Linux on IBM System z

# Best Practices for WebSphere Application Server on System z Linux

Steve Wehr
System z New Technology Center
Poughkeepsie

An introduction to setting up an infrastructure that will allow WebSphere applications to run efficiently on Linux for System z. This infrastructure consists of LPARs running VM, running multiple Linux guests, each running WebSphere, running your applications. That's a lot of layers, where everything has to work together well. This presentation tells you how to start setting up such an architecture, how to make these parts work together optimally, and how to allocate memory between all the systems involved.

# Trademarks

**The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.**

e-business logo
IBM*
IBM eServer
IBM logo*
VM/ESA*
WebSphere
z/OS*
z/VM*
zSeries*

* Registered trademarks of IBM Corporation

**The following are trademarks or registered trademarks of other companies.**

Intel is a registered trademark of the Intel Corporation in the United States, other countries or both.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Penguin (Tux) compliments of Larry Ewing.
Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.
* All other products may be trademarks or registered trademarks of their respective companies.

**Notes**:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

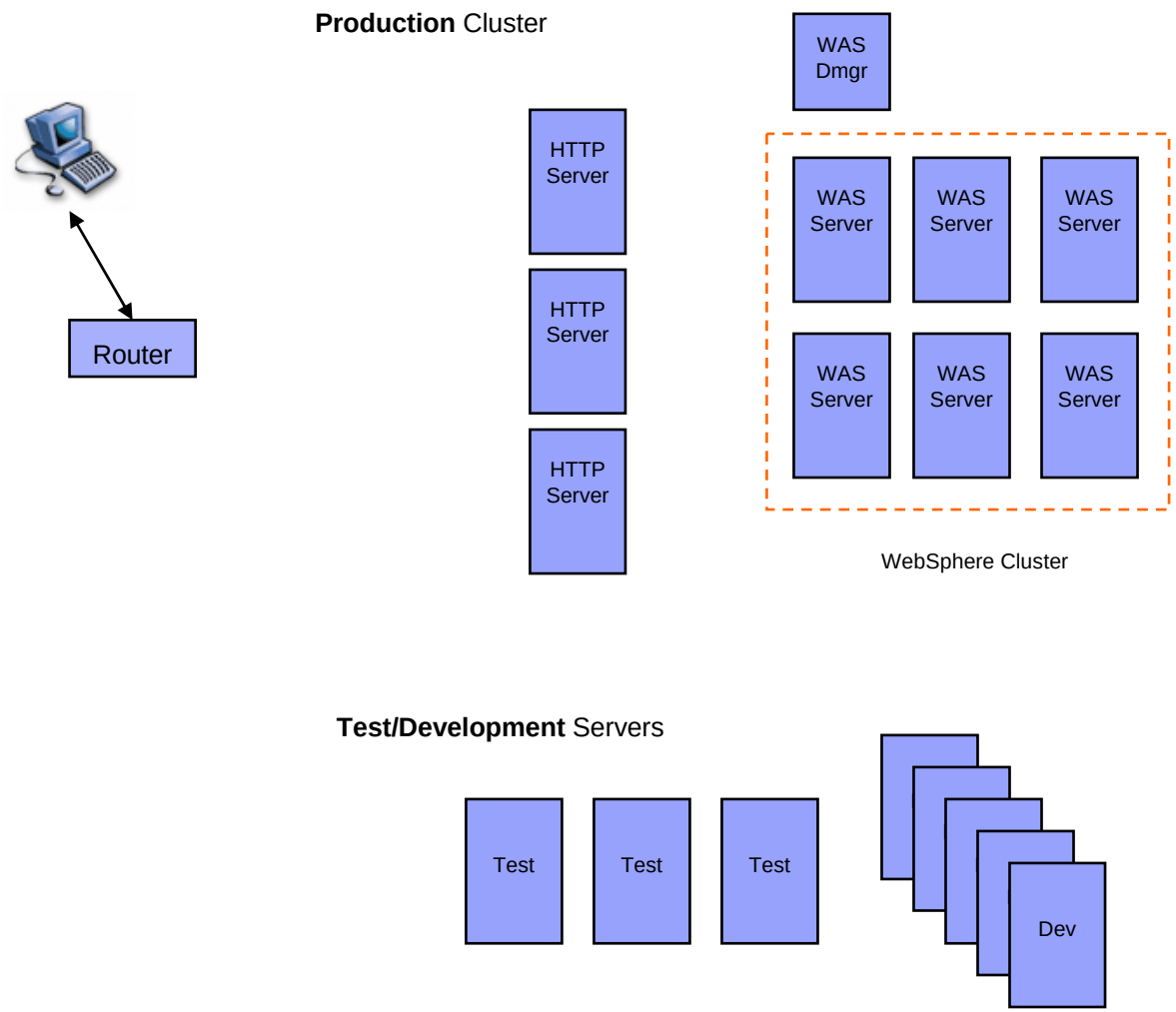
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products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

