

The Cornerstone Systems **zFrame**:

An Overview



A small zSeries & S/390 compatible solution for all businesses

There are many reasons why a business of any size may require a relatively small zSeries or S/390 compatible computer system but there are few solutions for this need. Many IT groups in smaller companies can run all their mission-critical applications on a system with 120 MIPS or less, while some large companies may need disaster recovery systems, program development environments, or application server systems which are much smaller than their very large IBM zSeries production systems. The Cornerstone Systems Inc. **zFrame** family of systems can provide an economical and compact solution to these requirements. This paper will provide an introduction to the **zFrame** and some of its' unique capabilities. The first section will provide a general introduction and overview of the zFrame, the enabling software, *FLEX-ES* and what the combination is capable of. The second section will provide a more technical and detailed view of how the system works, followed by the final section that will describe some of the more advanced functions and ways to truly exploit the capabilities of the system.

zFrame, zDev and zPad are registered trademarks of Cornerstone Systems, Inc
FLEX-ES and FLEXCUB are registered trademarks of Fundamental Software, Inc.
IBM, zSeries, and S/390 are registered trademarks of the IBM Corporation.

The zFrame: What is it?

While a zFrame is made up of a number of components, which we will discuss in some detail, it is important to remember that, above all, a zFrame is a totally integrated, turn-key package that can be visualized as a standard IBM zSeries system. It is Cornerstone's intent to deliver every zFrame system such that it looks like an IBM technician installing a zSeries system. It is important to consider the zFrame as a complete packaged solution, not just as a collection of related parts. The basic hardware is always installed and configured to customer specifications by an experienced zFrame technician who will also provide customer training and integration into the target environment. Indeed, the entire zFrame concept was built upon this complete, integrated and packaged solution view.

In short, a zFrame is a small, but surprisingly powerful and capable server that will run zSeries and S/390 operating systems and applications just as a true IBM zSeries would run them. (We'll point out some differences, both good and bad, later.) A typical zFrame will provide about 60 MIPS of S/390 processor capacity, four Gigabytes of system memory (for processor memory and disk caching), 700 GB of internal disks and a combination of ESCON and Parallel channels. As we will see later, zFrames can be smaller or larger, but this is



somewhat typical. All of this is contained in a standard IBM eServer xSeries server that can

either be in a 'tower format' (see the picture to the left) or mounted in a standard 19" rack. The system includes a flat-panel system display and is powered by standard 110-120V (or other appropriate standard local voltage) electrical outlets. It is equally at home in either a computer center, or in an office environment. One of the key components of a zFrame is the software called FLEX-ES (from Fundamental Software, Inc.). We will cover FLEX-ES in more details later, but for now think of it as software that allows an Intel (or AMD) processor system to emulate and run the same software as a standard

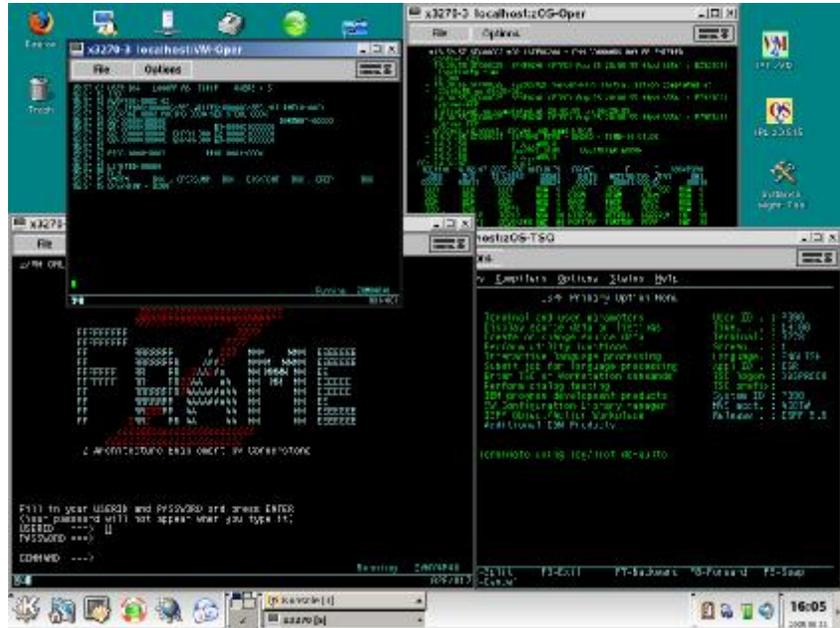
IBM zSeries system. It does this magic via the emulation process. It is important to understand that this is not a set of APIs (Application Programming Interfaces), but rather a very complete emulation of the entire zSeries architecture: instructions, memory, and I/O subsystem. All of the operating systems (such as zOS, zVM, zVSE, OS/390, VSE/ESA, Linux, etc.), all the "middleware" and subsystems (DB2, CICS, IMS, WebSphere, TSO, etc.) and applications run just as they would on a conventional IBM mainframe. We'll discuss emulation in more detail later. But first, what will it do??

Cornerstone Systems is an IBM Premier zSeries Business Partner. This means that CSI meets IBM's highest standards for selling and supporting zSeries hardware and software. Cornerstone is also the only zSeries IBM Business Partner that is authorized to sell FLEX-ES based systems. This gives CSI the unique ability to analyze a customer's requirements and recommend the most appropriate solution, whether IBM zSeries or Cornerstone zFrame. The strength of Cornerstone truly lies in the experience, knowledge, and capabilities of the technical support people for both zFrame and IBM zSeries based solutions. CSI offers multiple levels and types of support and services, from occasional 'on demand' services to ongoing and complete system support for a data center.

What will a zFrame do?

So, just what is a zFrame? Why should you (or anyone else) consider using one? In the simplest of terms, a zFrame will do about everything a traditional mainframe will do, and sometimes more, but in exchange for a lower maximum speed (about 120 MIPS), you get a more compact system, greatly reduced cost, and increased operational flexibility.

A zFrame will run any current IBM 31 bit operating system. (zFrames are capable of running 64 bit systems also, but present IBM licensing restrictions limits 64 bit availability to members of IBM's "PartnerWorld for Developers" or PWD.) zFrames can run multiple "instances" at one time, very similar to IBM's LPAR capability. As an example, here we see a zFrame system display where we have both zVM and zOS running. For zVM we see the OPERATOR user and a logo screen. For zOS we see the operator console and a user logged on and using TSO/ISPF. In this example zOS is **not** running as a guest of zVM, although it certainly could do so on a zFrame. You may recognize the background as SuSE Linux and KDE. That is the "host operating system" and we'll talk more about that later.



