

new idea
ENGINEERING



Mark Bennett

Search and the Virtual Machine

Agenda

- Intro / Business Drivers
- What to do with Search + Virtual
- What Makes Search Fast (or Slow!)
- Virtual Platforms
- Test Results
- Trends / Wrap Up / Q & A

Business Drivers?

Virtual = Flexible

Virtual machines give you flexibility
in **testing** and **deployment** .

But perhaps **not** short term cost savings ...

Draws on Many Areas

- General **Virtualization** Advantages
- Search Engine **Architecture**
- Search Engine **Performance**
- Hardware, OS and **Networking**
- **Distributed** and Parallel Computing
- Vendor Licensing and **Pricing** Policies

Key Issues for Virtual + Search:

- **Vendor** “support”
- Search engine **architecture**
- **Automation** of common tasks
- **Predicting** performance
- Training and buy-in from **IT** and **Dev**
- **Licensing** (tech & legal)

Search Tasks that Virtualization Can Help With

“Search on Virtual” Tasks/Benefits



- General Benefits
 - **Standardized** machine setups/images
 - Snapshot of **complex** Restore Points
 - **In Sync** on OS, Java and Vendor patches
- Search Engine specific benefits
 - Multi-node Dev environment & **Frame of Mind**
 - **SAME topology** in Dev, Stage & Prod farms
 - Failover and **rollout** flexibility

Validation Tasks

- System bootstrapping
 - Have we **completely documented** bare-machine to a customized and running search farm?
- Multi-Node Installs
 - **ssh & keys**, search engine config & **licensing**
- Testing Failover and Recovery scenarios
- Network **firewall** and security issues
 - Distributed search uses LOTS of **ports!**
- Security issues
 - SSO, LDAP, search engine passwords
 - Spider **Über Login**



Simulating Hardware Failures

- Kill a machine, validate failover, **recovery**
- Kill a **disk**, validate recovery, failover
- VMWare hardware abstraction layer



What About Search Engine Performance?

Fundamental Performance Questions

- When does it really matter
 - Prod vs. testing procedures, topology
 - Virtual ***Might*** Be Fast Enough for Prod
- Can you reliably scale virtual benchmarks to physical benchmarks??
 - Can you ever just say:
"multiply by **125%** for virtual to physical" ?

