run WebSphere Application Server, the development work takes place on a distributed machine and the application is then ported to the mainframe.

While there will continue to be fundamental differences between the operating systems and the hardware platforms, in today's operating environment, it's almost essential to have some knowledge of both z/OS and the distributed operating systems.

Terminology

This section provides a quick reference and comparison of commonly used terms in the mainframe and distributed environments.

Unique terminology terms have different names in mainframe and distributed environments. This terminology identifies mainframe and equivalent distributed. Any similarities or differences are also identified, where applicable.

Similar terminology terms are identical in both environments; however, they might have different meanings. This terminology explains the similarities and differences between the two environments.

Unique terminology

In this section, the Z terminology is on the left; the distributed terminology, an explanation, and similarities and differences are on the right.

ABEND (ABnormal END of program)

Distributed terminology: Crash, failure, hang
Explanation: Any time a program terminates with anything other than a normal completion (see condition code).
Similarities and differences: ABEND. In the z/OS arena, an ABEND results from an operator or user job cancellation, or the result of an error in a job or program. In the distributed environment, a hang is when a process stops functioning, but it does not terminate. The process either terminates manually or eventually terminates due to lack of memory.

Base control program (BCP)

Distributed terminology: Kernel
Explanation: The core processes of an operating system, which load first and remain in main memory, provide all of the base services needed by an operating system and auxiliary services.
Similarities and differences: BCP is the core MVS, OS/390®, or z/OS operating system, not including subsystems such as JES (job entry subsystem), ISPF (interactive system productivity facility), and TSO/E (time sharing option extensions). Most UNIX systems consist of a kernel plus a shell (user interface). UNIX environments have kernels with many different types of shells, originally Bourne, then C-shell, Korn, Bash (Bourne again shell), and so on. The Windows equivalent is the NT kernel (its Win32 executive), the command-line shell, plus tools, and background services.

Catalog

Distributed terminology: Directory (folder)
Explanation: File structure where datasets and files are stored on the system. It can also contain subdirectories.
Similarities and differences: Like dataset and file naming conventions, the catalog and directories where they are stored also have their own conventions.

- Dataset catalog structure:
  The mainframe environment, the catalog is the equivalent of the PC's directory. The catalog is a dataset that contains information about other datasets on the system, including size, location, type, and format. z/OS employs the Integrated Catalog Facility (ICF), which uses a master catalog and user catalogs.

  The master catalog is where the search begins for information on a dataset. It typically contains information about system datasets (those datasets with a high-level qualifier of SYS1) and information about each of the user catalogs. The master catalog contains entries for all aliases (high-level qualifiers). The aliases are associated with user catalogs at the discretion of the system programmer.

If an entry for a dataset exists in either the master catalog or a user catalog, the dataset is said to be *cataloged*. You can find cataloged datasets by simply typing in the name of the dataset. The catalog system automatically locates the dataset. If an entry for a dataset does not exist in either the master catalog or a user catalog, it is *uncataloged*. In that case, the user must know the DASD or tape volume on which the dataset resides. An entry for the uncataloged dataset is in the DASD or tape volume's VTOC (volume table of contents). However, uncataloged datasets cannot exist on SMS-managed volumes. We do not recommend having uncataloged datasets on your system.

- File directory structure:
  In the Windows operating system: C:\

  ```
  <directory>\<subdirectory>\<subdirectory>\<filename>.<extension>
  ```

  Where C: is the letter representing the disk drive where your file resides. In Windows, directories and subdirectories are commonly called folders and represented by a folder icon.
  In UNIX operating systems:

  ```
  /<directory>/<subdirectory>/<subdirectory>/<filename>.<extension>
  ```

Channel

Distributed terminology: Port
Explanation: An interface (internal or external) on a computer to which you can connect a device.
Similarities and differences: A zSeries channel is a specialized computer used in the IBM mainframe architecture that controls I/O transfers between physical devices and the CPU. Channel also refers to the cable used to physically connect the channel processor to peripheral devices. Channels used to connect to processors and devices with *bus and tag* connections. Bus cables carry data and tag wires carry control information. The traditional bus and tag was replaced by ESCON® (enterprise systems connection architecture) fiber optics during the 1990s, and ESCON, although still available to some degree, was replaced by FICON® (fibre connections) in the late 1990s. One FICON channel can handle as much data transfer as eight ESCON channels.

