The New Darkwing

Joe St Sauver, Ph.D. (joe@uoregon.edu)
Director, User Services and Network Applications

Computing Center Staff Briefing
January 19, 2004

http://darkwing.uoregon.edu/~joe/new-darkwing/

This Talk

- This talk is designed to give CC staff an overview of where Darkwing will be going/how it is being upgraded.
- We’ve attempted to hit the sweet spot between giving you too much detail and being vague.
- One area where we <u>will</u> be <u>intentionally</u> vague relates to the production network attached storage (NAS) filers because that procurement is ongoing at the time this is being written.
- Everything should be viewed as being subject to change; nothing’s written in stone (but we would like to stick to the plan if at all possible :-))
- While I’m delivering this talk, the new architecture is the work of a lot of people, and will require a lot of people’s continued effort to bring it to fruition.

The Recent Load Problems

- While people often think about Darkwing as an mail server, it actually does many other key things too, including web, authentication, streaming media, etc.
- Although Darkwing is/has been a mission critical host, it is six years old, has been struggling under its load, and had become basically unusable for some applications
- For example, green web email had gotten VERY slow
- Some services (such as POP and IMAP) could not be technically throttled without replacing inetd with xinetd; user response to requests for voluntary load reduction were less than fully successful even when users were directly and personally contacted.
- Once the system load got high enough, some services (such as mail delivery) simply ceased to run entirely

But The Problem Wasn’t Just Load…

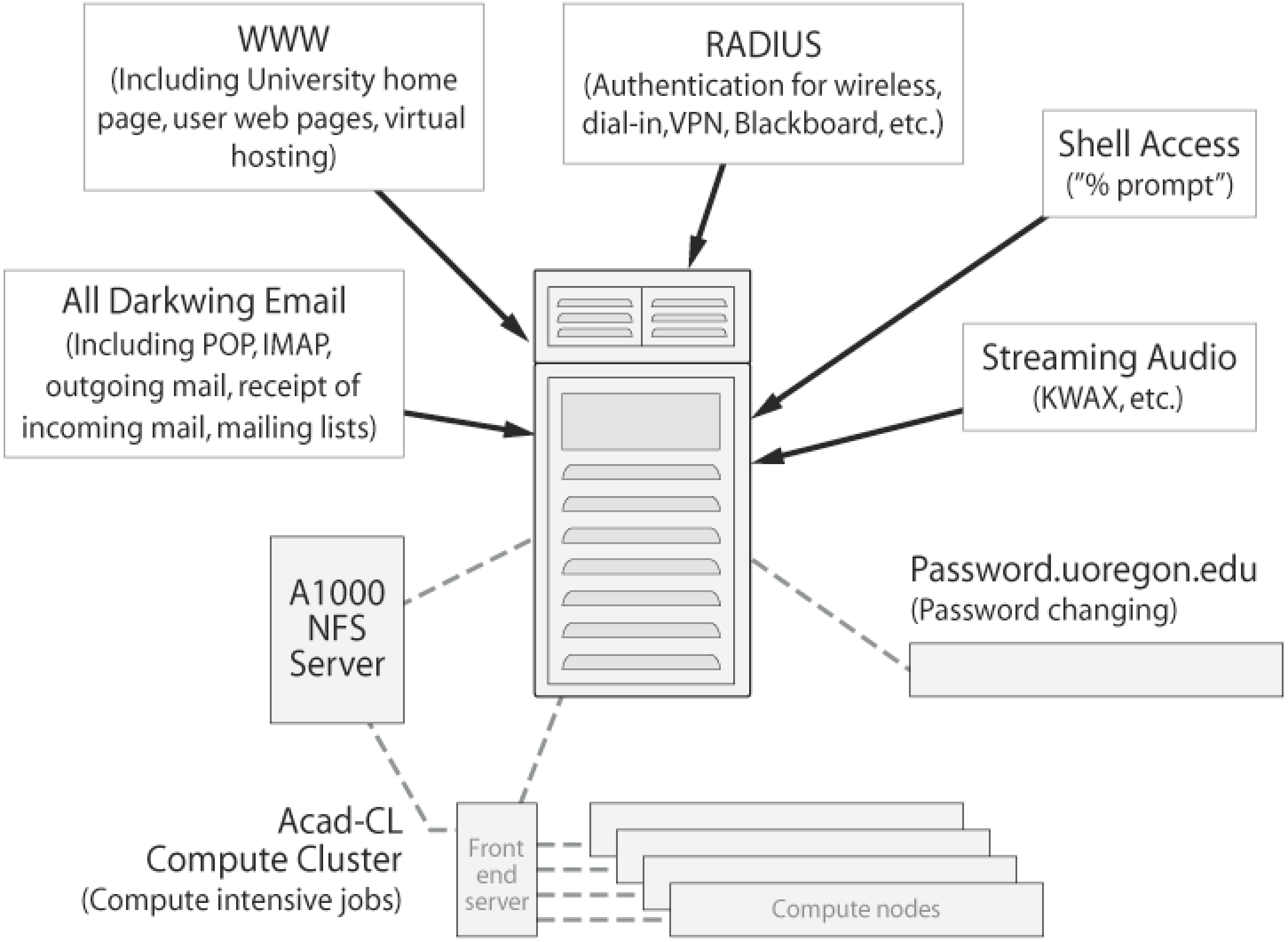
- Somewhere along the line, we’d made two fundamental mistakes:

  -- **Everything ran on Darkwing**, and

  -- **We only had one Darkwing**.

Classic Darkwing (Simplified)

Everything ran on one box: A tangled, interconnected …



WWW
(Including University home page, user web pages, virtual hosting)

RADIUS
(Authentication for wireless, dial-in,VPN, Blackboard, etc.)

Shell Access
("% prompt")

All Darkwing Email
(Including POP, IMAP, outgoing mail, receipt of incoming mail, mailing lists)

Streaming Audio
(KWAX, etc.)

A1000 NFS Server

Password.uoregon.edu
(Password changing)

Acad-CL Compute Cluster
(Compute intensive jobs)

“Okay, So… Let’s Just Buy A Bigger Darkwing, Just Like Our Current One?”

- One potential solution to the problem of classic Darkwing being “out of gas” would have been to just buy a bigger version of the same thing. However…

  -- Large monolithic servers have limited markets. Because of that, large systems capacity is expensive in $’s/unit capacity vs. capacity from smaller systems

  -- We’d still be on Solaris

  -- We’d still have too many services on one box

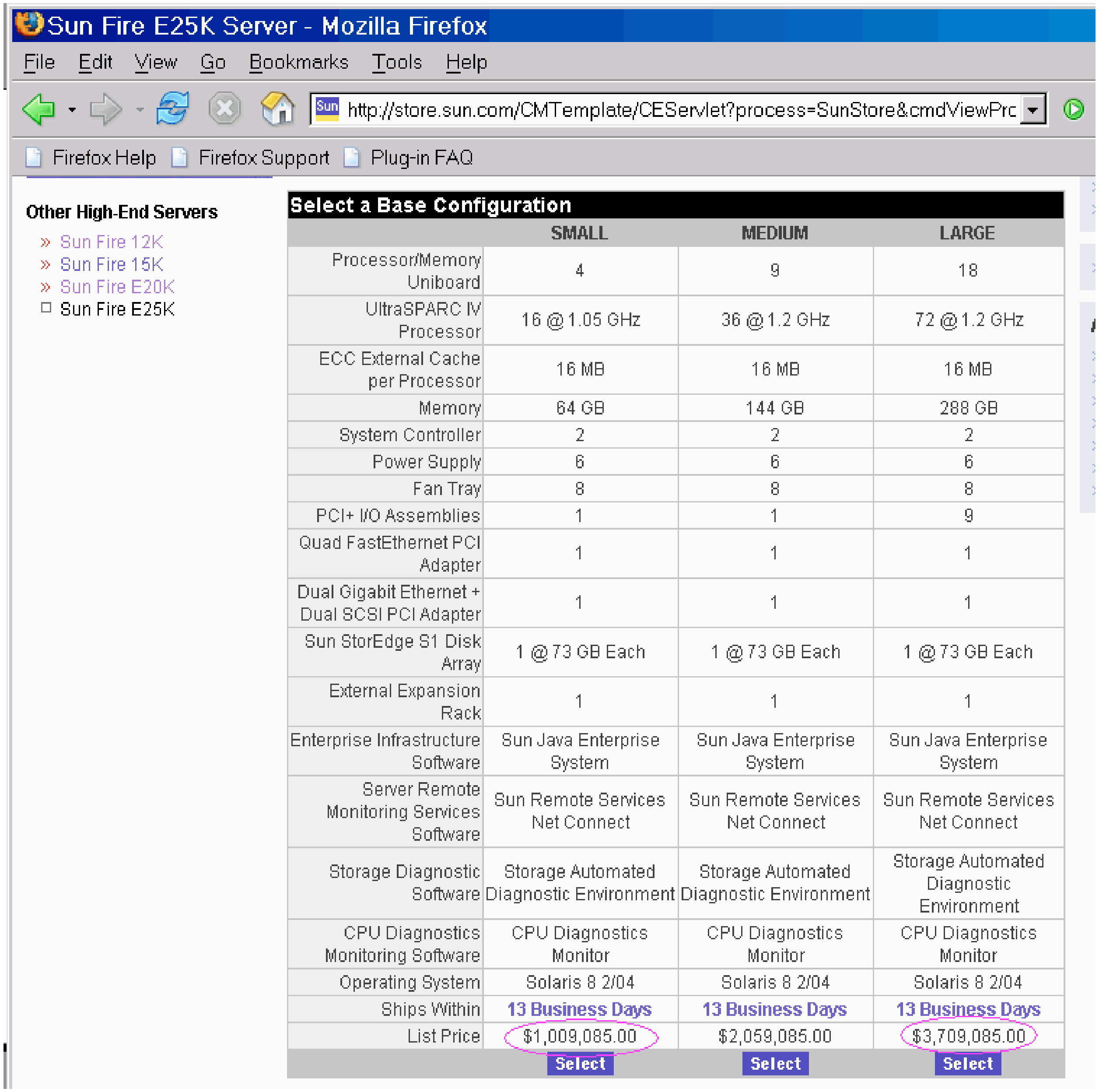
  -- Running a single system would still result in periodic downtime for maintenance

  -- Continuing to run everything on one system would do nothing to help our disaster planning/survivability

  -- At some point, we would NOT be able to keep buying bigger and bigger monolithic systems

  -- What about storage?

Some Large Sun Options, For Context


Sun Fire E25K Server - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://store.sun.com/CMTemplate/CEServlet?process=SunStore&cmdViewPrc
Firefox Help
Firefox Support
Plug-in FAQ