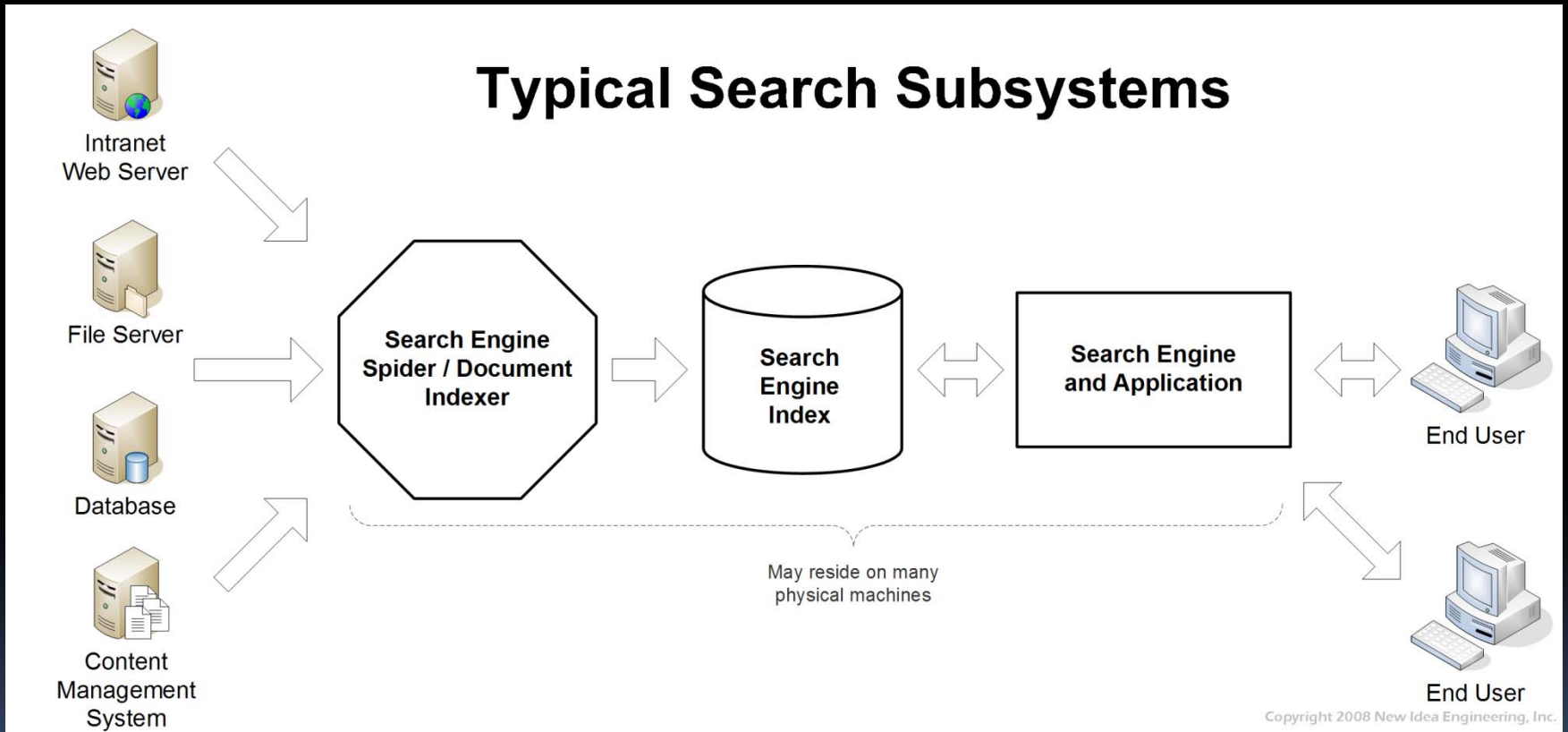


Search Engine Subsystems

- Data Processing
 - Spider (FETCHER) and connectors
 - Indexer / search indices
 - Distribution, Replication, optimization
- Handling Search Requests
 - Inbound application stack
 - Query preprocessing, expansion, early binding and federation
 - CORE search engine -> search indices
 - Results list formatting, facets, late binding
- Admin, IT, Monitoring, Scheduling & Security