In the distributed environment, a single computer can have internal and external ports. Internal ports are used for:

- Connecting disk drives
- CD/DVD drives
- Graphics cards
- Modems

External ports are available for:

- Connecting printers
- Mice
- Displays
- Sound devices
- External disk drives
- Modems
- CD/DVD drives

Command list (CLIST) (also REXX -- Restructured Extended Executor Language)

Distributed terminology: Shell script
Explanation: A file containing a series of commands you need to execute. You often use it to automate repetitive tasks by combining complex commands into a single command file.
Similarities and differences: In the distributed environment, shell scripts are common to UNIX operating systems and a variety of shells. You can find similar functions in a DOS or Windows batch file (.bat file). In the mainframe world, CLIST has largely been supplanted by REXX as the CLIST tool and language.

Condition code

Distributed terminology: Return code (RC)

Explanation: This completion code identifies whether a program has ended normally or abnormally (see ABEND).
Similarities and differences: On the mainframe, it's a four-digit hexadecimal code. In most circumstances, a condition code of 0000 indicates a normal termination of a program. A condition code of 0004 almost always indicates a normal termination for which there might be some seemingly abnormal condition associated. You should view a 0004 condition code as informational. It warrants a perfunctory examination by the programmer, but any error condition associated with a 0004 condition code is almost always non-fatal. Condition codes of 0008, 0012, and 0016 indicate an abnormal program termination (ABEND).

In the distributed environment, a RC of 00 indicates a normal termination. Major and minor codes can also follow RCs, further isolating the reason for failure.

Datasets (or data sets)

Distributed terminology: Files
Explanation: A way to group and store information and identify it by name (filename or dataset name). Almost all information stored in a computer must be in a file or dataset. Public files are viewable and accessible to the user, while sensitive system files are often protected and might be hidden from the user.
Similarities and differences: The dataset is the mainframe equivalent of a file in the UNIX/Windows arena. In the mainframe environment, a dataset with a high-level prefix (or qualifier) of SYS1 is a system dataset, containing information related to system parameters, system startup, catalog info, and so on. The UNIX/Linux/Windows equivalents are configuration files (.ini files and .profile file).

In the mainframe environment, creating a dataset is much more complicated than creating a file in UNIX/Linux/Windows. On the mainframe, the data set must actually be allocated (space must be reserved) before the data set can be created. Adequate space on the disk must exist to build the dataset. Allocating a dataset means specifying a dataset name (adhering to the naming conventions established for that particular mainframe system), a primary allocation size, a secondary allocation size, the units of allocation (tracks or cylinders), the volume where the data set is to be located, and several other variables.

- The naming convention of a dataset is:

  ```
  <high level qualifier>.<qualifier>.<qualifier>
  ```

- The naming convention of a file is:

  ```
  <filename>.<extension>
  ```

  Each dataset name qualifier can have as many as eight characters. The total dataset name, including the period or dot (.) between each qualifier, cannot exceed a maximum of 44 characters.

Dump (or standalone dump)

Distributed terminology: Core dump
Explanation: Copying data from memory to a display screen, printer, or file. It's usually triggered when a process is aborted or abnormally terminated due to internal processing.
Similarities and differences: A standalone dump is when you copy the contents of a computer's main storage to another storage device, usually for debugging a programming problem. You primarily use a standalone dump in the mainframe environment. When an IBM mainframe stops operating because of a *hard wait state* or a *machine check* condition, an operator or system programmer can perform a store status operation. A hard wait implies that instruction execution has been halted and the instruction counter (IC) is no longer advancing. A machine check is typically the result of a hardware malfunction. The Store Status operation saves the program status word (PSW), also known as the IC, and the 16 general purpose registers (GPRs), which tell exactly what the computer was doing when it stopped operating, and then performs a standalone dump to copy real and virtual memory to tape or DASD (disk storage).

Initial Program Load (IPL)

Distributed terminology: boot (bootstrap)
Explanation: This is the first step in loading and initializing the operating system into the machine.
Similarities and differences: IPL on the mainframe is the equivalent of booting up a PC or

workstation.