The zFrame exploits the technology to reduce costs and improve performance in a number of ways. For example, we utilize internal SCSI attached disks in a RAID-5 configuration to ensure data integrity, provide very high performance, and reduce costs. With up to 2.8 Terabytes of internal disk capacity, the zFrame can satisfy the disk requirements of almost all small to medium size system needs. Likewise, the zFrame uses standard PCI adapters for local area network connections. Standard multiport Gigabit Ethernet adapters provide high-speed interconnection at a low cost for either TCP/IP or SNA. (Token Ring adapters are also available.) If traditional ESCON or Parallel (Bus & Tag) channel attached devices, such as tapes or impact printers, are needed, channel adapters are available to connect them to a zFrame.

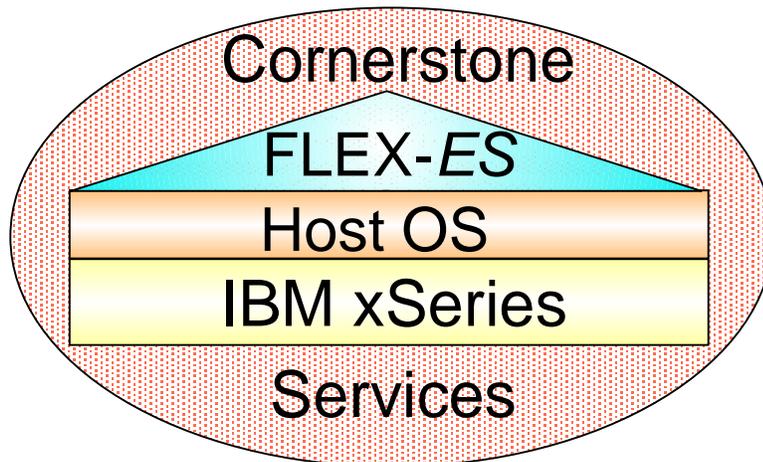
We will discuss the various capabilities and performance (sometimes called "speeds and feeds") later, but for now you can think of the zFrame as low cost, high flexibility, high availability, small sized, big performance, zSeries compatible "mainframe".

Components of a zFrame

Although zFrame users should think of a zFrame as a complete, packaged solution, it may be useful to look at the components to better understand how a zFrame works and how Cornerstone has taken advantage of the latest technology to deliver this highly functional system.

There are four primary components of a zFrame:

1. The IBM eServer xSeries server
2. The Host operating system that runs on the server
3. FLEX-ES emulator package
4. Cornerstone integration, configuration and services



IBM xSeries

Cornerstone pioneered the use of IBM servers for FLEX-ES based systems and now essentially all FLEX-ES systems run on IBM hardware. There are a number of reasons for this use of IBM equipment, but the two key ones are reliability and service. The IBM servers we use have a number of high-availability features, such as dual redundant, hot-pluggable power and cooling, ECC memory with multi-bit error correction, and built-in system management facilities. If there should be a problem, all zFrames include 7 x 24 on-site service by IBM, just like traditional IBM mainframe systems. IBM xSeries systems include features such as mainframe inspired “Enterprise X-Architecture” technology, Predictive Failure Analysis, and light-path diagnostics that provide improved availability and performance. Some models of zFrames that are specifically for program development make use of other IBM systems, such as AMD processor equipped eServers and IntelliStations to provide the highest possible performance. All server models and major subsystems are subjected to rigorous compatibility testing by Cornerstone Systems and a strict certification process by Fundamental Software before they are approved for use as zFrames. Only certified xSeries models and related components can be used for zFrames.

Host Operating System

Since FLEX-ES is essentially an application, it requires an operating system that runs on the underlying hardware. Cornerstone currently uses two different operating systems, depending on specific system requirements and customer preferences.

- SCO UnixWare (currently version 7.1.4)
- Linux (currently SuSE Pro 9.3)

Although UnixWare has been the primary host for many years, we are seeing a shift more toward Linux and expect this to continue in the future. In some cases the system configuration may dictate the host operating system, but usually the customer can choose the preferred system. The selected host operating system is installed and configured to very specific standards developed jointly between Cornerstone Systems and Fundamental Software. Cornerstone technicians include specific CSI developed tools, depending on the host system. For example, Linux based zFrames include REXX and a REXX based set of system management and backup/restore tools.

FLEX-ES

You might think of FLEX-ES as the “brains” of the system. This is the software that allows IBM zSeries and S390 systems and applications on Intel (or AMD) processors. FLEX-ES runs as an application on the host operating system and provides a number of functions, primarily:

- Instruction decode and emulation
- Storage management
- Device (I/O) emulation
- Resource management (across multiple system instances)
- Supporting utilities

Fundamental Software is the creator and provider of FLEX-ES. We will cover FLEX-ES in more detail later and you can find additional information at the FSI web site (www.funsoft.com) as well as in the reference material listed at the end of this document.

Cornerstone Services

The services and support provided by Cornerstone Systems acts as the glue or binding that brings all the pieces together to create a usable and productive system. This support starts during the initial proposal stage when CSI staff will analyze your requirements and suggest a configuration that best meets your needs. It continues through the system build and integration process, including a “System Assurance Review” and through our on-site installation activities. After the installation is complete, CSI support is available via a 7 x 24 hotline to provide problem resolution or, more likely, to provide system upgrades or assist with configuration changes. CSI service always includes assistance with the host operating system and the FLEX-ES related components, and it can even be expanded to include support for the IBM zSeries operating systems and software.

How are zFrames being used?

You may hear comments about zFrames and FLEX-ES being interesting, but only for development use. While program development is certainly a good use for zFrames, they are also being used for many other uses. In addition to development, zFrames are being used for:

- Mission-critical government and military use, including applications that have very high availability requirements, such as 911 phone systems and sensitive/secure military use
- Production use in banking and insurance with high-volume CICS transaction processing
- Application servers for companies that require continued access to historical data but the primary application has been migrated to another platform
- Disaster Recovery systems, as a backup for larger zSeries installations
- Bringing out-sourced applications back in-house, either for better control or cost savings

Is Emulation a good thing?

Some people tend to be concerned about using emulation-based technology, thinking that if it's not the "real thing", it can't be as good. While that may be true in some cases, *FLEX-ES* is somewhat unique, as we'll see.

History

Emulation has been used for a long time, in many forms. The very first S/360 instructions were executed on an IBM 7030 that was emulating the S/360 architecture. The tables were later turned and many S/360 systems were used to emulate 1401 and 70xx systems. In more recent times, the IBM P/390 and S/390 Integrated Server used micro-coded bit-slice hardware and complete I/O emulation. In many respects, the *FLEX-ES* emulation layer functions very much like "firmware" or "microcode" in many other systems today.

