DreamWorks

ANIMATION SKG®

10,000,000,000 Files Available Anywhere

NFS AT DREAMWORKS ANIMATION SKG

DreamWorks Animation SKG

DreamWorks Animation
Glendale, CA

PDI/DreamWorks
Redwood City, CA

HULA

Who We Are

- Mike Cutler
  - Principle Engineer, focusing on Storage
  - 15 year veteran of PDI/Dreamworks
- Sean Kamath
  - Co-Supervisor of the Enterprise Systems and Services team
  - 4 year veteran of PDI/ Dreamworks

Why we're giving this talk

- Gave a talk at LISA08 giving an overview of Dreamworks.
- Got asked the question: Why use NFS?
- This is (kind of) the response.

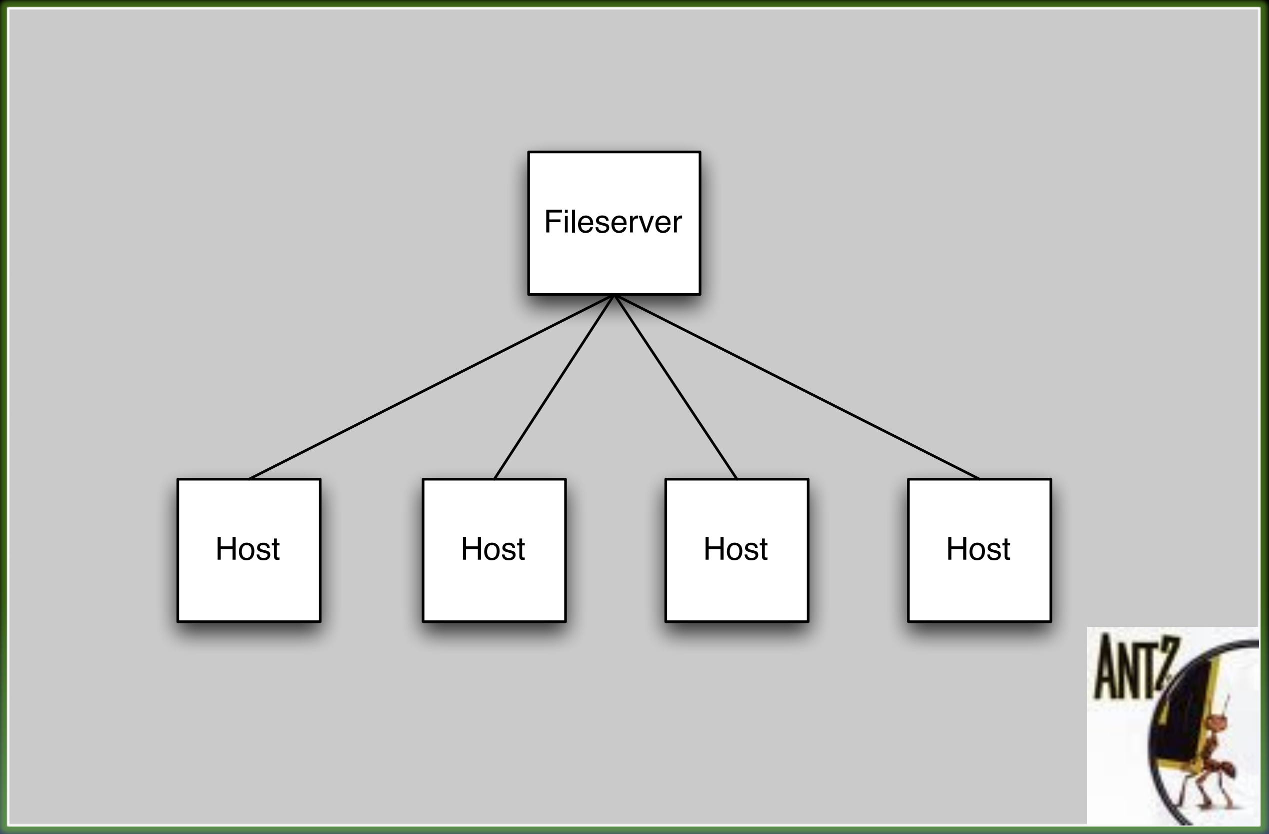
What this talk is not

- Lots of pretty graphs and charts
  - Some lo-tech ones, however!
- Lots of statistics
  - Well, maybe a few near the end
- Specific details of PDI/Dreamworks usage
  - You likely don't have the exact problems we do
  - Its about the process, not just the result
  - We've written an awful lot of our own software
    - Can be tuned to our environment

Why NFS?

- Why not?
- What are the alternatives?
  - FTP/rcp/rdist/etc
    - Create multiple copies
    - Whole file copy from/to
    - Where’s the authoritative copy?
  - sshfs/webdav
    - Just won’t handle the load
  - AFS/DFS/etc
    - Lack of support
    - Reliability
    - Speed