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

# Agenda

- Recommended System Architecture (Infrastructure)
- Setting up:
  - LPARs
  - VM/ESA®
  - Linux
  - WebSphere®
- Memory Sizing and Overcommitting Memory
- Deployment patterns
- Sharing binaries
- Monitoring performance
- Summary.

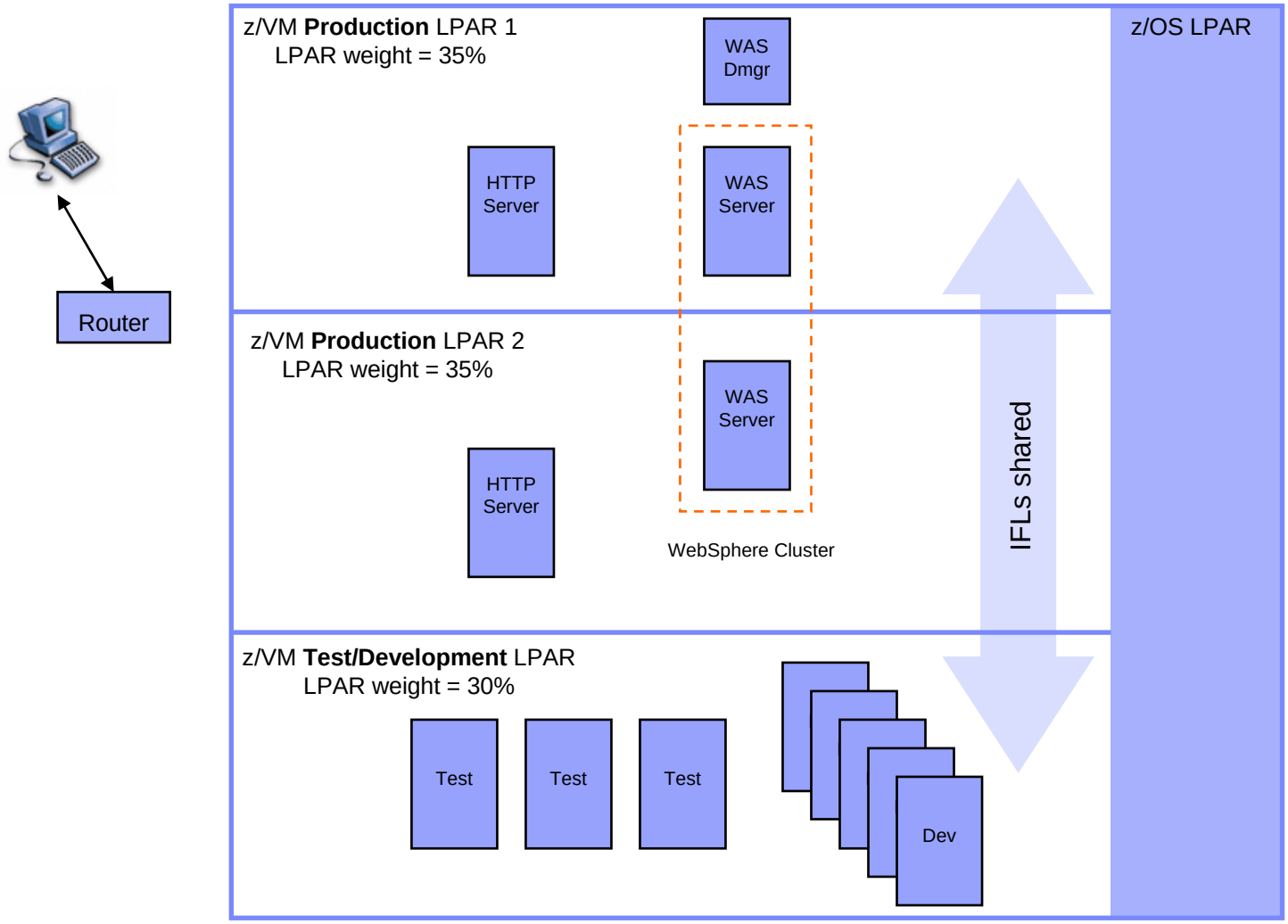
# Typical Problems with Physical Servers