Passes the same IBM tests

IBM recommends and supports the use of *FLEX-ES* for the development of zSeries software by companies who will sell it to other companies. (This is the IBM PWD program mentioned earlier and the Cornerstone zDev product line is specifically for this audience.) To make sure *FLEX-ES* is suitable for this use, IBM and Fundamental Software subjected *FLEX-ES* to the same set of tests that IBM uses on new zSeries hardware. IBM and FSI verified that *FLEX-ES* complies with all the same architectural definitions and requirements as IBM zSeries hardware. *FLEX-ES* is the only zSeries emulator that has even been subjected to these tests, much less passed them.

Many of the features and characteristics of *FLEX-ES* have direct equivalents in zSeries hardware. The following table covers some of these:

IBM CMOS - zSeries	<i>FLEX-ES</i>
CP (Central Processor) and SAP (System Assist Processor)	Emulated Processor (CP) and Host Processor (SAP)
CP Instruction cache	Translated Instruction cache
IOCP Configuration	<i>FLEX-ES</i> Configuration File
Logical Partitions (LPARs)	Multiple Instances

As another form of endorsement of *FLEX-ES*, IBM itself is by far the largest single user of *FLEX-ES* in the world. To date, IBM has licensed over 500 copies of *FLEX-ES* for its own internal use: they know a good thing when they see it!

Current Models

We'll first take a quick look at the zFrame and zDev models and their “speeds and feeds”, then we'll look a little closer at what the various number mean.

zFrame Models

Feature Model	MIPS (approx) ⁽¹⁾	Std. Avail. Memory ⁽²⁾	Std. Avail Disk ⁽³⁾	#Processors ⁽⁴⁾	xSeries base model ⁽⁵⁾
Z8	8 (throttled)	3.4 GB	330 GB	2 / 1	232 2 x 1.4 GHz
Z18	14-18	3.4 GB	330 GB	2 / 1	232 2 x 1.0 GHz
Z30	30	3.4 GB	330 GB	2 / 1	236 2 x 3.2 GHz
Z60	60	3.4 GB	330 GB	2 / 2	236 2 x 3.2 GHz
Z90	110	6.0 GB	330 GB	4 / 3	260 4 x 3.6 GHz

Footnotes to the above table:

1. MIPS historically means “Millions of Instructions Per Second”, or perhaps more accurately, “Meaningless Indicator of Performance”. In any case, it is a frequently used yardstick for comparing mainframe processor performance. You should always consider overall total system performance and not just raw processor speed.
2. Standard Memory is the base memory size for that model. This is memory actually available as S/390 – zSeries memory and disk cache memory, after the memory for Linux/UnixWare and FLEX-ES has been allocated. The z30, z60, and z90 can be configured with significant additional memory, up to about 12 GB on the z90.
3. Standard Available disk is the number of Gigabytes of disk space available on the standard configuration. In all cases significant additional disk space can easily be added either internally or in an external disk unit. The disk capacity is always quoted as space actually available to the S/390 or zSeries environment. For example, a z30 with 330 GB could emulate about 118 volumes of 3390-3. Or a smaller amount can be allocated as disk and remaining space used for “FakeTape” files (to be discussed later).
4. Under “Processors”, the first number is the number of processors installed in the xSeries machine and the 2nd number is the number of processors that can be used for instruction emulation.
5. The xSeries base model column indicates the model of IBM xSeries system used and the number and speed of Intel Xeon processors installed.

zDev Models

Feature Model	MIPS (approx) ⁽¹⁾	Std. Avail. Memory ⁽²⁾	Std. Avail Disk ⁽³⁾	# processors ⁽⁴⁾	xSeries base model ⁽⁵⁾
zPad ⁽⁶⁾	38 (I/O limited)	1.25 GB	60 GB	1 / 1	ThinkPad T43
zDev1	36	3.4 GB	160 GB	2 / 1	236 2 x 3.6GHz
zDev2	65	3.4 GB	330 GB	2 / 2	236 2 x 3.6GHz
zDev3	110	3.4 GB	330 GB	4 / 3	260 4 x 3.6 GHz
zDevY ⁽⁷⁾	130	6.0 GB	420 GB	2 / 2	IntelliStation 2 x 2.8 GHz

Footnotes to the above table:

In general, the footnotes described above apply here also, with the additional following notes:

1. MIPS: Be especially careful of the MIPS for the zPad as it has very limited I/O performance.
2. Standard Memory: all zDev systems can be increased to larger memory size. The zPad is most limited with a maximum of about 1.6 GB of available memory.
3. The disk space of all systems can be increased, but the zPad will generally be restricted to about 150 GB maximum, using a second disk unit.
4. Under “Processors”, the first number is the number of processors installed in the system and the 2nd number is the number of processors that can be used for instruction emulation.
5. The xSeries base model column indicates the model of IBM xSeries system used and the number and speed of Intel processors installed.
6. The standard zPad uses a ThinkPad T43 with a 2.16 GHz Pentium M processor and a single 80 GB disk (high performance/capacity model is also available)
7. The zDevY uses an IBM IntelliStation (6217) with two AMD Opteron 2.6 GHz processors. The Opteron processors provide significantly higher emulation performance “per GHz” than the Intel Pentium Xeon processors in the xSeries.

Other PWD information

- zDevs are available only to companies who are active participants in IBM’s PWD program
- For developers of software to be sold to zSeries customers
- IBM Software available “on loan” (zOS, zVM, zVSE and related components)
- IBM and FSI provide special pricing on FLEX-ES for PWD members

Software Licensing

For most “mainframe” users, software license cost is the largest piece of the system expense. zFrame systems can often help in this area also, by utilizing lower cost license structures.

IBM Software

For example, IBM software for the commercial zFrame systems are generally licensed as described in the following table:

zFrame model / MIPS	License base	Comments
z8 / 8	MLC/Group⁽¹⁾	IBM has withdrawn the formerly available ESL licensing
Z18 / 16	1/3 GOLC	Attractive pricing. Limited to Pentium III 1.0 GHz processors
Z30 / 30	2/3 GOLC	Fastest Uni-Processor Less cost than 36 MIPS z800
z60 / 60	GOLC H30	Same license as MP3000 H30
Z90 / 110	GOLC H50	Same license as MP3000 H50

Notes:

- (1) IBM has withdrawn the ESL license effective Jan. 1, 2006 (ESL was introduced by IBM to support the P/390 product line and was a very attractive one time charge price for systems of 8 MIPS or under.) The zFrame Model 8 is still available, but licensing must be via traditional Group based MLC.
- (2) (General/all) Each IBM software license for a zFrame is applied for uniquely as a “Special Bid request”. Because of this process, IBM can change the license requirements or characteristics at any time, without warning.