Job (batch Job)

Distributed terminology: Process
Explanation: This process allows you to run a program (a series of computer instructions). It can also be a sequence of one or more related programs.
Similarities and differences: You can normally start and stop a process manually, whereas a job is a bit different. You can start a job manually or automatically and end it manually or automatically. In the mainframe environment, a job (or a batch job) is a set of programs linked together with Job Control Language (JCL) to perform a desired task. An individual user usually submits a batch job; however, you can use scheduling software to automate the job submit process so certain batch jobs run at designated times. A batch job is not to be confused with a batch (.bat) file in the distributed environment, which is a series of commands.

Parallel processing

Distributed terminology: Multi-tasking/multi-processing
Explanation: This is an environment where you can execute multiple processes at the same time, independently of each other. The terms multi-tasking and multi-processing are often used interchangeably, although multi-processing implies that more than one CPU is involved.
Similarities and differences: Parallel processing is the processing of program instructions by dividing them among multiple processors with the objective of running a program in less time. Parallel processing has continued to evolve, and it is this philosophy that ultimately led to the reality of Parallel Sysplex in the mainframe environment.

Split screen

Distributed terminology: Window
Explanation: A way you can partition or divide the viewable display area of a computer screen to enable different programs to run and be viewable in different display areas, each independent of the other. It provides the ability to switch back and forth between the display areas without interrupting the performance of the activity within that window.
Similarities and differences: When you employ the split screen on the mainframe, you can only have two screens. On the distributed environment, there is theoretically no limit to the number of windows you can have on your machine; your only limitation is the amount of virtual memory available.

Started task

Distributed terminology: Service (also known as daemon)
Explanation: This program or process runs in the background of an active session. Unlike processes run by the user, it's not interactive with the user. However, it can respond programmatically to events or operations and can spawn other processes.
Similarities and differences: In the distributed environment, a service is not tied to a specific log in session, so it continues to run even if the user logs out. In the mainframe environment, you typically use started tasks for long-running address spaces needed for system functions (either "IBM operating system functions" or other software vendor product address spaces) or critical applications like CICS®, DB2® , and IMS. You can run a user program as a started task but they are generally run as batch jobs. Started tasks might also have more dataset access and utility privileges (from a security standpoint), so using them is normally controlled by protecting PROCLIBs (the started task). This is defined to JES and requires a special definition in your security product (RACF, ACF2, or Top Secret) for the started task.

Sysplex/Parallel Sysplex

Distributed terminology: Domain
Explanation: A group of computers (or processors) administered as a single entity, allowing one administrator to manage all the computers within the domain or sysplex using a common set of procedures.
Similarities and differences: A sysplex or system complex refers to a processor complex that forms by coupling multiple z/OS processors into a unit that you can manage as one system rather than a series of standalone systems. You can share storage, hardware devices, and workload among the members of a sysplex.

The Parallel Sysplex configuration allows for parallel transaction processing in a sysplex environment and provides resource sharing, workload balancing, and continuous availability. Initially, Parallel Sysplex supported up to 32 z/OS systems, presenting them as a single systems image (*cluster*). The major hardware components consist of a coupling facility, a sysplex timer, and links to both.

Task

Distributed terminology: Thread
Explanation: A single unit of work contained within a program that works independently from other units of work; a job or process can have many tasks or threads. You can usually schedule tasks and threads by priority to optimize utilization.
Similarities and differences: In the mainframe environment, a task is represented by an MVS control block.

Thin client (also known as 3270 terminal, 3270 emulator, dumb terminal)

Distributed terminology: Thin client
Explanation: A thin client is typically a PC or terminal that performs doesn't perform application processing. It only processes keyboard input and screen output, functioning as an input/output device only. All application processing is performed on a remote (separate) server. A thick or fat client is a typical desktop or even a laptop PC in which all processing occurs.

TSO/E

Distributed terminology: Shell
Explanation: This user interface functions as a command interpreter to the Base Control Program or kernel.
Similarities and differences: In the mainframe environment, TSO/E is an element of the z/OS operating system that provides an online interactive environment for programmers and users. TSO/E is most readily associated with the ISPF/PDF (interactive system productivity facility/program development facility), which runs on TSO/E. In a UNIX environment, there are many shells available -- the Bourne shell being one of the most common.

