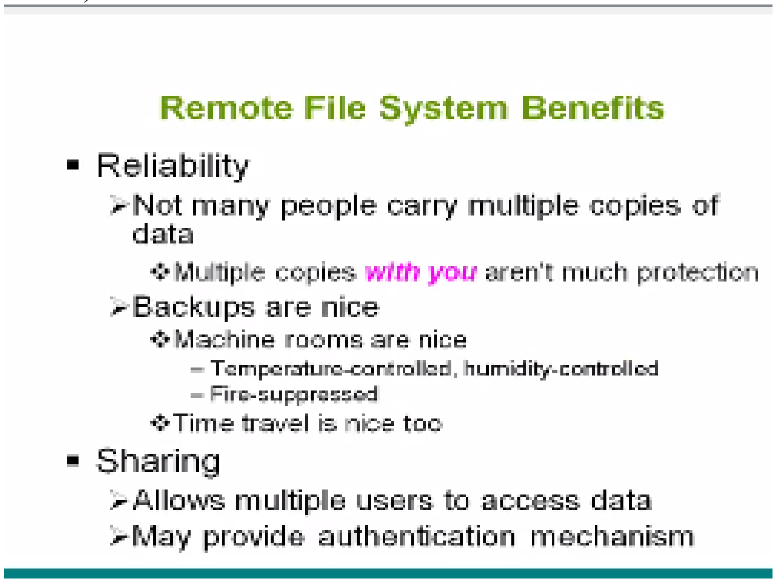
SEER AKADEMI
Linux Networking – Lecture 7
AFS and NIS

Oh okay everyone again welcome to the Linux and the programming languages course the 80s and today we will be continuing our discussion on the Linux networking basics of seven on the Linux networking. We started learning about the distributed file system we learnt about the manifest file system now we will be learning today the AFS for the Andrew file system this is the second half and then we do some comparison between the NFS and the ASF.

And then finally we will look at the NIS which is the network information system and that will pretty much conclude the networking portion of this course. So before I start about start talking about the FS or the Andrew file system let me summarize what we discussed last time we actually started talking about the distributed file system or file system.

(Refer Slide Time: 01:00)



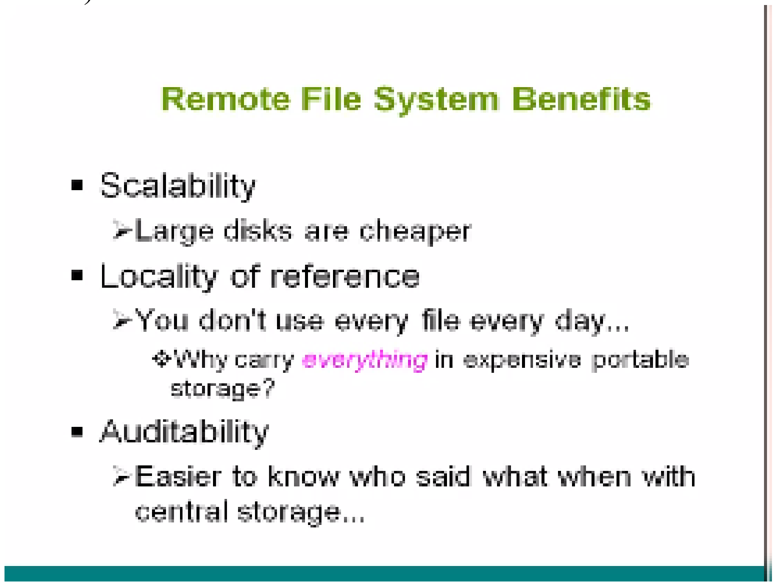
Remote File System Benefits

- Reliability
  - Not many people carry multiple copies of data
    - Multiple copies *with you* aren't much protection
  - Backups are nice
    - Machine rooms are nice
      - Temperature-controlled, humidity-controlled
      - Fire-suppressed
    - Time travel is nice too
- Sharing
  - Allows multiple users to access data
  - May provide authentication mechanism

Some of the benefits of the remote file system one is reliability so again this goes back to like I mean not many people not many people carry multiple copies of the data and if you are carrying multiple copies of the data it does not it is not like a much protection so you need to have like that kind of reliable mechanism and then the second thing is like the backups are very nice and that especially like means the backups work done in a temperature controlled humidity control room the fire suppress facilities and the time travel is nice too meaning.

That you can go back in time and after some of the data things like that it is also like prevent the second aspect of the remote file system is for the sharing it allows the multiple users to active data but you need to provide authentication system security becomes an issue there but in general it allows the sharing so again that is another benefit of the remote pad system we actually talked about several eat terminologies like transparent who think like that so again it all goes back to these benefits or what the remote file system is the other benefit is the scalability again buying large discs are cheaper.