Other High-End Servers

- » Sun Fire 12K
- » Sun Fire 15K
- » Sun Fire E20K
- □ Sun Fire E25K

Select a Base Configuration

	SMALL	MEDIUM	LARGE
Processor/Memory Uniboard	4	9	18
UltraSPARC IV Processor	16 @ 1.05 GHz	36 @ 1.2 GHz	72 @ 1.2 GHz
ECC External Cache per Processor	16 MB	16 MB	16 MB
Memory	64 GB	144 GB	288 GB
System Controller	2	2	2
Power Supply	6	6	6
Fan Tray	8	8	8
PCI+ I/O Assemblies	1	1	9
Quad FastEthernet PCI Adapter	1	1	1
Dual Gigabit Ethernet + Dual SCSI PCI Adapter	1	1	1
Sun StorEdge S1 Disk Array	1 @ 73 GB Each	1 @ 73 GB Each	1 @ 73 GB Each
External Expansion Rack	1	1	1
Enterprise Infrastructure Software	Sun Java Enterprise System	Sun Java Enterprise System	Sun Java Enterprise System
Server Remote Monitoring Services Software	Sun Remote Services Net Connect	Sun Remote Services Net Connect	Sun Remote Services Net Connect
Storage Diagnostic Software	Storage Automated Diagnostic Environment	Storage Automated Diagnostic Environment	Storage Automated Diagnostic Environment