Each blue box is a standalone server.

- Production Cluster sized for peak usage. Under-utilized most of the time.
- Cluster includes extra servers so that if one fails, others have adequate resources to handle workload.
- Dev/Test servers greatly under-utilized.
- Long lead time to provision a new server.
- High Software costs.
- High power and space costs.

# Recommended Virtualization Solution on z




Each blue box is a virtual Linux server.

- All Linux virtual servers draw from a common pool of memory and IFLs.
- Resources from a failed server flow to surviving servers
- Small application clusters (Just enough nodes for failover)
- Smaller cluster reduces failure points
- Two LPARs run production workload.
- Applications run in clusters split between the prod LPARs.

# Setting up the LPARs

- Ensure that Production guests are not impacted by Test/Dev.
  - Define 2 LPARs:
    - Production: 70% weight, no cap.
    - Dev/Test: 30% weight, no cap.
  - or Run all guests in one LPAR. Use VM SHARE to prioritize guests.
    - Production: SET SHARE 400 REL LIMITSOFT
    - Test: SET SHARE 200 REL LIMITSOFT
    - Dev: SET SHARE 100 REL LIMITSOFT
- Share all the IFLs between all the Linux LPARs.

# Setting up VM

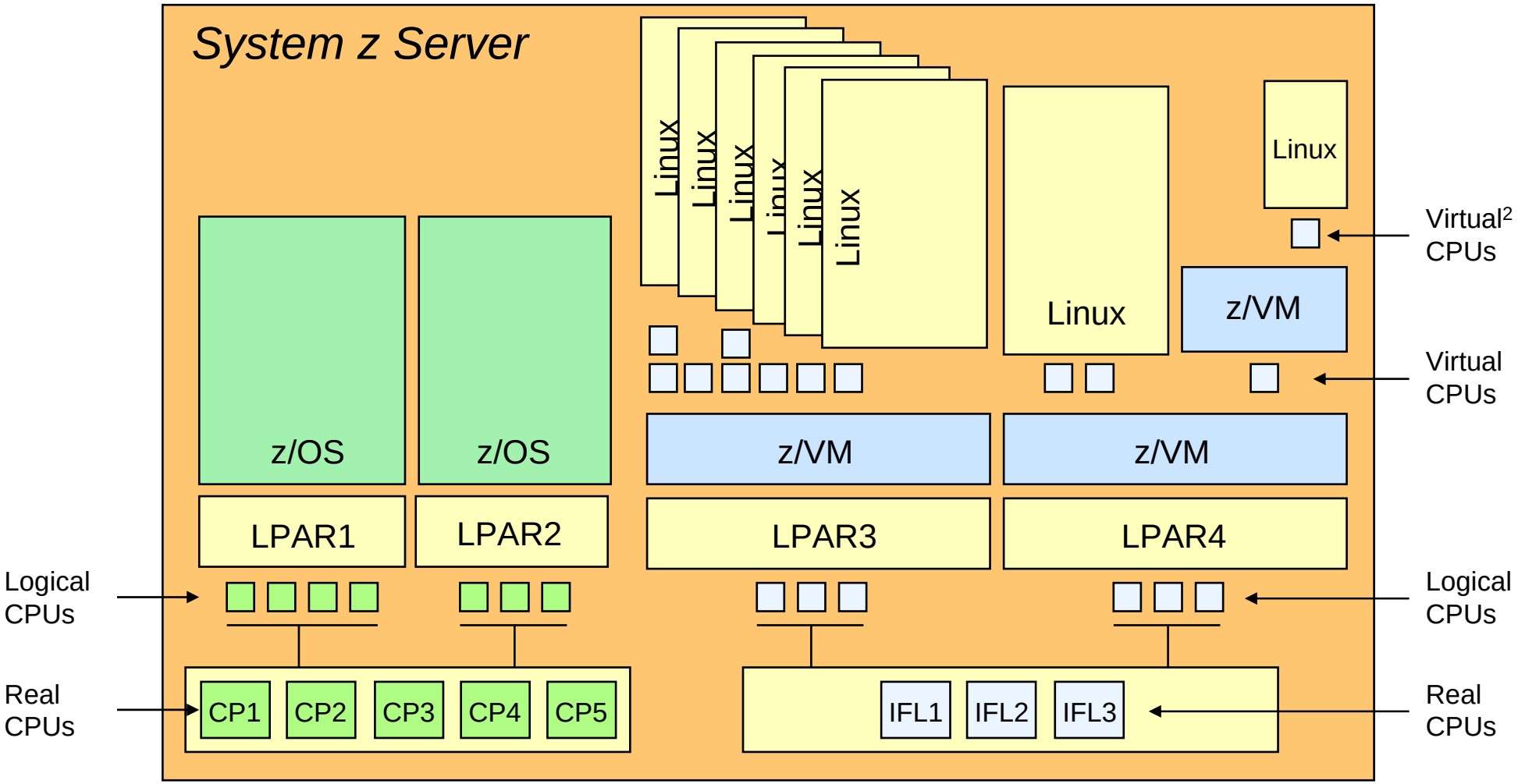
- Design a good paging subsystem