Non-IBM Software

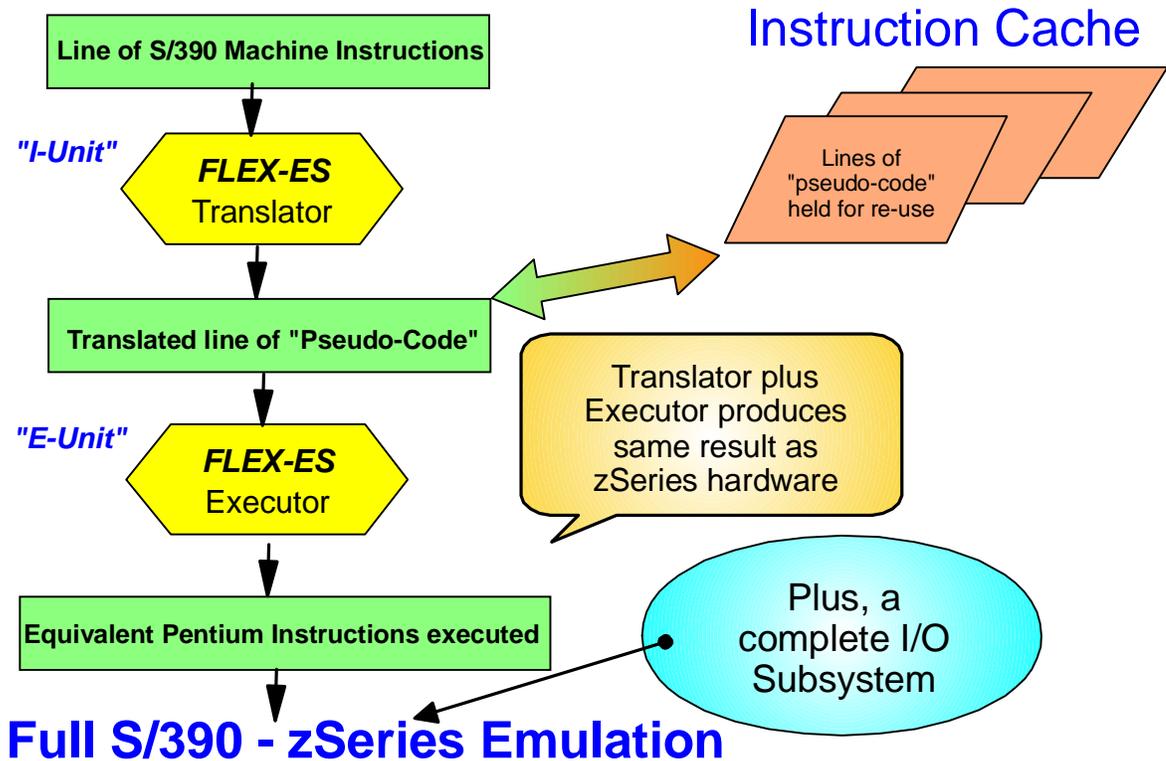
Other software vendors frequently follow the example of IBM when pricing software for FLEX-ES systems, but not always. Those who do not follow IBM may or may not offer special software pricing for these systems. CA, for example, does not follow IBM’s model, but does have a unique pricing structure, at a lower cost, for many of their products when running on FLEX-ES systems. Users should check with their software vendors for specific pricing. If you have questions or problems getting software licenses, check with Cornerstone; we have contacts at many of the software vendors and may be able to help. It is worth noting that most software vendors do have their own FLEX-ES systems (PWD zDev development systems) although the marketing or administrative groups may not be aware of it.

What makes a zFrame Work?

First, a gentle warning.... This section gets a bit more technical in describing just how zFrames and FLEX-ES work. Readers who only want a general overview or introduction may want to skip this section and go directly to “The Cornerstone zCenter” on page 18.

FLEX-ES at the detail level

First, let’s take a look at how FLEX-ES works: (Fundamental Software does not provide any detailed descriptions of how FLEX-ES works. The following diagrams and descriptions are somewhat simplified and based purely on my understanding and use my terminology.)