CPU Diagnostics Monitoring Software	CPU Diagnostics Monitor	CPU Diagnostics Monitor	CPU Diagnostics Monitor
Operating System	Solaris 8 2/04	Solaris 8 2/04	Solaris 8 2/04
Ships Within	13 Business Days	13 Business Days	13 Business Days
List Price	$1,009,085.00	$2,059,085.00	$3,709,085.00
	Select	Select	Select

What About Alternative Approaches?

- If we’re not going to just buy a bigger version of what we had, what else could we do?

  -- we could do nothing (but the system as a whole would likely undergo congestive collapse, which would be something we’d rather avoid)

  -- we could try (again) to shed load, but that would be both hard to do and unlikely to provide either substantial immediate relief or required long term capacity; not a winning strategy

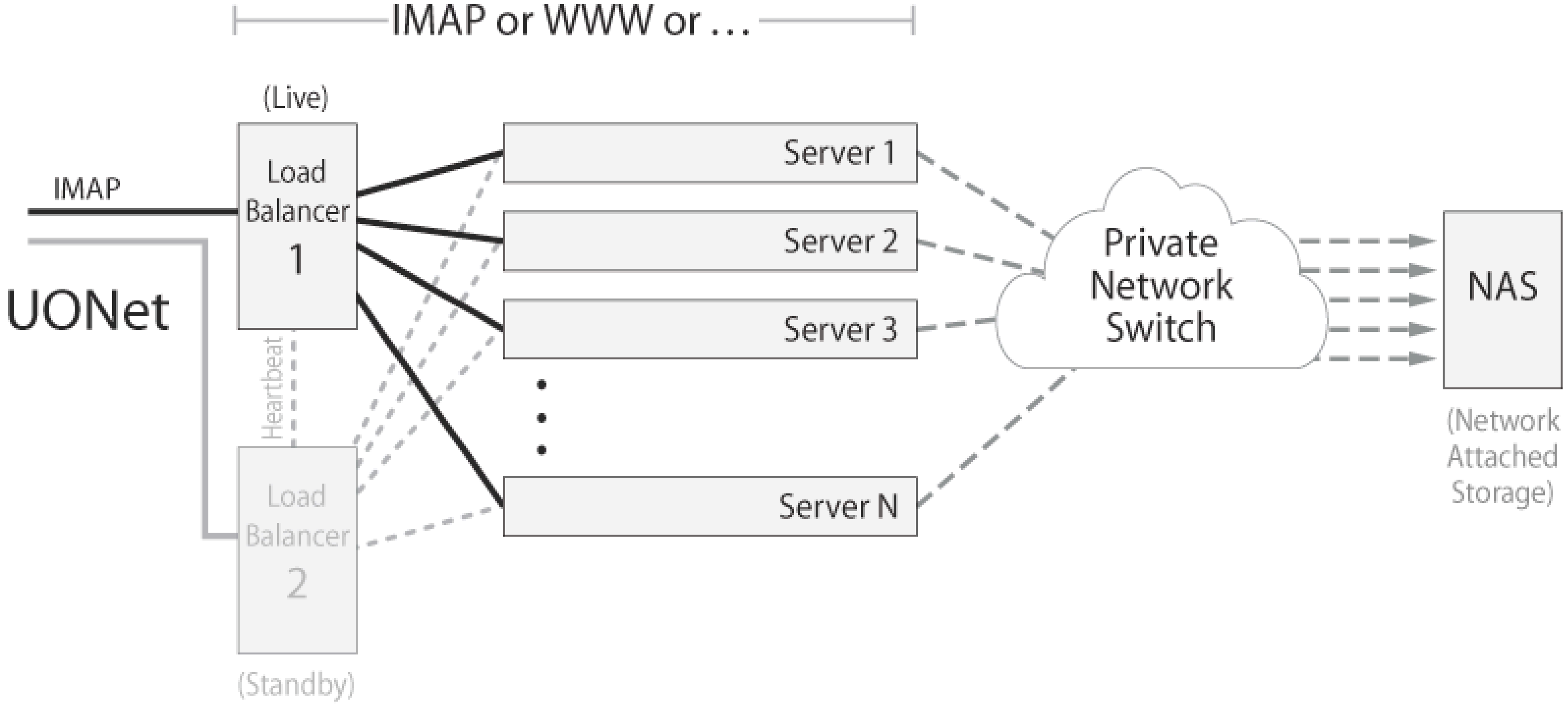
  -- OR we could try a new architecture that subdivides our load into chunks that could be handled by smaller systems, breaking the load up either by chunks of users or by service. Which of those two approaches to dividing the load should we do?