(Refer Slide Time: 02:34)



Remote File System Benefits

- Scalability
  - Large disks are cheaper
- Locality of reference
  - You don't use every file every day...
    - Why carry *everything* in expensive portable storage?
- Auditability
  - Easier to know who said what when with central storage...

So centralize large both for multiple users is a good thing calorie assistance again so this is one thing is we don't use all the files all the time in a single day so why do we need to carry everything in portable storage might as well have it in remote eyes and access only the pice that we need again that is another big benefit of file system and then finally the audit ability is form essentially we can make sure we know who did or who submitted the data what they submitted then with the same storage it is much more well that we can get to that so there are all these benefits with what we talked about in the distributed file system then we looked at the NFS file system.

(Refer Slide Time: 03:42)

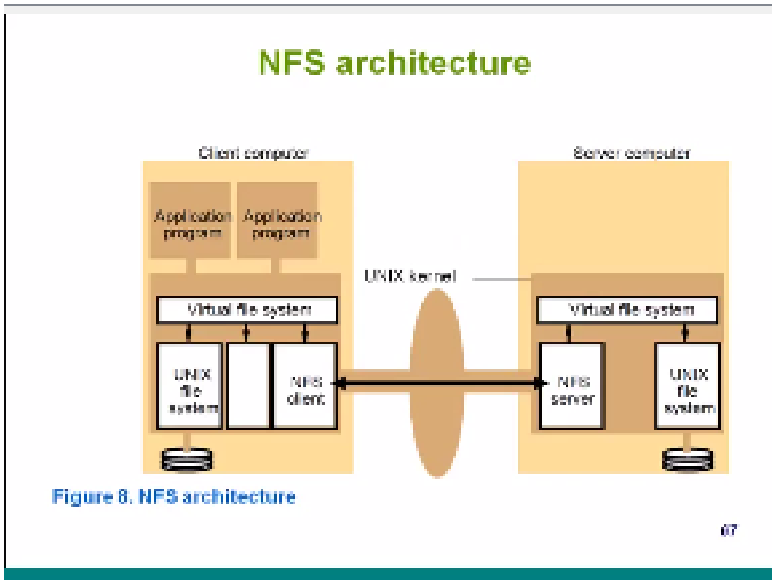

NFS architecture

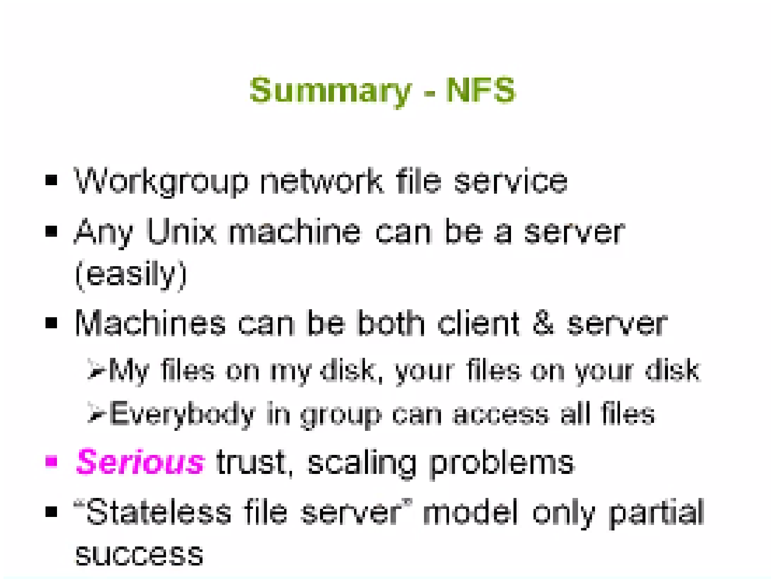
Figure 8. NFS architecture

67

Before the architecture is fairly simple it is a stateless model or the server all the information about that particular files are stored the client-side the client has this NFS client application program accesses form tests like it is a local PI through a virtual file system the virtual file system understands whether it is so part of the local files is the UNIX file system or it is a remote file in this in that case it actually mounts those files as a demo file and then it communicates to the through the network to the NFS server on the server computer and then get that data back into the application program.

So from the application program it still looks at looks like a local pilot sub and it uses this mounting as way to get the files so the NFS client keeps track of what goes on there is a file all those things whereas the server-side is stateless machine so the advantage is that you can add as many times as possible till you do not lose any performance so edge does not degrade by accessing so this is all we saw basically a link-local unit with and Paul works and then while the operation happens now let us look at the summary on NFS the network edge system.