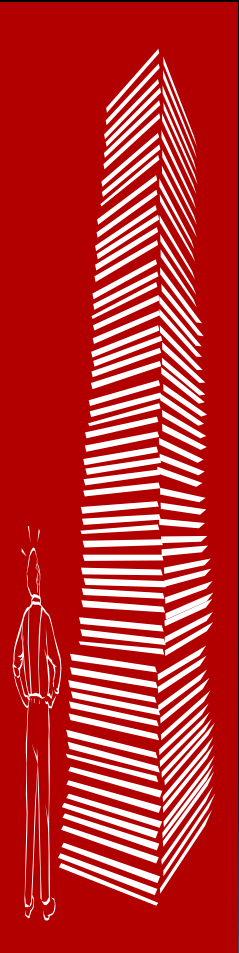
Search Engine Diagram



Sizing and Scaling of Search

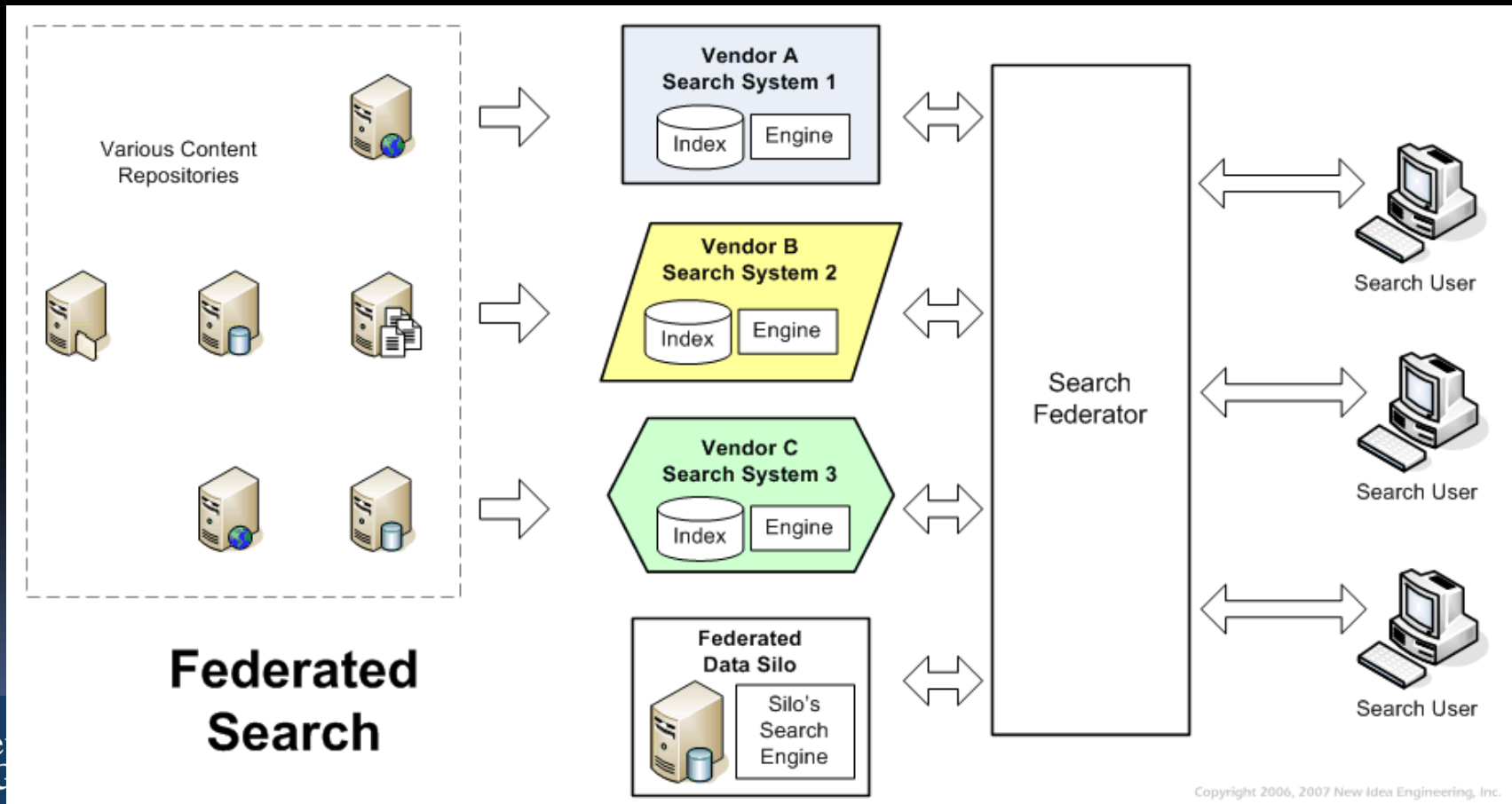
Scalability

- Monolithic vs. **Distributed** Software
- Data / Documents **vs.** Queries
- Documents **vs.** Terabytes, **Text** vs. Graphics
- **Indexing**
 - Overall average, **Latency** and Failover
 - # of Documents vs. # of **Transactions**
- Queries
 - **Peak** vs. Average
 - Average Throughput vs. **Individual Query Time**
 - Compound Queries, Federated Search



How Search Engines Scale

- Monolithic vs. **Farmed** vs. Distributed
 - Rows and Columns (meaning **machines**)
- It's mostly about where the **Index** lives



How Search Engines Scale: Spider

- Spider “breakups”
 - **Fetching** data
 - **Parsing** data
 - Writing indexing data to **disk**

How Search Engines Scale: Search

Search “breakups”

- App server
- Query preprocessing
 - Query Parsing, query “**Cooking**”
 - Stemming, Thesaurus, Search **Filters**
- Formatting Results
- Low level search – the **core** engine
- **Relevance** and Ranking
- Post-Filter, Sorting and Dedupe
- Monolithic vs. **Federated**

Bottlenecks for Search Engine Performance

- Usual Suspects: **High Latency** Ops
 - Network **Fetches** (fixed with threading)
 - Disk I/O **Round Trip**
- A few others in **some** cases:
 - Huge **binary** files
 - “**Extreme**” document indexing
 - **Atypical** usage patterns
 - **Starved** machines (**memory**, cpu)

Other Scaling Techniques: Google Map/Reduce and Hadoop

- General: **Multi-core** driving new parallel ideas
- Google popularized **Map / Reduce**
- Open Source responded with Hadoop and **Nutch 0.8**
- Maybe Clouds ...