  - Define central/expanded storage near an 80/20 ratio (Up to a maximum of 2GB xstore, unless you have very large guests of 30GB and more)
  - Define paging disks with fast access and no bottlenecks. Get help with this if you need it.
- Communication:
  - Use VM vswitch to communicate between guests in the LPAR.
  - Use Hipersocket to communicate to a guest on another LPAR on the same CEC. If there is enough data being moved to warrant it.
- Linux swap files:
  - We don't want Linux to page.
  - Use VM VDISKs for Linux swap files.
  - Define swap files as 15% of the guest size.
  - Do not use minidisk cache
- Change default VM SRM settings to:
  - STORBUF=300,200,200
  - LDUBUF=100,100,100

# Setting up Linux

- CPUs
  - Initially define Linux guests with 1 VCPU.
  - 2 VCPUs for WebSphere guests. (Contact me for rationale and discussion)
  - Never more than the # of physical IFLs.
  - Then adjust # of VCPUs to match actual CPU usage.
  - Can use CPU Hotplug deamon to adjust on the fly.
- Memory
  - Define Linux guests to be the minimum memory size needed to run your WebSphere or other Linux applications.
  - Do not give Linux more memory than it needs, it will just waste it (from z/VM's point of view).

# What this all looks like so far

Note: There are typically dozens or hundreds of Linux servers in a z/VM LPAR.