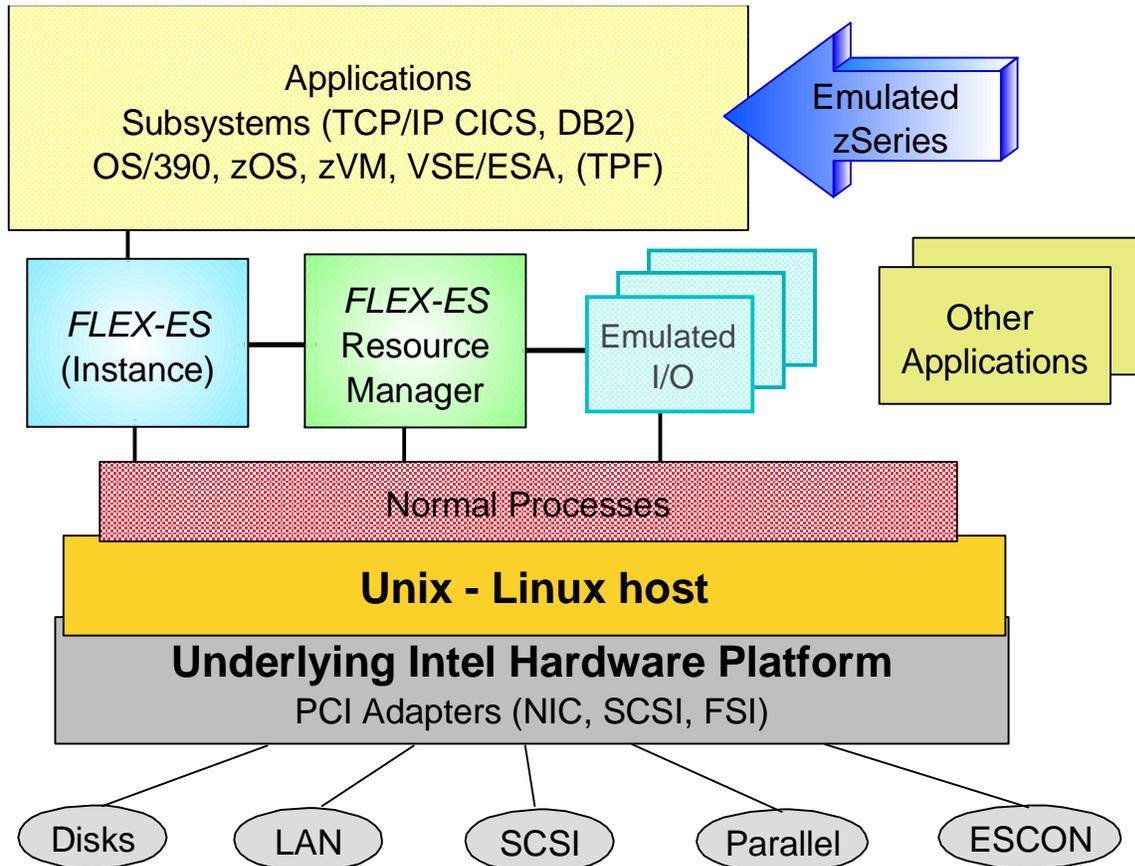


When FLEX-ES encounters a new section of zSeries code (referred to as a “line”) it processes it in the FLEX-ES translator, which compiles it into a line of “pseudo-code”. This pseudo-code is very close to native Intel x86 machine instructions, so it can execute very quickly. A copy of this pseudo-code is also saved in the translated instruction cache for possible later use. (The compilation process is time consuming, so caching the pseudo-code provides significant performance benefits. When analyzing and tuning zFrames, we look for about 97% cache hit ratios for the instruction cache.) The FLEX-ES executor will actually execute the pseudo-code, either the just compiled copy, or one from the cache. This execution will result in the exact same result as if the code had been executed on a zSeries system (storage locations, registers, condition codes, PSW, etc.)

FLEX-ES also includes a complete I/O subsystem emulation that is tied into the instruction processor. The I/O subsystem operates very much like a zSeries I/O subsystem, including channels and control units. The combination of the instruction processor, I/O subsystem and storage management routines will provide the full S/390 or zSeries emulation.

Readers who are familiar with somewhat traditional computer architecture may recognize the similarity of the FLEX-ES translator to the “I-Unit” (Instruction Unit) and the similarity of the Executor to the “E-Unit” (Execution Unit). As technicians look closer at how FLEX-ES works, they see many parallels to traditional processor and system operation.

A system-Level view of a zFrame

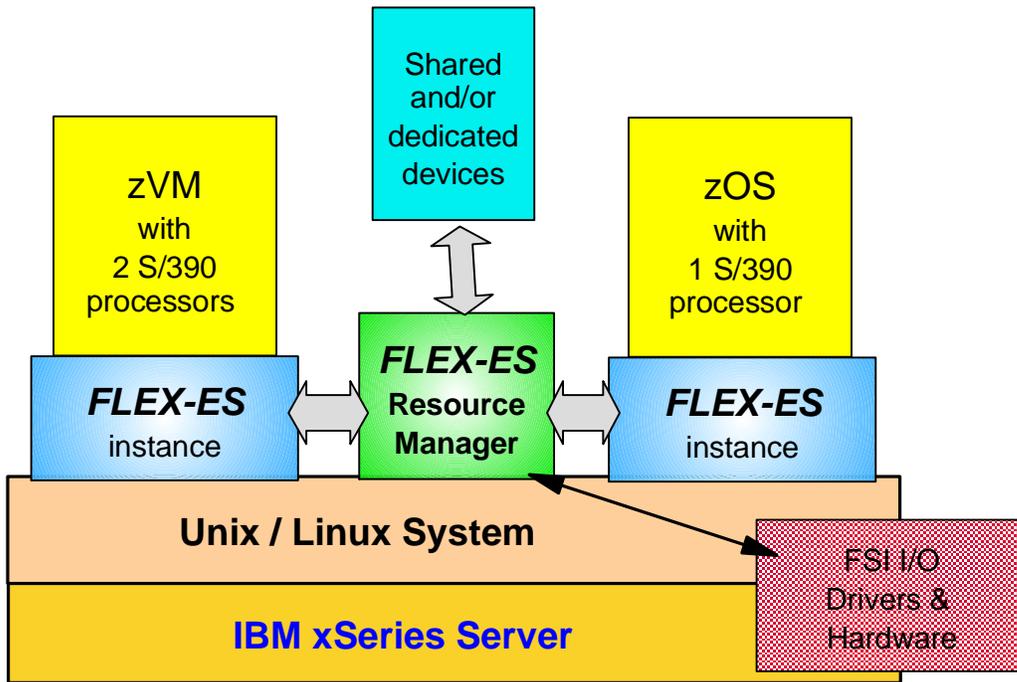


In this diagram of a FLEX-ES system, we want to start at the bottom and work up. At the lowest level we have the physical I/O devices, such as disk drives, tape drives, network connections, printers, and so forth. These can be connected via SCSI channels, traditional Parallel (Bus & Tag) channels, ESCON channels, or plugged directly into the system bus. These are all connected to, or plugged into, the IBM Intel (or AMD) hardware system, typically an xSeries server. Cornerstone System technicians assemble the hardware components, do basic system setup (including RAID configuration) and then install Linux or UnixWare with the appropriate drivers and supporting programs. FLEX-ES is then installed and configured on top of the Linux/UnixWare layer.

FLEX-ES runs as a normal Linux process with some hardware device drivers for the FSI provided channel adapters. A FLEX-ES “instance” provides instruction emulation and memory management for a single S/390 or zSeries system. The FLEX-ES Resource Manager coordinates the action of the emulated I/O devices and makes sure all instances get the proper resources (processor time, memory, I/O operations). Once FLEX-ES is initialized (the instance), normal

system functions, such as system IPL, starting and stopping the processor, and many other FLEX-ES commands can be used. The above diagram, for example, suggests that an operating system, perhaps zOS, has been IPL'ed and is running in the instance. zOS may have access to internal emulated disks, SCSI tapes, and any device which can be attached via Parallel or ESCON channels. The drawing above shows a FLEX-ES system running a single instance: let's look at an example of a system running multiple instances.

A Multiple Instance Example



In this view we again see the IBM xSeries server with UnixWare or Linux running as the host operating system. This time we show the Resource Manager coordinating access to the I/O devices and communicating with the two FLEX-ES instances. This system is running zVM and zOS systems, each in it's own instance. There are several means of communicating between instances; including using shared DASD, emulated channel-to-channel devices, or normal network connections. This example points out that the instances may have much different characteristics, such as the number of emulated processors available to it. The instances could even run in much different modes: one instance could be z-Architecture (64 bit) and the other one could be S/370 mode. There are some limitations (for example, no instance can have more emulated processors than actual 'enabled' processors on the system), but the functional capabilities are very powerful. The number of instances that can be started at one time is generally limited only by the memory available and the ability of operators to keep track of and manage many instances on one system.