Virtual Platforms for Search

Types of Virtual Machines

- Older Emulators (ex: Parellez, etc.)
- VMWare, Xen, Microsoft, Baynote, others
- Full HW-software layer
- Special Client OS layer
- Special Host OS layer
- Virtual Hardware components
- Virtual Networks
- Cloud Computing
- Some overlap with Blades

Operating System Issues:

- OS Selection
 - Host OS
 - Client OS
 - 32 vs. **64 bit** OS
 - 32 vs. 64 bit Search Engine Version
- General
 - Networking on **clients**
 - Disk storage strategy
 - **File Handles** – local vs. network
 - OS Licensing and EULA issues with Microsoft

The Cloud

Not quite primetime yet for search, but some good choices are evolving:

- Amazon
- Google
- Microsoft in 2009
 - For now: use **RackSpace** or **Joyent**



Clouds can Add Temp Capacity

- If you only need to Spider **100,000,000** pages **once** a month
- You've just won a contract to index the **.gov** top level domain
- You need to do a one-time global reindex

Amazon Cloud: Pros

- Extreme Flexibility
- Low entry cost,
pay as you **grow**
- Based on familiar **Linux** environment
- Prebuilt **template** virtual machine images
- Web based Administration



Amazon Cloud: Cons

- **Micromanagement** of machines
 - Can be mitigated by **third party** software
- Assignment and visibility of **IP addresses**
 - Dynamic DNS not good for busy sites
 - Can pay extra to fix
- No **Windows** based servers
- Lacks traditional Tech Support channel
 - Complaints about other channels during the Beta

Google Cloud

- Pros

- No micromanagement of machines
- Free to start, pays as you **grow**
- Uses **Python** (so far)

- Cons

- Unlike any of your existing platforms
- No way to **port** existing apps
- No **relational** database, Google: “joins = bad!”
- New Data Store programming model flawed ?
 - Prone to reference leaks and cycle-explosion
 - Google architects seem to be in denial mode



Test Results

Expected Results

3% to 20% overhead

Leaning towards 20% because of non core-CPU operations

Actual Results

In range and pretty **good**:

Average: approx **10%**

Range: max **0%*** to **17%**

Had predicted **3%** to **20%**

Overall results:

Slightly **better than expected**

Test Platform: HW & OS



- **HP** mid-tier workstation
 - CPU: Dual Core
 - Memory: **8 Gigs**



- MS Windows **Server 2008 64-bit**
 - For **both** Host and Guest
 - With Hypervisor