z/VM skills are essential to successful virtualization

# Good Memory Allocation for Linux

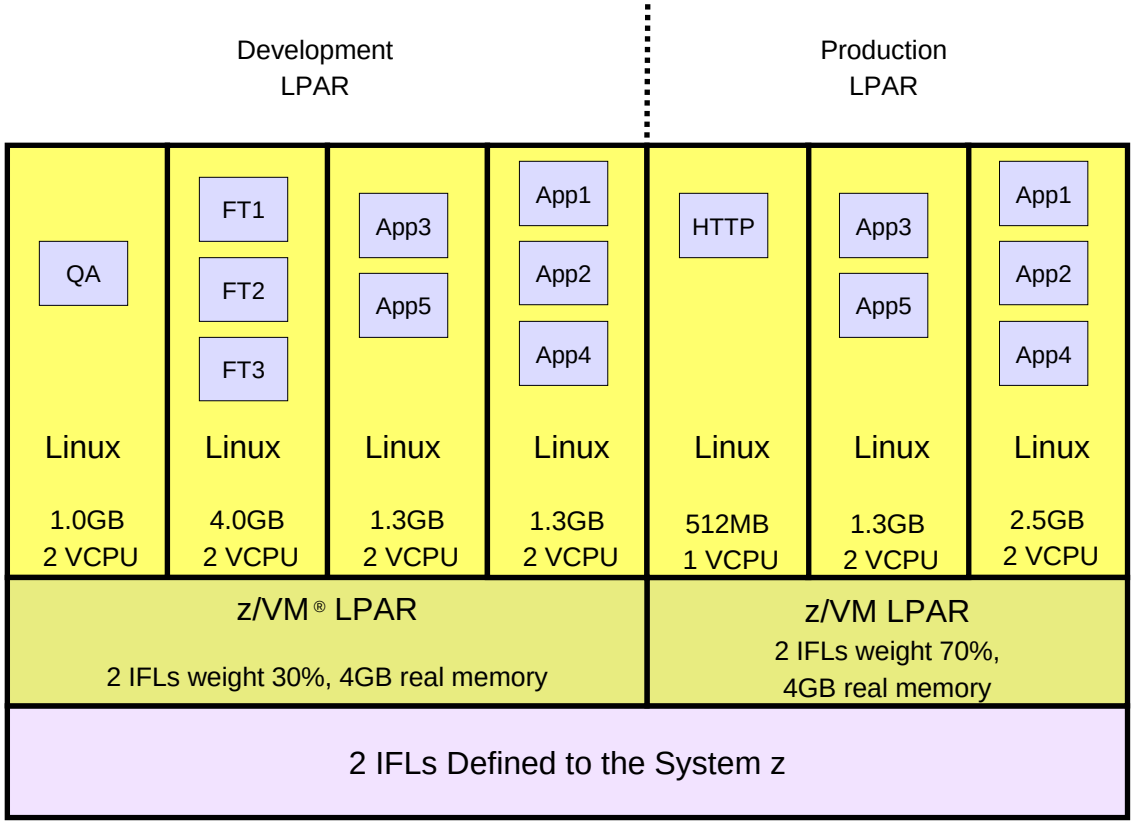
- Your Goal -- Give Linux as little memory as possible, without causing it to swap.
  - Use z/VM VDISKs for swap disks. Allocate no more than 15% of the guest size to the swap file size. This small swap file ensures that Linux will not be able to heavily swap.
- Use the “si” and “so” columns of the vmstat command output to determine if Linux is swapping. As long as these numbers stay in single digits or zero, you are good.
- Increase the guest memory at z/VM if there is too much swapping seen (hundreds of pages/sec for several minutes).
- Decrease the guest memory at z/VM if there is too much free space. (Remember to add buffer/cache to get an accurate count of free space).

```
lnx1: /home/testuser>free
             total       used       free     shared    buffers     cached
Mem:       1027540    1005928      21612          0      90772     493756
-/+ buffers/cache:     421400     606140
Swap:      1052248     102400     949848
```

# Overcommitting Memory

What does this mean?

You are "overcommitting" memory when the total defined memory of all the started guests is larger than the physical memory assigned to the z/VM LPAR.