System Components

Let's take a look at some of the common system components and see how each is provided by a zFrame. We'll discuss any special capabilities or limitations as we cover each component.

Processors

Processors could be considered the heart of a system, so we will start there. As part of the software license for each copy of FLEX-ES, there is a limit as to the number of Intel architecture processors that are 'enabled' to run FLEX-ES instruction emulation. A zFrame model z30, for example, only has one enabled processor, while the z60 has two enabled processors. Since the z60 can run two processors, it is about twice as fast as the z30. An attempt to define a dual processor instance on a z30 would fail. In some cases the license also restricts the maximum speed of the processor.

Related Instance Attributes

There are a number of attributes that can be assigned to an instance which are not necessarily processor specific, but they generally make sense to discuss at this time, as they are very related to the processor.

Mode: As mentioned previously, each instance is defined as a specific architecture mode, such as S/370, ESA(390), or Z (64 bit).

LPAR: Instances can be defined as like a LPAR so that operating systems in that instance think they are running within an LPAR.. This has several possible effects. For example, if VM detects that it is running in an LPAR environment it will not utilize it's "active wait" dispatcher, conserving processor cycles for other instances.

LPAR Name/Number: If the LPAR feature is specified, then values should also be specified for the LPAR name and LPAR number. The LPAR name is frequently used on zOS systems as part of the PARMLIB, allowing multiple systems to share and use the same SYSRES and PARMLIB.

Memory

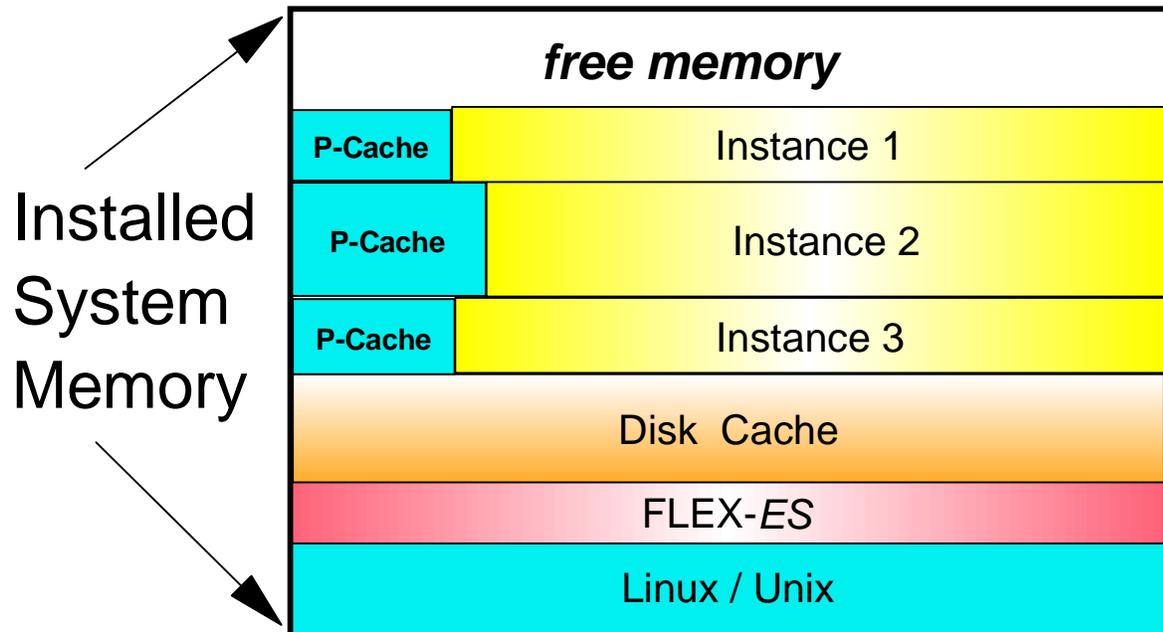
There are four main uses of memory in a zFrame system:

- Linux/Unix operating system (typically 256 – 512 MB in normal configurations)
- FLEX-ES code and control blocks
- FLEX-ES controlled disk cache
- Instance Memory: Processor cache and emulated zSeries memory for each instance
 - Processor cache
 - Emulated zSeries memory

It is worth mentioning here that, in a zFrame system, we do not want the host OS (Linux/Unix) to page or swap at all. If the host OS does page or swap, it will slow down the entire system so one design/configuration point for all zFrames is to have enough memory to eliminate any host OS paging.

It may be helpful to view this memory use as a diagram:

Memory use in a zFrame



Linux-Unix: The host operating system requires about 256 – 512 MB in typical zFrame configurations. While Linux or Unix will certainly run in much smaller memory sizes, we want to make sure that the operating system does not have to page itself at all.

FLEX-ES: The FLEX-ES code and control blocks are rather small, certainly no more than 64 MB for any reasonable configuration.

Disk Cache: FLEX-ES maintains a tunable disk cache for the emulated zSeries disks. The size of this cache will depend on the size of the disk configuration and how well it is to be cached. A reasonable rule of thumb is to provide about 30 tracks of cache for each 3390-3 equivalent, or about 1.5 MB of memory per 3390-3. For a system with 700 GB of emulated disks, this would be about 375 MB of memory for disk caching. If additional memory is available, the disk cache size can always be expanded. The disk cache is effectively allocated at zFrame system initialization.

Instance Memory: Instance memory consists of the combination of processor cache and the emulated central storage. Because of restrictions in the 32 bit versions of Linux or Unix, this will generally be a maximum of about 1.7 or 1.8 GB per instance. FSI has disclosed that they have developed a new version of FLEX-ES which will run on a 64 bit base Linux system. This will be called FLEX-ES Version 8 but no availability date has been announced. FLEX-ES Version 8 fully supports and runs on 64 bit Linux so these memory limitation will be essentially eliminated when it is used. 31 bit S/390 systems will be able to get a full 2GB of central memory, and 64 bit zSeries systems will be able to get “greater than 2 GB” central memory.

Emulated Disk Devices

Emulated disks are a somewhat special type of I/O device, so we will discuss them separately. zFrame emulated disks are always configured in a RAID-5 (redundant with distributed check-sum data) configuration and using IBM ServeRAID adapters. This ensures very high

performance and full redundancy. The physical disk drives are typically 10,000 RPM SCSI-320 disks, although 15,000 RPM disks are also available. FLEX-ES uses “Raw I/O” to the emulated disks to provide the very high performance levels, so Cornerstone technicians configure the disks using either UnixWare raw disk slices or Linux Logical Volume Manager volumes mapped to raw devices. The slices or volumes are named in such a way as to be familiar to traditional mainframe users.