Where we started



Fileserver

Host

Host

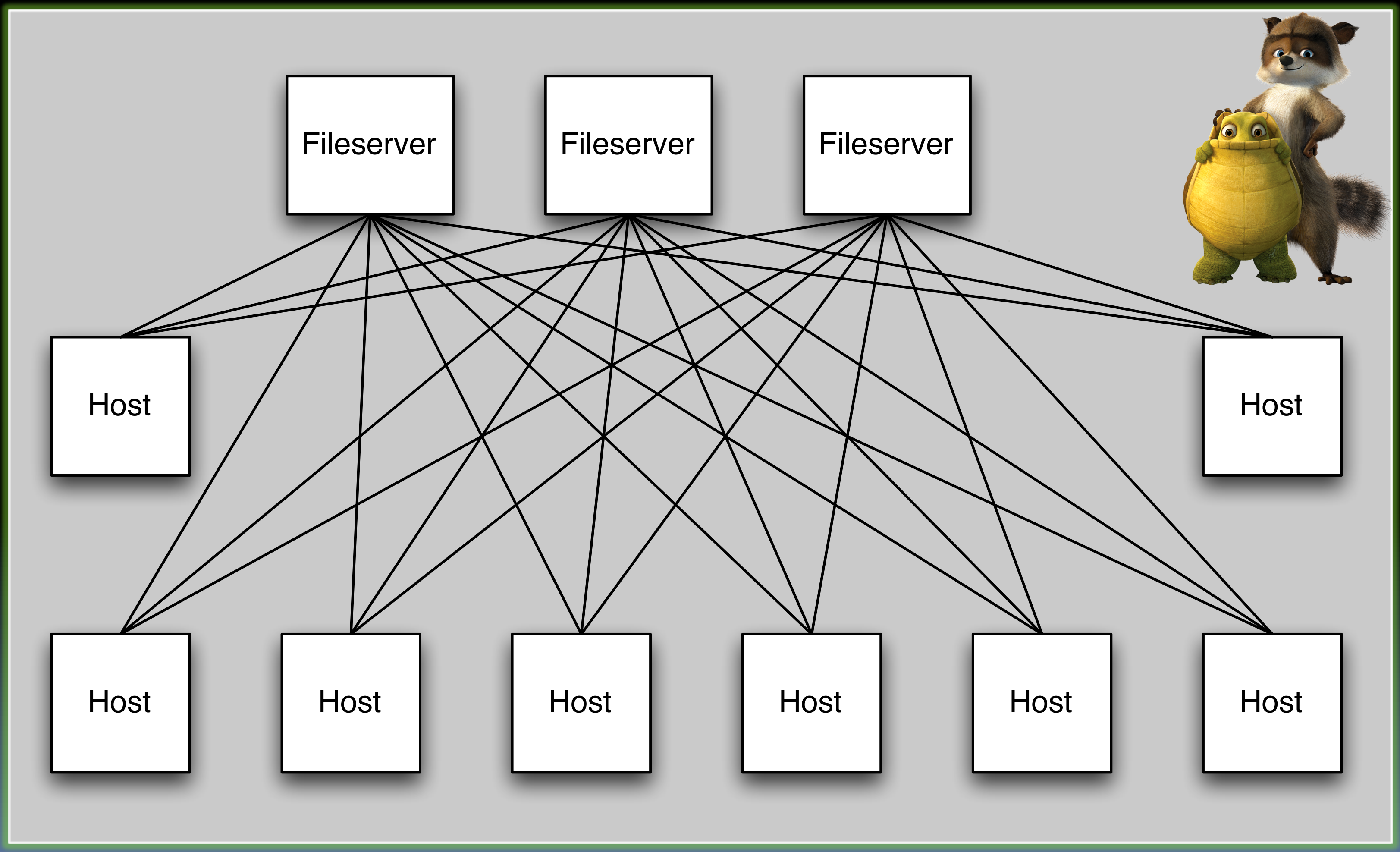
Host

Host

What we did

- 30 years ago
  - 15 hosts
  - 1 fileserver
  - 2-3 filesystems
  - rdist'ed fstab files
- Sophisticated from the beginning!

What happened next



Fileserver

Fileserver

Fileserver