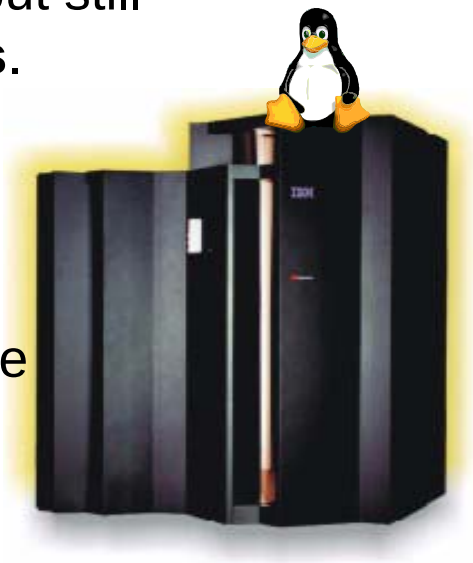



Please note that the memory and CPU values above are not necessarily recommended values, but used to illustrate a point.

- **Your Goal:** Reduce costs by overcommitting as much CPU and memory as possible, without impacting performance.
- One of the trickiest concepts of virtualization.
- In this example:
  - Overcommit CPU:
    - 13 VCPUs. 2 Real.
  - Overcommit Memory:
    - **Prod**: 4.3GB virtual, 4GB real. (ratio: 1.1:1)
    - **Dev**: 7.6GB virtual, 4GB real. (ratio: 1.9:1)
- The amount of overcommitment varies greatly by workload.
- You may be able to achieve:
  - **Prod**: 1.1:1 – 2:1 (Virtual:Real)
  - **Test**: 1.5:1 – 3:1
  - **Dev**: 2:1 – 5:1
  - Valid only for 25 or more guests.

# Good Memory Configuration is Crucial for Good Performance – The main points

- The Environment
  - Make sure **z/VM is not excessively paging**. Make sure you have enough real memory and that Linux guests are small.
  - Make sure **Linux is not swapping** (much). Make sure Linux virtual memory is large enough.
  - Make sure you have the CPU Power to drive the workload. Get a processor Sizing.
- WebSphere Tuning
  - Tune the JVM Heap size so that it is as small as possible, but still large enough so that the **JVM is not doing excessive GCs**.
  - Follow WAS doc for other small tuning tweaks for Linux.
- The Application
  - The application has the **largest impact on performance**.
  - **Inefficient code** that nobody noticed on dedicated hardware **will be noticed** on virtualized hardware.
  - Ask IBM for an application review for performance.