Dividing The Load

- Dividing by user requires that:
  -- each node run all services
  -- we have some mechanism to hash users to nodes in a user-transparent way
  -- we be willing to accept substantial potential licensing pain (if we needed to purchase and deploy licensed products at each node)
  -- if we need to add capacity, this implies shuffling users
  -- we accept it is hard to share load among nodes evenly
- Dividing by service allows us to:
  -- tailor nodes to the needs of particular services
  -- point traffic for a given service via DNS
  -- limit the extent of licensed products
  -- add capacity transparently by replicating servers behind a load balancer, should we need to do so

Sample Service Node, New Model

Each major service (or group of services) runs on its own set of redundant, load banced hosts



Server Building Blocks; Load Balancing

- An inherent part of the new model is the use of “building block” servers. We’ve chosen to use commodity dual AMD Opteron–based 1U rackmount systems which cost less than $5K each. [1U dual Opterons have proven themselves nicely in service as the core of the acad-cl compute intensive hosts]
- Multiple building block servers sit behind a pair of load balancers (which are themselves simply yet another building block server, this time configured to run the LVS load balancing software). Load balancers look at incoming connections along with backend server load, and then route incoming connections to a suitable server, either working round-robbin style or by monitoring the load on each backend server, and sending connections to the most lightly loaded host.

Resilience

- Achieving operational resilience requires the ability to work on a component of the system without having to take an outage for the whole thing.
- By deploying N servers behind a load balancer, we gain the ability to take one of those N servers out of service for maintenance or upgrades, without users even noticing.
- The down side of this approach is that it increases server count (redundant servers are required) and it requires a way to share load across the servers (e.g., use of load balancers).

What Do The Building Block Servers Actually Look Like?

An Evolving Issue: Do We Also Need *Smaller* Server Building Blocks?

- We may actually end up adding one additional type of cheaper (~$1K/unit) and less powerful building block server, for situations where we don’t need the horsepower of the dual Opterons (or even the horsepower of a single Opteron), but do want/need to field physically separate servers rather than aggregating functions on shared hosts.
- Examples: syslog hosts, monitoring hosts.

Some Notes About the New Model

- Besides fixing the load problem, the new model…
  -- gives us a clean path for future growth
  -- reduces the extent to which everything happens on a single monolithic system
  -- minimizes downtime for routine system maintenance
  -- eliminates reliance on expensive proprietary hardware
  -- eliminates reliance on proprietary operating systems
  -- potentially improves survivability in the event of a physical disaster (fire, flood, etc.) or network attack
  -- reduces growing cost of maintaining aging hardware
  -- gets Darkwing to the point where it had the capacity to handle the remaining-to-be-moved OpenVMS users
  -- sets the stage for consolidation of Darkwing/Gladstone
  -- enable new services (larger quotas, CIFS access, true email virus scanning, SpamAssassin scanning/tagging)

So What Do Users See?