Some prospective customers want to connect to existing large disk arrays, such as IBM 2105 (“Shark”) units, but we suggest using internal (xSeries) disks whenever possible because the performance is much better and the internal disks cost less than just the channels to the external units would cost. The disk cache (discussed above) is very effective and, because it is internal to the system, cache hits do not even require moving data across a channel. Measured disk I/O response times for read cache hits are in the 20 – 30 *microsecond* range; orders of magnitude better than a traditional channel connected external disk/cache unit.

Standard “mid-size” zFrames (30 – 60 MIPS) can easily provide 1.4 TB internally and several TB more in separate SCSI enclosures. Larger systems, such as the 110 MIPS model z90 can provide 2.8 TB internally.

Other I/O Devices

There are several other classes or types of I/O devices available on a zFrame, so we’ll look at several of them. This will by no means be an exhaustive list.

LAN/Network

ZFrames can connect to almost any local area network via either Ethernet or Token Ring adapters (NICs). These are standard PCI bus adapters and are available in several types including several with 1 GB (gigabit) speeds. These adapters can be configured as either 3172 (LCS) or OSA devices with either SNA or IP protocols.

Channel attached devices

ESCON and Parallel (“Bus and Tag”) channel adapters are available for zFrame systems. These allow the connection of almost any traditional channel attached device such as tapes (3490s), printers, and communications controllers (3745, 3174). The number of channels is generally limited by the number of available PCI slots in the system, but since the internal disks do not require channels, and emulated devices replace many “real” devices, relatively few channels are typically needed.

Tapes

Tape drives can either be traditional channel attached (3490, 3590, etc.) or SCSI attached tapes. Channel attached tapes behave and function just as on a traditional mainframe system. SCSI tapes can either be of traditional 3480/3490 media compatible types (which appear to the zSeries system just as standard 3480/3490 drives) or newer technology such as SDLT or LTO. These new devices can be defined to the zSeries system to look like traditional 3480/3490, but with very large capacities (300 – 400 GB per cartridge) and high performance (20 MB/Sec). A minimum of one SCSI tape drive of some type is required to provide disaster backup/recovery of the Unix/Linux environment. Since this is typically 10 – 12 GB, a media compatible 3490 type device will not suffice: a SDLT, LTO or DDS type tape drive must be available. Using SDLT or LTO technology, many medium sized installations can backup all of their data to a single physical cartridge.

Exploiting the Technology

So far in this paper, we have mainly discussed making the zFrame behave as much as possible like a traditional “mainframe”. In this section we will look at how we can take advantage of the zFrame’s capabilities and exploit the FLEX-ES technology to enhance the effective functionality of the system.

Network printers

Using FLEX-ES provided capabilities, zSeries systems can print to what appears to be standard system printers (4245, etc.) and have it redirected to network attached printers. While this does not have the degree of functionality as products as IP PrintWay on zOS, it is a quick and simple solution for basic printing requirements.

Channel to Channel connections

Emulated channels can be configured between multiple FLEX-ES instances on a single zFrame, or between instances on two different physical zFrames. These Network Channels between two systems can be transparent to the zSeries systems, so that the operating system on one zFrame can access a device, such as a SCSI tape drive, on another zFrame, as though it was actually locally attached.

Shared DASD

DASD can be shared between instances, whether all on one zFrame or between instances on different zFrames. Sharing between physical zFrames is done using Network Channels. Reserve – Release functions are honored and we always recommend care in configuring shared DASD systems to ensure good performance with data integrity.

“FakeTape”

FakeTape is a FLEX-ES provided emulation of tape devices using files on disk. There are a number of ways to exploit this capability.

Fast local Backups

Regular disk backup “tapes” can be written to FakeTape rather than to real tape media. Then, when convenient, the FakeTape files can be copied to tape media (such as SDLT) for physical backup. Since FakeTape operates at disk speeds, this backup is faster than writing to real tape, so the backup “window” does not have to be as long. Because FakeTape files are actually just disk files, no operator intervention (loading tape cartridges) is required, allowing unattended operation.

Backup directly to remote system

A variation of this that is in use by several zFrame customers is to transmit the FakeTape files directly to a remote location (where a recovery system may be located) without ever writing the files to real tape. This can be done in several ways:

- After backing up to FakeTape files on the local system (as above), those files can be FTP’ed via batch FTP to a remote system.
- The FakeTape files can be written to a filesystem that physically exists on a remote system and is NFS (network file system) mounted on the zFrame. Thus, as the zSeries

system is writing the backup “tapes” they are actually being immediately written to the remote (recovery) system.

AFLIB provides VTS function

Fundamental Software offers an optional software package that integrates FakeTape functions into zOS’s DFSMS-HSM and RMM so as to provide a rather complete Virtual Tape System on FakeTape. These FakeTape files can physically exist on the zFrame’s internal disks, on a separate set of external disks, or on a network connected (NFS) file system. This allows a very high performance and highly responsive VTS at a very attractive price point.

Although FSI does not provide equivalent packages for VSE, several other software vendors do offer FakeTape automation products and functions for VSE. For example, both BIM-EPIC and CA-Dynam/T recognize and exploit FakeTape capabilities on VSE.

For zVM customers who want to automate FakeTape mounting, FSI does offer “fsihost” which allows VM/CMS users to issue commands to the underlying Linux/Unix systems and to the FLEX-ES command line interface in order to ‘mount’ new FakeTape files.

Several zFrame customers have developed their own FakeTape “automation” systems and these are occasionally discussed on the FLEX-ES List Server (see references for information).

FakeTape Compression

FLEX-ES Ver 7 provides the option of compressing the FakeTape files as they are being written. The compression is done at the Linux/Unix level, not by the zSeries processor, so it has little or no impact on zSeries processor performance. Most zSeries type data will experience 3 or 4 to 1 compression factors. This makes some of the other FakeTape capabilities even more attractive. Think of the VTS function of AFLIB, operating across an NFS Gigabit connection to a NFS server machine where the data resides in compressed form. The effective transfer rates would be much higher because of the compression and the FakeTape files would take up less space on the NFS server’s disks.

FakeTape Encryption

In response to the increasing need for data protection, FSI has announced the availability of FLEX-ES version 7.0.6 which includes optional encryption of FakeTape data. Encryption is via OpenSSL128, 192 or 256 and it includes comprehensive key handling features.