USS

Distributed terminology: UNIX (also AIX®)
Explanation: A Multi-user, multi-tasking operating system consists of a kernel, system utilities, and configuration files. AT&T Bell Laboratory employees developed UNIX in the 1960s as an interactive operating system that runs on a workstation, allowing multiple users to simultaneously execute multiple functions.
Similarities and differences: UNIX has grown and developed into many different forms, such as the IBM AIX operating system and Linux operating systems, and is now available on z/OS. USS, formerly MVS/Open Edition, is a fully functional UNIX environment running under z/OS. USS is fully compliant with all POSIX standards. It is implemented in z/OS as an operating system that you can switch within z/OS. You can reach USS by entering the command OMVS from ISPF Option 6, or in a menu-driven format by entering the command ISHELL, also from ISPF Option 6.

For more on USS, see the IBM documentation at UNIX System Services -- A Powerful Blend.

Virtual storage

Distributed terminology: Virtual memory
Explanation: Virtual memory or storage is a way to provide programs with more memory than the computer physically has available, giving programs the illusion they have large amounts of main storage available only to themselves. It uses a combination of hardware and software to allow for a larger address space not tied to the physical memory of the computer. The technique works by allowing programs to address lots of virtual memory, making the operating system page the required data in and out of real main storage, and to and from a paging device at the appropriate time.
Similarities and differences: In a mainframe environment, the technique enables you to use inexpensive DASD instead of expensive main storage.

Similar terminology

This section explains terms that are identical in both environments, but might have different meanings. The Z terminology is on the left and the distributed terminology, an explanation, and similarities and differences are on the right.

Address space

Distributed terminology: Address space (process)
Explanation: The amount of virtual storage allocated to an executing task.
Similarities and differences: While both environments use an address space to identify the storage area being used by a task, there are subtle differences. Within z/OS, the address space typically refers to the space used by a batch job, a system task, or a TSO user. In the distributed environment, each process has its own address space. A process on a distributed environment is limited to one address space whereas a single process on mainframe might use multiple address spaces.

Local/remote

Distributed terminology: Local/remote
Explanation: Traditionally used in networking terms, local/remote refers to whether the resources are on or directly connected to your own system (local) or if you must connect through other systems to access the resources (remote).
Similarities and differences: In a mainframe environment, local/remote was traditionally used for distributed systems because everything ran off a single main server (hence everything was local). It's now possible to have local and remote resources on a mainframe with mainframe client/server environments.

Proxy server

Distributed terminology: Proxy server
Explanation: A server that provides a shared cache of resources for the client machines at a site or across several sites. Web proxy servers reduce load on the wide-area network and act as a security measure across a firewall.

Server (eServer)

Distributed terminology: Server
Explanation: A computer or device manages resources on a network. A server is often a single computer running its own tasks. You can also have a situation with many processors on the same computer (multi-processor) where different servers manage different resources.
Similarities and differences: Traditionally, "server"was used in a distributed environment. Today, you can also use the term in the mainframe environment to describe the mainframe capabilities (the world's largest server) in the client/server arena. You can also have multiple servers on one distributed machine (single processor), for example FTP Server, Web server, mail server, and so on.

WLM

Distributed terminology: WLM
Explanation: Workload management helps you to more efficiently and effectively use system resources, such as processors and storage. It also helps you to manage workload, workload throughput, and system utilization and responsiveness.
Similarities and differences: In z/OS, workload management is achieved interactively, as the system programmer is required to assign performance goals and a business importance level to each goal. Based on specified input criteria, WLM decides how much resource, such as CPU and storage, should be given to workload components (jobs, started tasks, and so on) to meet the goal. WLM constantly monitors the system and adapts processing to meet the goals. With the advent of z/OS, a new feature, Intelligent Resource Adapter (IRD), was introduced as a complement to WLM. IRD is designed to allow WLM to dynamically move resources to a data center's most important work.