Host

Host

Host

Host

Host

Host

Host

Host

That's not scaling…

- 25 years ago
  - More hosts
  - More fileservers
  - More filesystems
  - A desire to not mount everything everywhere
- The automounter to the rescue!

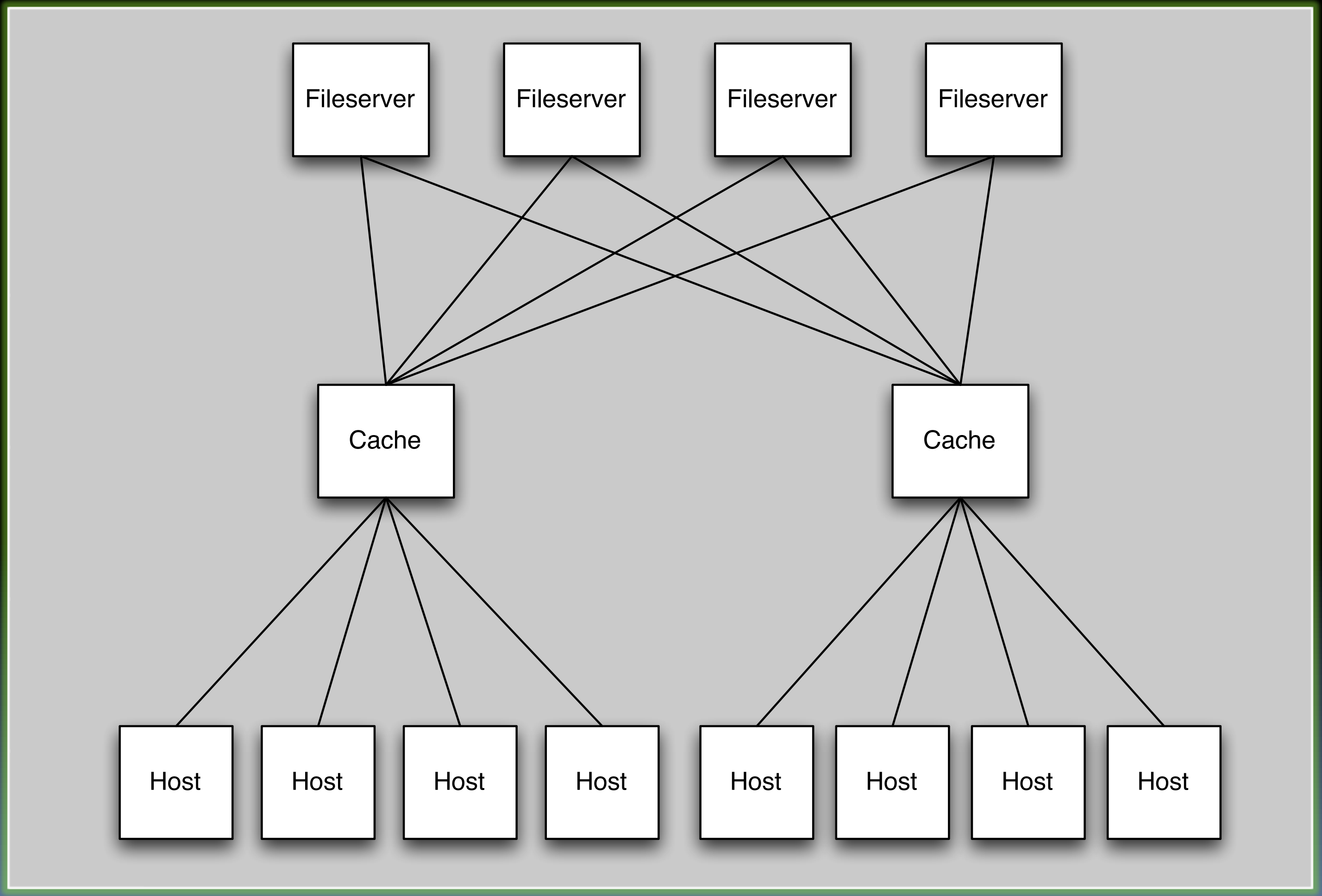
Technologies change. . .