Cornerstone zCenter

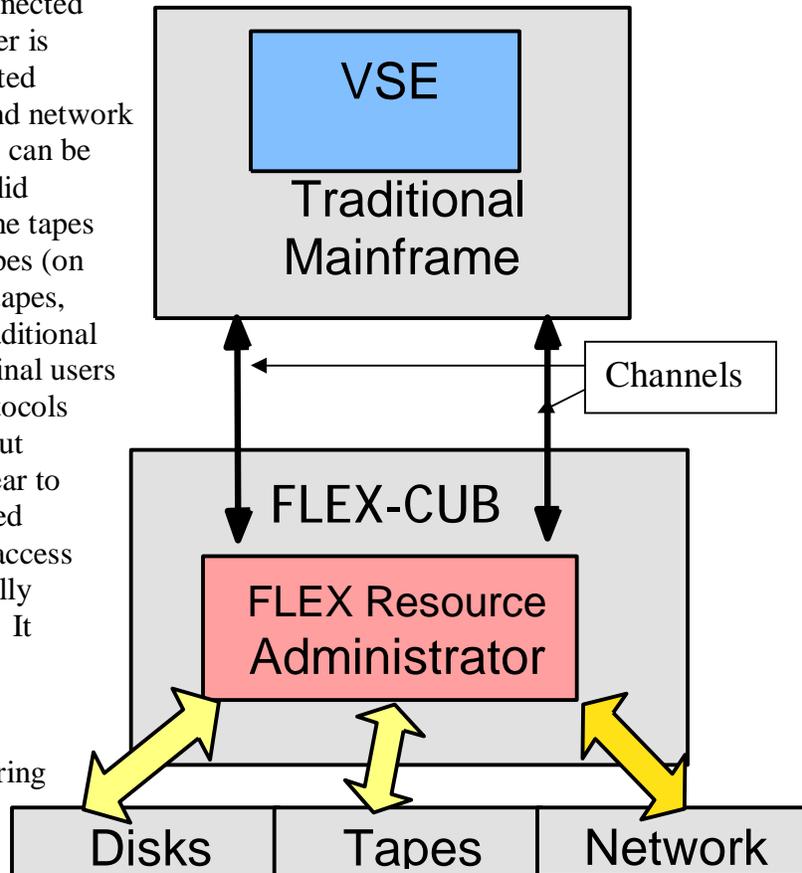
The Cornerstone zCenter is built on the zFrame technology base and uses the Fundamental Software “FLEX-CUB” (Control Unit Behavior) software package. It is a multi-function I/O device emulation subsystem that can provide disk storage, emulated tapes (SCSI or FakeTapes) and IP Network interconnection to traditional mainframes or zFrames across a channel (ESCON, Parallel, or FLEX Network channel).

A picture might look like this:

VSE is running on a traditional “mainframe”, and is channel connected to a zCenter system. This zCenter is providing emulated disks, emulated tapes (FakeTape), SCSI tapes, and network attachment. The emulated disks can be defined as 3390s or any other valid VSE recognized DASD type. The tapes can be a combination of FakeTapes (on disk) or possibly SCSI attached tapes, such as SDLTs, that look like traditional 3490s to the VSE system. Terminal users can access the system via IP protocols across the Network connection but the terminal sessions would appear to VSE as non-SNA, locally attached 3270 devices. This provides IP access to the VSE system without actually running a TCP/IP stack on VSE. It can also function much like an IBM 2074 display controller. There could be multiple LPARS connected to the zCenter, all sharing the resources it provides. In general, the disk emulation component can perform the same function of external disk arrays

such as an IBM 2105 Enterprise Systems Storage (Shark) unit. However, the zCenter will typically not have sufficient channels to provide the data bandwidth needed for high performance disk subsystems. The zCenter is more likely to be used as a backup device, or for large databases with very low access density. The AFLIB product can run on zOS on the traditional mainframe, accessing compressed FakeTape files in the zCenter, providing a very complete and flexible VTS capability.

In general, it is not any one function that makes the zCenter an outstanding concept, but the ability to combine these functions in one compact unit that offers some very interesting capabilities.



Summary and References

The Cornerstone zFrame provides an exciting alternative to traditional mainframe computers, while still preserving and protecting the investment in S/390 and zSeries software and subsystems. zFrames offer a viable alternative in the “under 120 MIPS” marketplace while offering IBM hardware service/support and the services and support of Cornerstone Systems, one of the largest zSeries IBM Business Partners, and the only Premier IBM zSeries Business Partner that can offer the zFrame family of products: zFrame, zDev, zPad, and zCenter.

For additional information, check the Cornerstone Systems Web site (www.csihome.com) and follow the zFrame link. To learn more about FLEX-ES and zFrames in general, check the following references and publications.

IBM Redbooks

Most of the IBM Redbooks on the subject are oriented toward developers who are using FLEX for development work, such as on a zDev, but most of the information is applicable to commercial system use also.

SG24-6215: “NUMA-Q Enabled for S/390: Technical Introduction”

Still the best overall introduction to FLEX-ES (Skip over NUMA-Q (xSeries 430) specific sections.)

SG24-6501: “S/390 PWD Netfinity enabled for S/390”

SG24-6507: “S/390 PWD ThinkPad Enabled for S/390”

SG24-6834: “S/390 Partners in Development: EFS Systems on a Linux Base”

SG24-7007: “EFS Systems on a Linux Base: Getting Started”

SG24-7008: “EFS Systems on a Linux Base: Additional Topics”

Excellent for users who want to get a little deeper ,not just for Linux based systems
These IBM Redbooks are available at: <http://www.redbooks.ibm.com>

Whitepaper

“Exploring the I/O Performance Characteristics of Intel Based FLEX-ES Servers for z/OS” by Dr. H. Pat Artis, Performance Associates, Inc.:

http://www.perfassoc.com/flex-es_io_performance_02.pdf

This was on an old server: recent tests on new models show approx 2x this performance.

Presentations

“Implementing Virtual Tape in a little Itty-Bitty 64 Bit World” This presentation given at SHARE (Feb. 2005) by Dr. Pat Artis of Performance Associates discusses his experiences in testing and benchmarking an early version of Fundamental Software's AFLIB. The focus is on using a remote server, connected via NFS, as the VTS "server" but general concepts are also covered.

[www.csihome.com/wwwfiles/newwel.nsf/ImageLibraryLookup/SHARE2005/\\$FILE/AFLIBWorld.pdf?OpenElement](http://www.csihome.com/wwwfiles/newwel.nsf/ImageLibraryLookup/SHARE2005/$FILE/AFLIBWorld.pdf?OpenElement)

Support Listserv (FLEX-ES)

www.listserv.uga.edu to subscribe

Web pages

www.csihome.com (and take the zFrame link)
www.funsoft.com (Fundamental Software Inc)