WebSphere Application Server uses WLM to control the performance of and the number of application server regions in z/OS. You can deploy WebSphere applications within a WebSphere generic server. You must define one or more server instances on one or more systems within the WebSphere node. Each server instance consists of a controller region and one or more servant regions. You need to start controller regions as MVS started tasks, while servant regions are started automatically by WLM on an as-needed basis. WebSphere servers are configured, by default, to allow only one servant region. The Application Server administrator must use the administrative console to allow the generation of multiple servant regions.

Once the Multiple Instances Enabled feature has been selected, WLM starts the servant regions (as needed) based on the WLM criteria (performance, workload, and importance), as specified by the system programmer.

Distributed environments are multi-tasking environments. The operating systems handle managing the workload and resource use within the system. However, distributing work across multiple platforms and systems present more challenges requiring a combination of system tools and processes to meet the level of control and reliability available on a mainframe system. In the distributed environment, the WebSphere middleware contains built-in features to handle workload distribution across servers with the WLM functionality.

Workstation

Distributed terminology: Workstation
Explanation: This term refers to a single-user computer with a moderate amount of computing power and high quality graphics. It is often used interchangeably with PCs; however, PCs are often standalone, while workstations are typically connected into a network of workstations.
Similarities and differences: Historically, workstations and PCs were the staple of the distributed environment, while mainframes used dumb terminals to access the mainframe where all of the processing was done on the mainframe and displayed back to the user. Over the past fifteen years or so, this trend appears to be changing. You can now use workstations for dual purposes. As clients, you can use the workstations for moderate computing power, while accessing the mainframe for heavy computing power and legacy data storage systems.

Cheat sheets

This section has a table with a cheat sheet for the mainframe -- sorted by mainframe terms -- and a table for the distributed environment -- sorted by distributed terms.

Mainframe

Mainframe term	Distributed term	Description
ABEND	Crash, failure, hang	Any time a program terminates with anything other than a normal completion.
Address space	Address space (process)	The amount of virtual storage allocated to an executing task.
BCP	Kernel	The core processes of an operating system.
Catalog	Directory (folder)	File structure where datasets and files are stored on the system.
Channel	Port	You can connect a device to this interface.
CLIST	Shell script	This file contains a series of commands to execute.
Condition code	RC	This completion code identifies whether a program ends normally or abnormally.
Datasets	Files	A way to group and store information and identify it by name (filename or dataset name).
Dump	Core dump	Copying data from memory to a display screen, printer, or file.
IPL	Boot (bootstrap)	This is the first step in loading and initializing the operating system into the machine.

Job (batch job)	Process	A running program (a series of computer instructions)
Local/remote	Local/remote	This refers to whether the resources are on or directly connected to your own system (local) or if you must connect through other systems to access the resources (remote).
Parallel processing	Multi-tasking, multi-processing	This environment allows you to execute multiple processes at the same time, independent of each other.
Proxy server	Proxy server	A server that provides a shared cache of resources for the client machines at a site or across several sites.
Server (eServer)	Server	This computer or device manages resources on a network.
Split screen	Window	This is a way to partition the viewable display area of a computer screen to enable different programs to run in different display areas.
Started task	Service/daemon	This program or process runs in the background of an active session; not interactive with the user.
Sysplex (parallel)	Domain	A group of computers (or processors) administered as a single entity.
Task	Thread	This single unit of work within a program works independently from other units of work.
Thin client, 3270 terminal, 3270 emulator, dumb terminal	Thin client	This PC or terminal doesn't perform application processing. It functions as an input/output device only.
TSO/E	Shell	This user interface functions as a command interpreter to the BCP or kernel.
USS	UNIX	This multi-user, multi-tasking operating system consists of a kernel, system utilities, and configuration files.
Virtual storage	Virtual memory	A way that provides programs with more memory than the computer physically has available.
WLM	WLM	Utilize system resources to help manage workload, workload throughput, and system utilization and responsiveness.
Workstation	Workstation	This single-user computer has a moderate amount of computing power and high-quality graphics.

Distributed

Distributed term	Mainframe term	Description
Address space (process)	Address space	The amount of virtual storage allocated to an executing task.
Boot (bootstrap)	IPL	This is the first step in loading and initializing the operating system into the machine.
Core dump	Dump	Copying data from memory to a display screen, printer, or file.
Crash, failure, hang	ABEND	Anytime a program terminates with anything other than a normal completion.
Directory (folder)	Catalog	File structure where datasets and files are stored on the system.