- NIS -> LDAP
  - Lots of benefits
  - More than a few challenges
- SGI IRIX -> Red Hat Linux
- Dedicated NFS fileserver hardware
- Faster LANs
- Bigger/Faster/Better desktops

So the next iteration

- Renderfarm with hundreds of low-cost nodes
  - Significantly increased the NFS load
  - Introduced NFS caching to scale up
- Desktops used for nighttime rendering
  - But you don’t want to cache the desktops

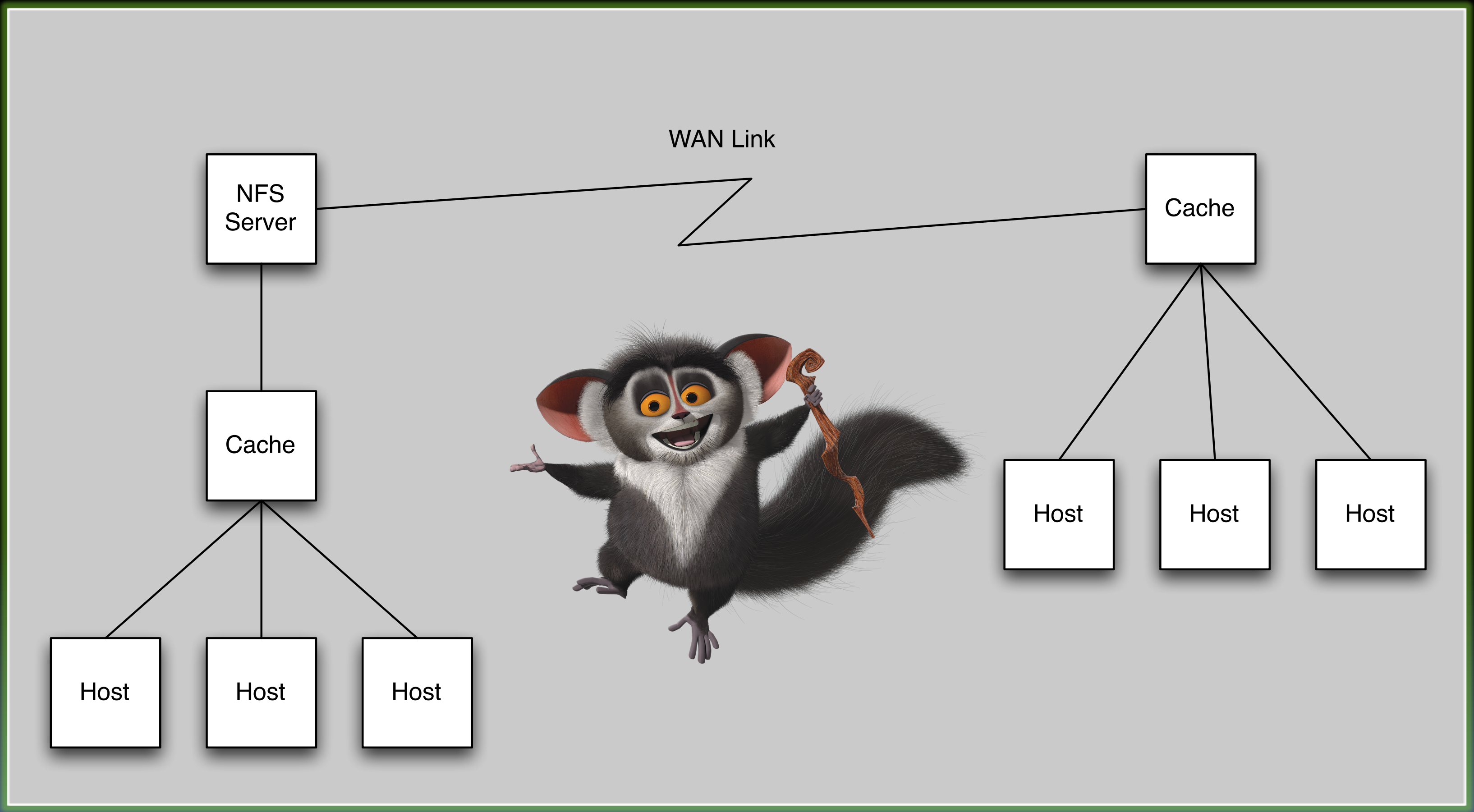
Looks like this



Merging with Dreamworks

- Another site
- More productions
- High-speed cross-site network
  - Reasonably low latency (8ms)
  - Caching helps here too
    - Keeps from saturating WAN

