

Enterprise Linux
Defining the Enterprise
\$Id\$
Christoph Doerbeck

Table of Contents

Who are you listening to?.....	3
Presentation Outline	3
What is Enterprise Computing?.....	3
Service Level Agreements (SLA)	4
Availability.....	4
Performance & Scalability	5
Monitoring	5
Backups.....	6
Capacity Planning.....	6
Storage Management.....	6
Product Lifecycle Management.....	6
Root Cause Analysis.....	7
Total Cost of Ownership.....	7
Review & Final Comments	8

Who are you listening to?

Before you take the advise of anyone, know your source. Especially in forums like this, where the author announces a disclaimer of liability.

I make no claim that what I report is accurate. If you wreck your systems/data/life based on knowledge which I inaccurately report, you cannot hold me nor those that I work for responsible.

This document (presentation) is intended for consumption by responsible individuals in the spirit of sharing knowledge about Linux and Open Source Software (OSS).

That said, here is my background

- Unix Admin Experience (10+ years)
- Dec Ultrix, HP/UX, IBM/AIX, Linux, SUN/Solaris
- Education, R&D, Retail
- Instructor (1 year)
- BLU member & contributor (5 years)
- BS Computer Science Engineering

Presentation Outline

1. What is Enterprise Computing?
2. Essential Components of Enterprise Computing
3. Open Discussion

Note: This is a NEW document and much of it is based on my opinion at this point. Hopefully, the input of others over time while improve the accuracy of the topics and make this presentation more useful.

What is Enterprise Computing?

I have no idea! But when I hear about it, the conversation usually includes terms like:

- Service Level Agreements (SLA)
- Availability (redundancy, HA, clusters)
- Performance & Scalability (vertical vs. horizontal)
- Monitoring

- Backups
- Capacity Planning
- Storage Management
- Product Lifecycle Management
- Root Cause Analysis
- Total Cost of Ownership

Service Level Agreements (SLA)

SLA definition aside, I'm also using this category as a catch-all for topics that aren't broken down below.

- Agreement between provider & customer which guarantees level of service
- To achieve SLAs, you need the other categories (Performance, Availability, etc...)
- Cost vs. Performance offerings
- Do you need a 24/7 Help Desk?
- How about a 1-800 nationwide call number?
- Remote Access
- Escalation Procedures
- Vendor Relationships (Management)
- How many machines / locations do you plan to support? (scale)
- Do you need lights-out management adapters? (I hope not!)

Availability

- The infamous 9s
- Server Availability
- Application Availability
- Single Point of Failure (SPOF) & risk assessment/management

Traditional techniques to improve availability?

- Climate controlled data center

- Redundant power from multiple power sources (ex: multiple lines, batteries, generators)
- Server class hardware with redundancy (ex: hot swappable devices [disks | power | PCI], ECC memory, raid, DMP)
- Clusters (ex: many systems which load balance and shift work load)
- HA software (ex: symmetric & asymmetric HA which take over services upon failure)
- Hire, train & retain responsible capable employees

Performance & Scalability

- Vertical
 - Symmetric Multi Processing (SMP - more CPUs per system)
 - VERY complex hardware
 - VERY complex software
 - Often measured in terms of linear scalability (cpus vs. performance)
 - There IS a limit! Whether it's hardware or software, at some point adding more cpus is either impossible or provides no measurable gain
 - Also consider that NOT all applications thread well for SMP
 - Know your SPOFs!!!
- Horizontal
 - More systems per application (clustering)
 - Performance is achieved thru network infrastructure & application design
 - Problems with shared access to read-write data might surface
 - Again, it's important to identify SPOFs

Monitoring

- Most Enterprise shops are NOT remodelling to support Linux
- Thus, Linux HAS to integrate (embrace and extend ;-)
- 3rd party client software availability (Tivoli, TNG, etc...)
- Alert modelling and response planning

Backups

- Most Enterprise shops are NOT remodelling to support Linux
- Thus, Linux HAS to integrate
- 3rd party client software availability (Veritas, Legato, ADSM, etc...)
- Offsite storage management

Capacity Planning

- Throwing darts to choose what hardware is suitable for a project is poor
- Purchasing hardware that sits idle is poor and cost prohibitive
- Need tools for accurately measuring & modelling capacity usage (sar, landmark, teamquest, tivoli, etc...)
- Process level accounting
- Need a historical baseline
- Need people who understand the baseline and can make accurate predictions (retention & training)

Storage Management

- Most Enterprise shops don't have single systems with single IDE disk drives
- Multiple terrabytes of data
- Disk storage is usually made up of mixed disk sizes, types & vendors, which have been aggregated over years (perhaps decades?) of service
- Storage Array Networks (SAN)
- Hierarchical Storage Management (HSM)
- Things like journal filesystems, logical volume management (striping/mirroring/raid)
- Again, most Enterprise shops are NOT remodelling to support Linux

Product Lifecycle Management

- Hardware ages - eventually it will fail (MTBF)
- Software ages - eventually it won't run on your newer hardware
- Vendors make improvements, release new stuff and end-of-life the old stuff
- Vendors stop supporting the old-stuff
- Without vendor support, your SLA is in jeopardy
- We can all agree that buying the cheap PC-of-the-month is not feasible for an Enterprise
- Product Lifecycle Management is an art that balances the cost and longevity of your "stuff"

Root Cause Analysis

Stop me if I rant too long on this topic!

- The ability to identify EXACTLY what caused a specific fault
- Memory leaks, poor performance, system panics, etc...
- Identify what clues & evidence are required
- Assure that evidence is saved and accessible
- Who's your guru (Crime Scene Investigator)
- What's your SLA?
- How many gurus do you need?
- Kernel CORE dumps are too often overlooked, but in my opinion an absolute must
- Native Linux kernels have NOT been able to produce cores
- More importantly, faults that a kernel reports are often lost on the system console
- Serial Consoles can help
- Steps in the right direction (ksymoops)

Total Cost of Ownership

- Consider all of the above topics and now put a \$\$\$ on it
- Consider one time vs. re-occurring costs
- Application consolidation might help (multiple apps per server)
- Server consolidation might help (LPAR/Domains)

- Virtual Machines might help improve capacity utilization (zVM, VMware)

Review & Final Comments

So, what is enterprise computing? In my opinion it's an environment where all of the above play a critical role in the successful day-to-day operation of a computing environment. It is a formula which takes into account all of the mentioned components and calculates the relationship from SLA & TCO. Most importantly, there can be no unknowns. For any crisis, there needs to be a clear procedure for resolution.

So, is Linux ready for the enterprise? I leave this to discuss amongst yourselves...