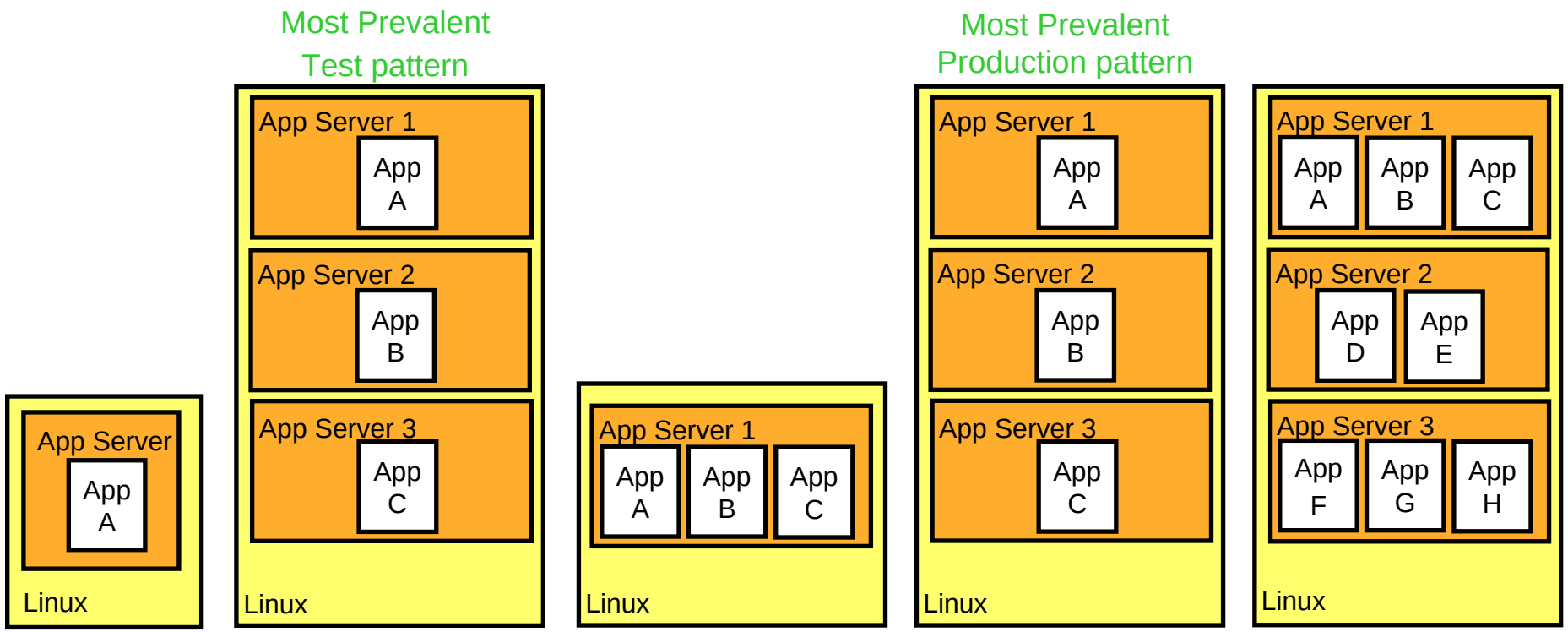
# Setting up WebSphere

WebSphere. software

- Generally there is little WebSphere tuning that needs to be done
- Most important to adjust the size of the JVM Heap
  - JVM Heap = memory used by the application(s) running in an app server.
  - Adjust max size to fit your application:
    - Monitor the JVM heap used by the application, while in development and QA test. Use:
      - Use Tivoli® Performance Viewer that comes with WebSphere.
      - Use Verbose garbage collection trace that comes with WebSphere.
    - Adjust max value so that JVM heap remains about 30% free.
    - Use your refined estimate for production.
  - Save memory costs by following these steps. (Contact me for a detailed process)

See my other presentations:
- Sizing memory for WebSphere Applications
- Efficient Configurations of WAS servers

# Deployment Patterns that save z/VM Memory




"Development" "Test" "Production"

Intermittant ⟷ Constant
CPU Requirement

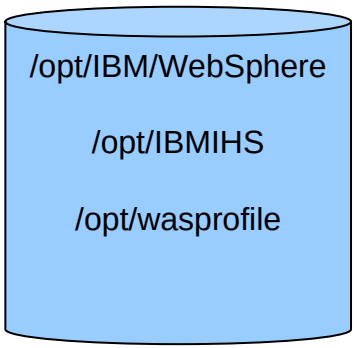
Many ⟷ Few
Number of Guests Needed

But there are many other reasons to pick a deployment pattern besides resource efficiency.