(Refer Slide Time: 05:21)

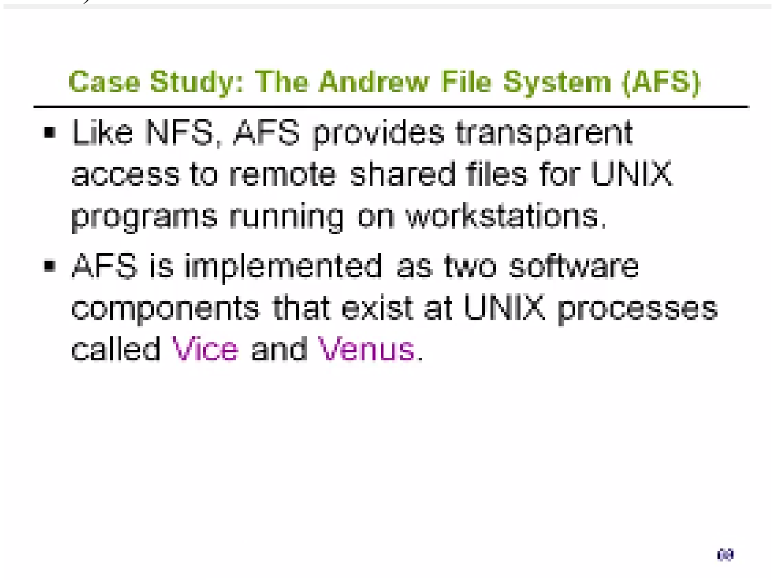

Summary - NFS

- Workgroup network file service
- Any Unix machine can be a server (easily)
- Machines can be both client & server
  - ➢My files on my disk, your files on your disk
  - ➢Everybody in group can access all files
- *Serious* trust, scaling problems
- "Stateless file server" model only partial success

Again it is basically it is a workgroup network file service long end of the protocol and any UNIX machine can become a server and it is fairly easily just easy to set up the server for the NFS any Linux machine the missions can be both the client as the lesser server so some files on my disk and some files on the other side and then everybody in the group can access all these files.

But as we saw there are some serious trust issues some scaling problems that we saw the stateless file server model is goes only to some point basically is only a partial success so now let us look at another file system which is completely different from the NFS or essentially like I mean it uses certain other concepts to build this remote file system let us look at the Andrew file system.

(Refer Slide Time: 06:24)


Case Study: The Andrew File System (AFS)

- Like NFS, AFS provides transparent access to remote shared files for UNIX programs running on workstations.
- AFS is implemented as two software components that exist at UNIX processes called Vice and Venus.

So again one similarity between the AFS and the NFS is that both provide transparent access to the remote shared files or UNIX programs running on both stations and similar to the client-server model what saw in the NFS the AFS is also implemented as two software components that exist at the unit UNIX processes they are called wise and beam so let us look at the picture on this one.
(Refer Slide Time: 07:02)

Case Study: The Andrew File System (AFS)

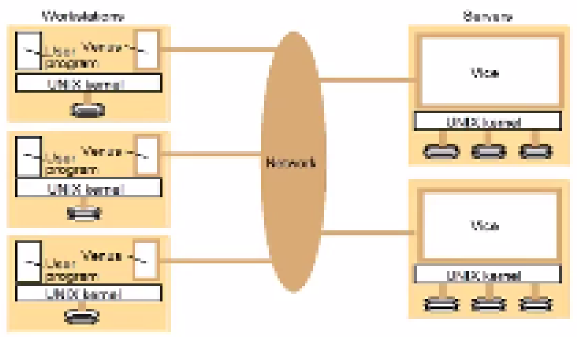


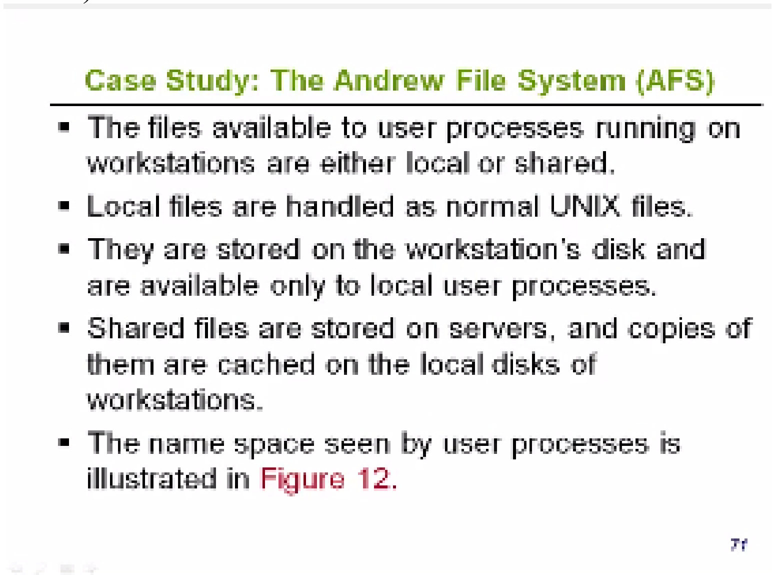
Figure 11. Distribution of processes in the Andrew File System