Adding another site

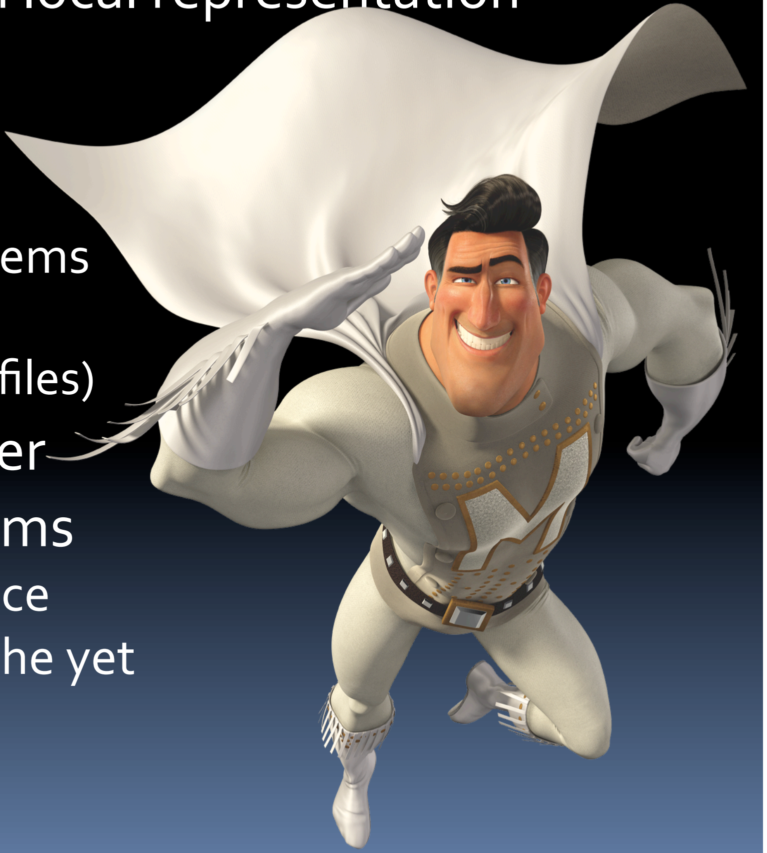


Current NFS Architecture

- Key infrastructure components
  - Global filesystem namespace
  - Data storage hierarchy
    - Not all data is the same!
  - Caching
  - Automounting

Global FS Namespace

- Most filesystems are presented in a global namespace
- Caches used for single source with local representation
- Exceptions
  - Local-only filesystems
    - Site specific applications
  - Location specific versions of filesystems
    - Same location
    - Different data (think configuration files)
- Heavy reliance on the Automounter
- CIFS access to many NFS filesystems
  - We use DFS to mirror NFS namespace
  - We haven’t found a “good” CIFS cache yet
  - Investigating Window’s based NFS



All data is not the same

- Active Productions
  - Many many many small files
  - Many really big files
  - Many files are regenerated multiple times.
- Semi-archived shows
  - Sequels need access to their predecessor's data
- Archived data/shows
  - And we have restored more than a few
- Application and Development software

Data Storage Hierarchy

- Multiple tiers:
  - Fast and Reliable
    - SAS/FibreChannel Drives
    - Really expensive Fileservers
  - Not so fast, but reliable
    - SAS/FibreChannel Drives
    - Less expensive Fileservers
  - Fast, non-critical
    - SATA Drives
    - Less expensive Fileservers
  - Specialized storage

Caching

- Two reasons to cache
  - To provide scalability
  - To provide geographic accessibility
- Introduces a fair amount of management
  - Not complex, but requires a good plan
  - Hard to figure out when to cache desktops

Caching (Scalability)

- Thousands of machines hitting the same filer will melt it.