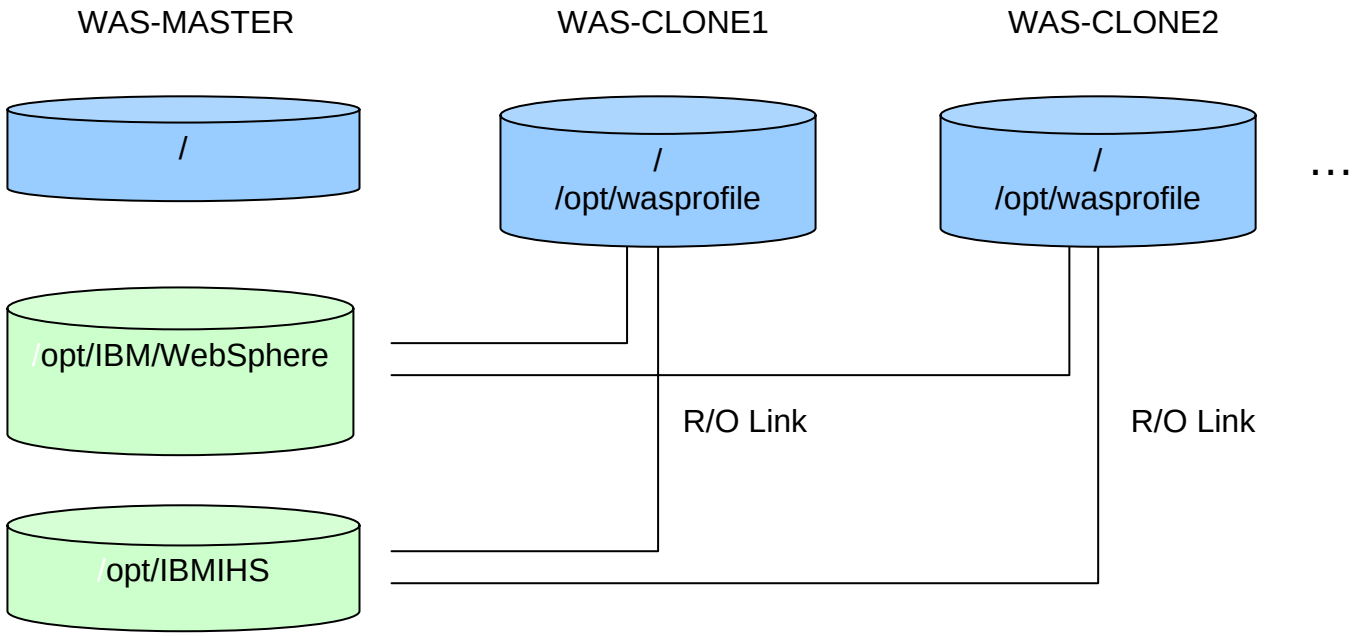
# Simplify Maintenance -- Share a WebSphere Install

- Install WAS once, share it among unlimited Linux guests.
- Apply maintenance once for all guests.
- Roll out maintenance or version upgrades at will.

Each Linux system sees this logical view.




Physical disk layout.



# Monitoring the Performance of this stack.

This is a complex, layered arrangement of software. It needs to be monitored so that you can see and debug performance issues.

## You must be able to monitor:

1. What resources the LPAR is getting.
2. What resources the guests are using
3. Performance of Linux on each guest
4. Performance of Java applications inside WebSphere

- Use a z/VM monitor (1,2,3)
  - Omegamon XE displays LPAR, z/VM, and Linux perf data.
  - So does z/VM Performance Toolkit, or Velocity Software.
- Use a WebSphere/app monitor (4)
  - Tivoli Monitoring for WebSphere
  - Wily Introscope
- Use Java profiling tools (4)
  - JinsightLive – IBM profiling tool.
  - VerboseGC trace.
  - Tivoli Performance Viewer.


Omegamon XE

# Getting Started

1. Take one of the classes on VM and Linux for zSeries.
   A good background in z/VM concepts and operations is crucial for success.
2. Estimate the CPU and memory required by your applications.
   IBM techline can do both.
3. Ask for a Solutions Assurance Review, with IBM or your BP.
4. Define Linux Guests. Install Linux and WebSphere. Deploy apps.
5. Tune memory and overcommitment iteratively:
   1. Test your applications.
   2. Monitor CPU utilization, memory usage, performance of apps.
   3. Tune memory settings based on your data.

# Getting more Information

- Linux Library website:
  - A multitude of helpful papers.
  - On the web at: www.ibm.com/servers/eserver/zseries/os/linux/library/
- System z Linux education
  - http://www.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=course_list&subChapter=467&subChapterInd=S&region=us&subChapterName=Linux+for+zSeries+and+S%2F390&country=us
- Step-by-step instructions for creating Linux virtual servers:
  - Virtualization Cookbook for SLES10. Redbook SG24-7493-00
  - Virtualization Cookbook for RHEL5. Redbook SG24-7492-00
- *Architecting z/VM and Linux for WebSphere.* Companion paper to this presentation.
  - Introduction to Memory configuration for z/VM, Linux, and WebSphere.
  - http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101803
- *Sharing a WebSphere Application Server V7 Installation Among Many Linux for IBM System z Systems*
  - Procedure to share the WebSphere binaries among many virtual servers.
  - http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101817

*The End*