Some Users Will See Temporary Inconveniences

For example…

- The recent transition-related disruptions
- Some users (e.g., POP/IMAP users) needed to change their email clients to use new server names (one time)
- When we get migrating web services, we'll need to deal with other places where Darkwing has been historically "hard-coded" (e.g., in URLs)

All Users Will See Lasting Improvements

- Same username/password/email address
- Faster performance
- Less downtime
- All the old familiar services (plus new improved services)

What About Internal to The CC?

Possible disadvantages…

- More complex architecture, with an increased box count (more hosts to patch, backup, monitor, etc.), increased reliance on the network as part of the architecture
- Decreased operational transparency (some things happen on systems to which access is limited)
- Subtle interactions have the potential to occur

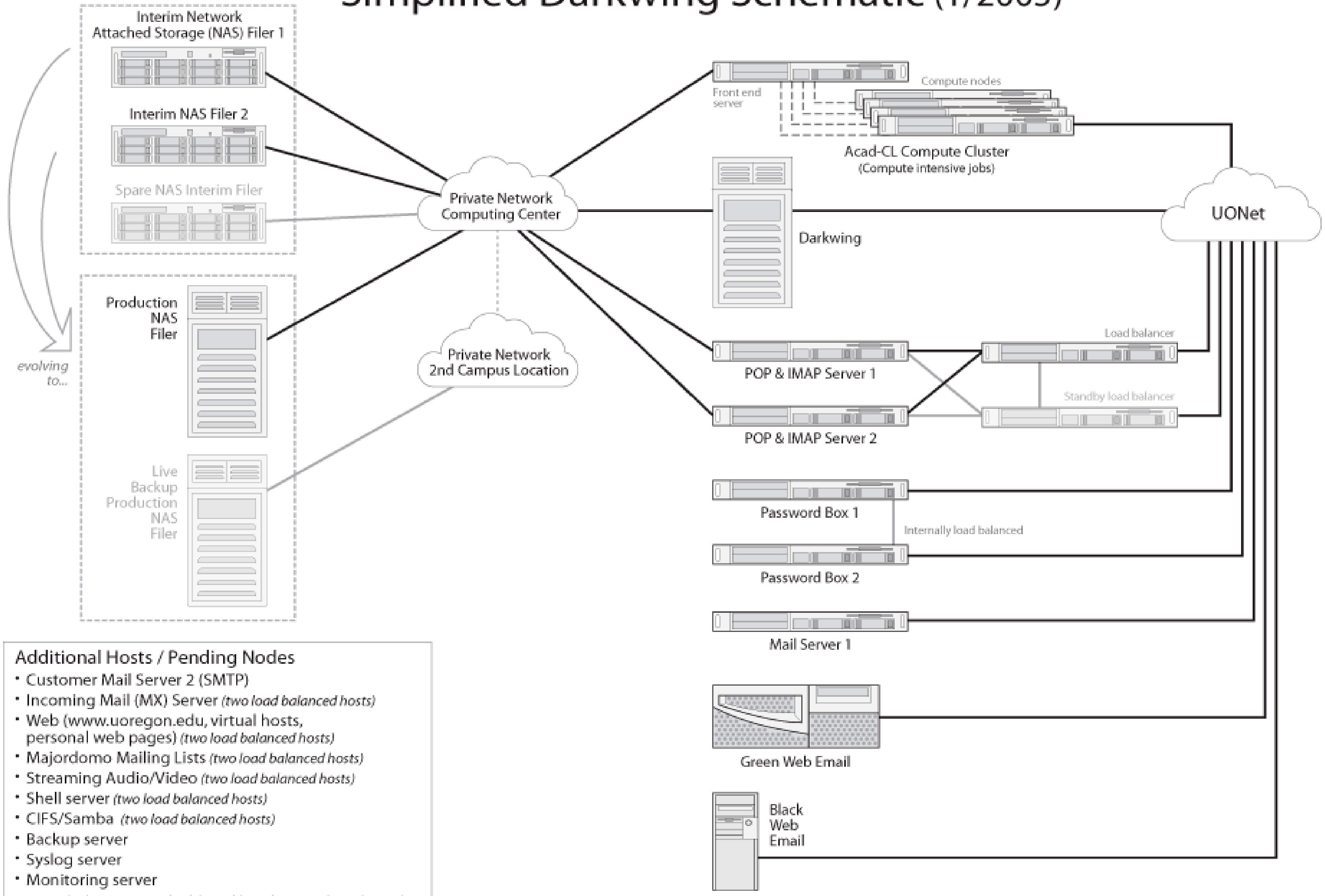
Certain benefits…

- A ton more capacity (which improves stability and gives us flexibility)
- Mainstream OS, mainstream hardware direction
- Less maintenance window time pressure
- Easier sparing (we're using common "building blocks")

Where Are We At Now?

- Some parts are done (e.g., enough to keep Darkwing afloat and uncrushed by the production load it faces)
- ***Lots more is still in progress***, including:
  -- clean up/review what's been done so far
  -- migrate remaining major services (inbound MX mail server, web, mailing lists, etc.)
  -- complete production filer procurement and installation
  -- complete Academic OpenVMS phase out
  -- eliminate black web email; evaluate and introduce third generation web email client
  -- adjust quotas
  -- consolidate Gladstone user base with Darkwing
  -- migrate to LDAP for auth
  -- introduce new services enabled by the new architecture

Simplified Darkwing Schematic (1/2005)



Additional Hosts / Pending Nodes

- Customer Mail Server 2 (SMTP)
- Incoming Mail (MX) Server *(two load balanced hosts)*
- Web (www.uoregon.edu, virtual hosts, personal web pages) *(two load balanced hosts)*