- Our workload consists of many, many machines accessing the same filesystems.
- Caching works well to protect the back-end filers from falling over
- Can add latency to the filesystem
  - Still faster than an overloaded filer
  - Requires awareness of close-to-open consistency.
    - http://nfs.sourceforge.net/#faq_a8

Caching (Geographic)

- Most data has “location affinity”
  - Not necessarily true in the future
- Caches provide location-based access for repeat requests
- Doesn’t help
  - “first access” latency
  - stat() calls through symlinks
  - Write latency
- Does help
  - Data reads
  - getattr() calls

Automounter Configuration

- All maps kept in LDAP
- Site-local variables to select maps
  - Global variable for site
  - Local variables for cache selection
- We have (US only)
  - ~275 automounter maps
  - ~6500 entries in those maps
  - 8 variables

Example automounter map

- Simple LDAP sub-map

  ```
  subpath  -fstype=autofs
      ldap:nismapname=auto.path-${LOCATION}-
      subpath,ou=automount,dc=...
  ```

- Simple Entry with OS variable

  ```
  pathname  -rw,nocto srv:/vol/volname/topdir/pathelement-
      ${OSNAME}-${ARCH}-${OSREL}
  ```

- Being clever

  ```
  pathname     -rw,intr  srv${CACHE}:/vol/rwc12/rel/map-${OSNAME}-
      ${ARCH}-${OSREL}
  ```

- NOTE
  - NFS caches get CNAMEs (e.g., cache1 is also ‘srvcache1’)
  - CACHE=“cache1”
  - Other variables set in /etc/sysconfig/autofs

Splitting a volume

- Given a entry:

  pathname -rw srv:/vol/volname/pathelement

- When certain directories become "hot", move them to new volume
  - Create a new submap

  pathname –fstype=autofs ldap:nismapname=auto.path-${LOCATION}-pathname,ou=automount,dc=…