- Sun Java SE SDK **v6 u10 64-bit**
 - JVM set to **1 Gig** of RAM

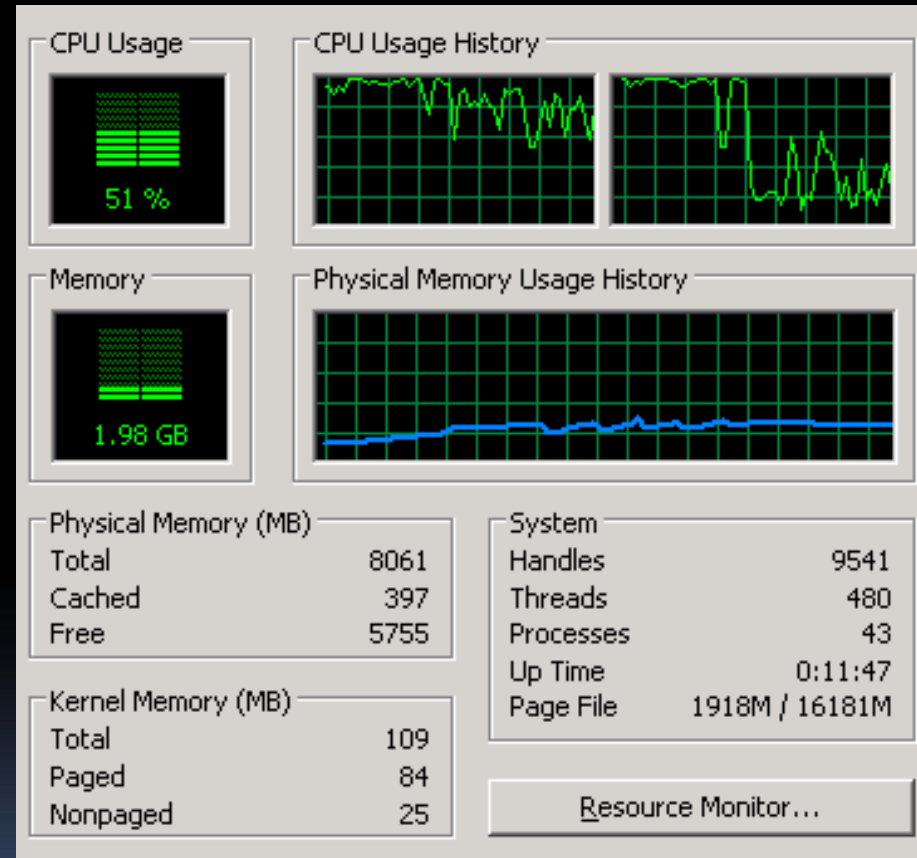
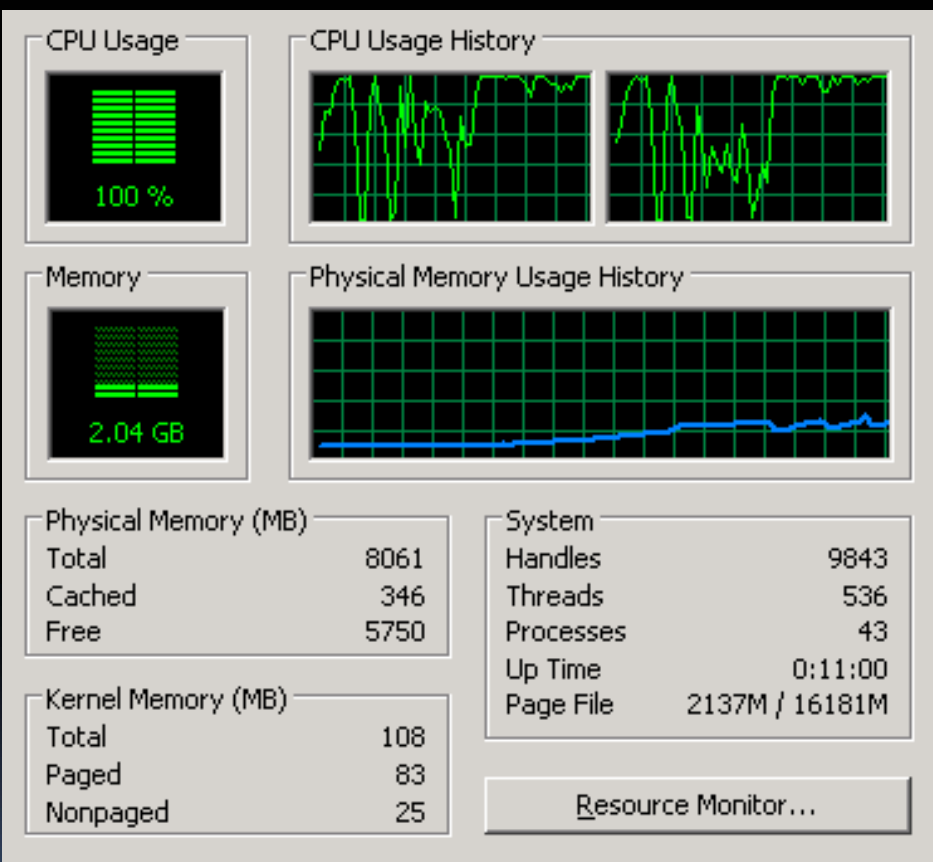
Test Platform: Spider and Data