Domain	Sysplex (parallel)	A group of computers (or processors) administered as a single entity.
Files	Datasets	A way to group and store information and identify it by name (filename or dataset name).
Kernel	BCP	The core processes of an operating system.
Local/remote	Local/remote	This refers to whether the resources are on or directly connected to your own system (local) or if you must connect through other systems to access the resources (remote).
Multi-tasking, multi-processing	Parallel processing	This environment allows you to execute multiple processes at the same time, independent of each other.
Port	Channel	You can connect a device to this interface.
Process	Job (batch job)	This process allows you to run a program (a series of computer instructions).
Proxy server	Proxy server	A server that provides a shared cache of resources for the client machines at a site or across several sites.
RC	Condition code	A completion code that identifies whether a program has ended normally or abnormally.
Server	Server (eServer)	This computer or device manages resources on a network.
Service/daemon	Started task	This program or process runs in the background of an active session and doesn't interact with the user.
Shell	TSO/E	This user interface functions as a command interpreter to the BCP or kernel.
Shell script	CLIST	This file contains a series of commands to execute.
Thin client	3270 terminal, 3270 emulator, dumb terminal	This PC or terminal doesn't perform application processing. It functions as an input/output device only.
Thread	Task	This single unit of work within a program works independently from other units of work.
UNIX	USS	This multi-user, multi-tasking operating system contains a kernel, system utilities, and configuration files.
Virtual memory	Virtual storage	A way that provides programs with more memory than the computer physically has available.
Window	Split screen	This is a way to partition the viewable display area of a computer screen to enable different programs to run in different display areas.
WLM	WLM	Utilize system resources to help manage workload, workload throughput, and system utilization and responsiveness.
Workstation	Workstation	This single-user computer has a moderate amount of computing power and high-quality graphics.

Summary

We hope this overview of mainframe and distributed terminologies will peak your interest in learning more about the two environments and how they are coming together. Integrating these environments is a key part of IBM's On-Demand business and can help companies respond more quickly to customer needs, therefore becoming more competitive and flexible.

It is difficult to work in mainframe or distributed environments today without some overlap into the other. The common terms explained in this article are meant to bridge the gap for a novice in either environment.

Acknowledgements

The authors are most grateful to Hany Salem, Chris Ballam, Ron Allan, and Yu Tang for their helpful reviews and contributions.

Resources

- Use this dictionary and search engine for computer and Internet technology definitions.
- Use Wikipedia, an encyclopedia, to find more information.
- Refer to a computing dictionary.
- Visit the IBM home page for a complete listing of redbooks.
- Visit the Developer Bookstore for a comprehensive listing of technical books, including hundreds of eServer™ titles.
- Want more? The developerWorks eServer zone hosts hundreds of informative articles and introductory, intermediate, and advanced tutorials on the eServer brand.
- Get involved in the developerWorks community by participating in developerWorks blogs.
- The IBM developerWorks team hosts hundreds of technical briefings around the world which you can attend at no charge.

About the authors

Barbara Hampson is an Application and Integration Middleware Software Manager, WebSphere App Server FVT. Before going into management, Barbara was a WebSphere Application Server OEM Enhanced Support team lead. She has 23 years of experience at IBM, working in mainframe (VM, VSE, and MVS) for 13 years and 10 years in distributed systems across all platforms. In the past 10 years, she has worked extensively with SOM/DSOM, CBroker, and WebSphere Application Server customers. You can contact Barbara at hampson@us.ibm.com.

Alan Beaubien has 26 years of experience in the IT industry. Most of his career has been spent providing technical support in the mainframe world (MVS, OS/390, z/OS). Alan has spent two years providing WebSphere Application Server Level 2 support, before moving back to the mainframe environment as part of the WebSphere Serviceability Team's WebSphere Application Server z/OS SWAT team. At present, he supports WebSphere Application Server on distributed and z/OS platforms. He also serves as a team lead for the SWAT team's distributed support team. You can contact Alan at wbeaubie@us.ibm.com.