  - Populate the new submap:

  / -rw srv:/vol/volname/pathelement

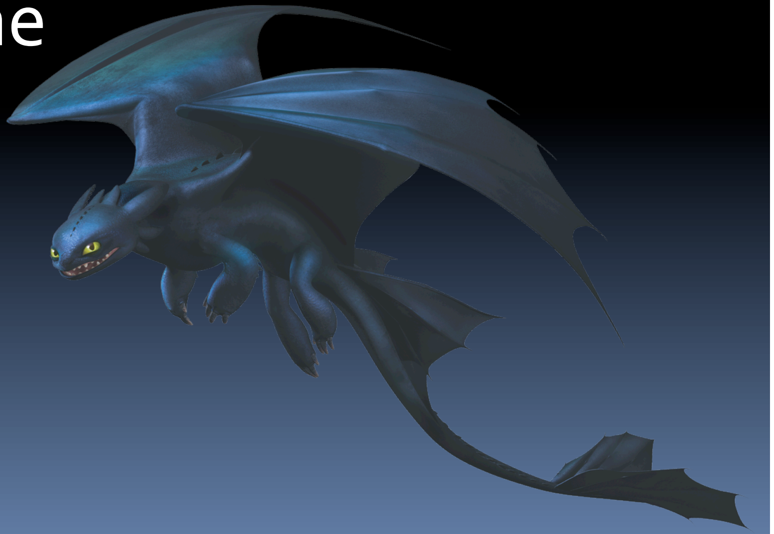
  sub1 –rw srv2:/vol/altvol/sub1path

Where we are now

- 4+ physical sites
- 2500 Users
  - 50/50 RedHat Linux/Windows Desktops
- 1000+ Render Farm Machines
  - Actual number varies almost monthly
  - Exclusively HP BladeSystem based
- 4+ PB of storage
  - 2+PB raw storage/600TB cache/700TB backup
  - >100 Fileservers
    - 75% Primary/25% Caching
- 10G core network
  - Includes connectivity between some sites

Current Capability

- Latest show peaked at 1.4-1.7M NFS ops/sec
- Our LDAP servers handle ~300 farm machines each
- Caching is variable, somewhere between 64-256 machines per cache



A little about our data sets

- Access patterns vary over life of show
  - Significant increase happens at different times
  - Unpredictable due to creative exploration
- Requires redistribution of data (“data moves”)
- Huge growth of “global tmp”