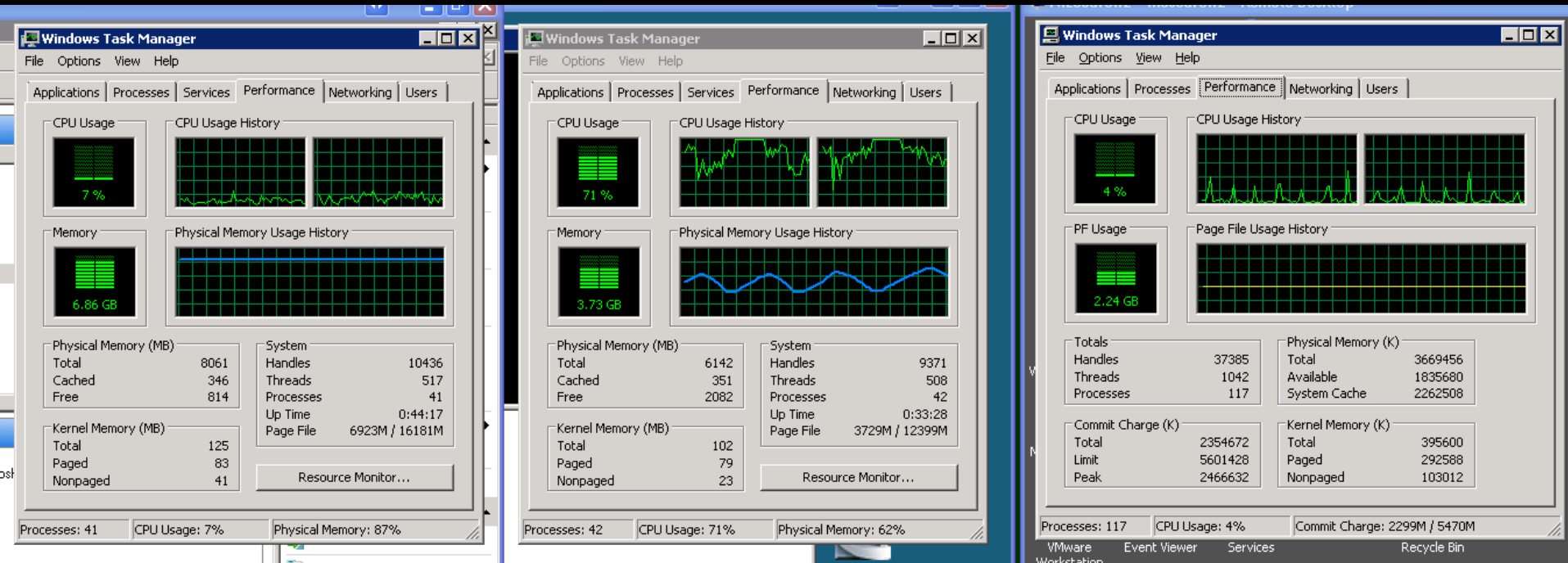
- Spider: **Nutch 0.9**
 - **Lucene** based
 - Includes **Hadoop** engine
 - Set to **50** threads
- Data: **Enron** public emails
 - **1.5 Gigs**, approx **1/2 M** messages
 - Served by **IIS** on
 - Separate local machine



Spider vs. Index CPU



Host, Client & Web Server



Wrapping Up...

What was Learned

- JVM and Memory
 - Not strictly limited by `-Xmx` switch
- Nutch and CPU usage
 - Spidering is distributed, indexing is not
- Virtualization and Search
 - Hypervisor or “bare metal” is viable option!

Is Your Search Vendor “Virtualize-able” ?

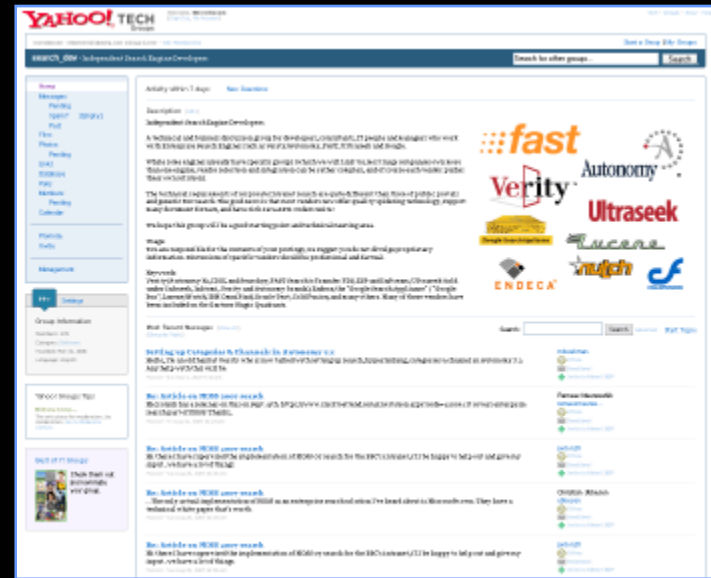
- Google, **no way** (at this time...)
- FAST, yes... mostly.... and try **ESPDeploy**
- Others ... TBD ...

The Near Future

Search and Virtualization

- Providing **Flexibility** and Validation
 - vs. huge Cost savings
- Scalability across many **commodity machines**
 - IT and **Developers** need to think “**distributed**”
- Ask Your **Vendor**
 - “Official **Support**”
 - Benchmark **metrics**
- **Clouds**: Search is not there... **Yet**

Resources



Newsletter & Whitepapers:
www.ideaeng.com/current

Search Dev Newsgroup:
www.SearchDev.org

Blogs:

EnterpriseSearchBlog.com

SearchComponentsOnline.com



Finish Line

Review & Questions

new *idea*
ENGINEERING

General Info info@ideaeng.com



Mark Bennett mbennett@ideaeng.com

main 408-446-3460

cell 408-829-6513