- Majordomo Mailing Lists *(two load balanced hosts)*
- Streaming Audio/Video *(two load balanced hosts)*
- Shell server *(two load balanced hosts)*
- CIFS/Samba *(two load balanced hosts)*
- Backup server
- Syslog server
- Monitoring server

Note: Multiple services may be delivered from the same physical pair of load balanced hosts, depending on system requirements and other factors

Production vs. Interim Filers

- The network attached storage (NAS) filers are really at the heart of the new architecture, albeit functioning in the background in a way that should be transparent to users
- We're currently using interim filers to buy time for the production filer procurement to complete; the technology we're using has proven itself storing data for RouteViews
- The production filers differ from the interim filers in that:
  -- they will begin larger than the interim filer (~8X interim) and can grow to ~40X the capacity of the interim filer
  -- they will have greater throughput
  -- they will have a higher level of vendor support
  -- they will be deployed as a pair of units, one in the CC, another (live backup) at another location on campus
  -- they will offer a broader range of services (e.g., CIFS/Samba access, user accessible backup restorations)

What Does An Interim Filer Look Like?

What’s CIFS/Samba?

- Traditionally Darkwing’s storage has been accessed via NFS (Network File System), a Unix-oriented file sharing protocol.
- CIFS (Common Internet File System) (also often referred to as Samba) is a file sharing protocol that comes out of the Windows environment; deploying it will improve access to Darkwing files from Windows systems, and increase Darkwing’s value to those who work in a Windows-oriented environment.
- N.B.: CIFS service is not available now, it is a *future service*. We’ll share more information with you about it once we complete procurement of the production filers and get closer to offering this service for users.

Thank You!

- Are there any questions?