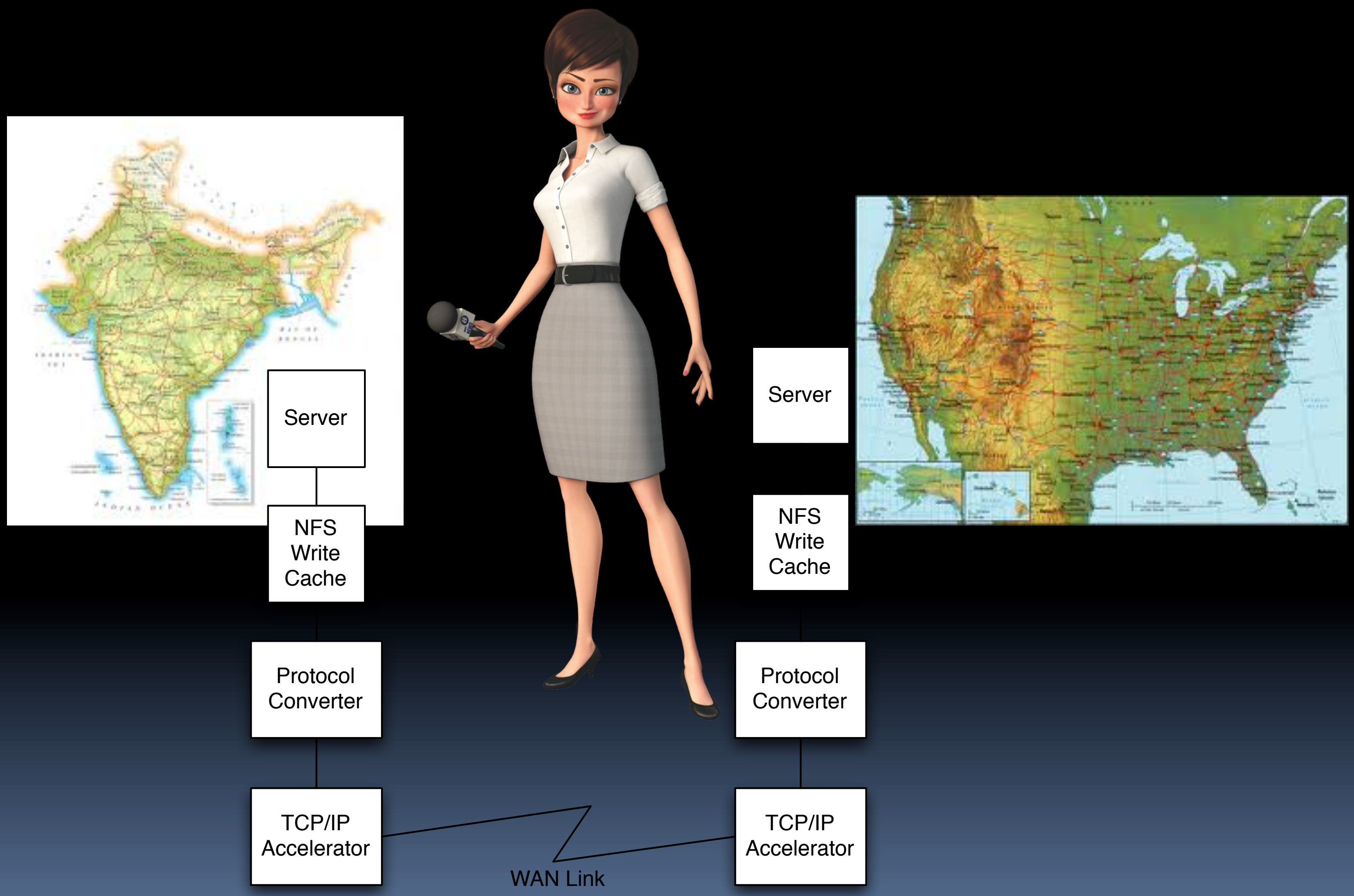
Challenges We Face

- Sometimes data really needs to be at the other site
- Much of our process needs to be automated
    - Have to manually balance the load on caches
        - Adding a cache is nontrivial
            - Setting up the CNAMES and autofs config files
        - Just rebalancing requires autofs config changes
    - Have to manually increase space allocations
    - Have to manually add new shows
        - Many filesystems need to be created
        - Must set up caches as well
- Need to report on disk utilization to departments in real time and via notification system

The future

- Fuzzy data affinity
- Indexed meta-data representation
- information tools to help users
  - 🔴 the data's not here please wait while I fetch it
  - 🟡 I'm getting your data "progress bar maybe"
  - 🟢 Data is ready, launching application
- Different filesystem semantics
- Do NFS v4 delegations do anything for us?
- New technologies to help

For example



Server

NFS Write Cache

Protocol Converter

TCP/IP Accelerator

WAN Link

Server

NFS Write Cache

Protocol Converter

TCP/IP Accelerator

Summary of our solutions

- Global Namespace
  - Important to establish from the beginning!
  - Leverage the Automounter
- NFS IOps offloading
  - Local Caching
- Latency mitigation
  - Cross-site caching
- Working with the developers and artists
  - Being flexible

Q & A

- Mike Cutler – Principle Engineer, PDI/Dreamworks
- Sean Kamath – SysAdmin Supervisor, PDI/Dreamworks