70

So here you have all these different workstations with its own UNIX kernel there are some user programs it also has what is called this beam is actually a process now in the NF it is we had the NFS server as a separate file system or it was actually like down here the Venus is actually a process that is running on the UNIX kernel just like they use a program so and then all these programs are running.
And then the server side you can see that Vice is also a process or a program that is running on top of the internal and a number of piles so these programs themselves come in a and make sure that what is there in the file system given to the user.
So that is the simple contrast between those pictures and we look at this picture versus this picture here the NFS client was under the file system as a particular file so somehow are somewhat static you can see and then even the NFS server was also another file which convertible to file system again a static and how they are linked together so that is why I like me you have all this some the NFS protocol.
Itself to link excites whereas in a module file system actually the Venus and the white or programs that are being is so their processes so and then the program actually the user program

for me gets through this process to get to the various types of systems and get that intractable on the program that is running you.

(Refer Slide Time: 09:02)



Case Study: The Andrew File System (AFS)

- The files available to user processes running on workstations are either local or shared.
- Local files are handled as normal UNIX files.
- They are stored on the workstation's disk and are available only to local user processes.
- Shared files are stored on servers, and copies of them are cached on the local disks of workstations.
- The name space seen by user processes is illustrated in Figure 12.

So in the Andrews Andrew file system case the files are available to the user processes running on the workstations are either local or share so and the local files are handled as the normal good form they are stored in the workstations desk in this pick self and they are available only to the local user processes the shared files are stored on servers and copies of them are cached in locating this so again it allows that up the caching mechanism the namespace seen by the user process that we will show in the next picture.

(Refer Slide Time: 09:45)

Case Study: The Andrew File System (AFS)

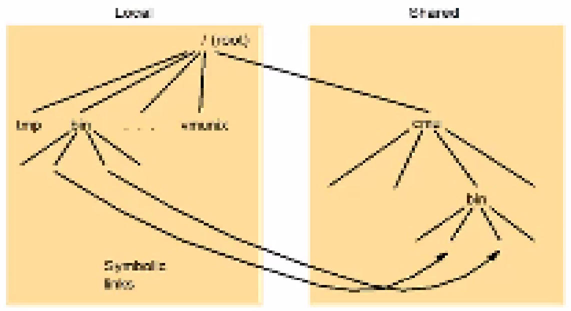
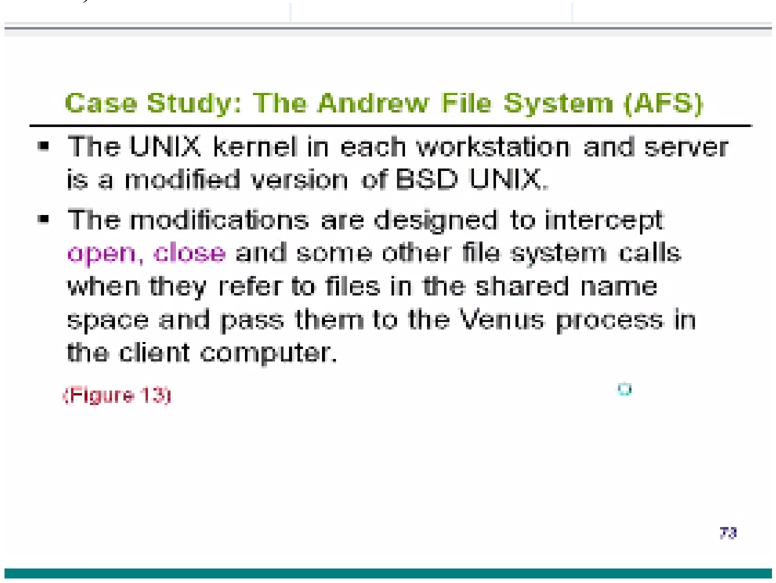


Figure 12. File name space seen by clients of AFS

So again in this one you can see that actually when you have the shared one they are actually shown as symbolic links between the those suffice and that is how it knows that okay that is the remote file.
(Refer Slide Time: 10:11)



Case Study: The Andrew File System (AFS)

- The UNIX kernel in each workstation and server is a modified version of BSD UNIX.
- The modifications are designed to intercept open, close and some other file system calls when they refer to files in the shared name space and pass them to the Venus process in the client computer.

(Figure 13)

73

So the UNIX kernel in Egypt workstation and server is a modified version of the BSD UNIX which is the Burton systems unit and essentially the modifications are designed to intercept the file commands this is essentially the open or closed and some other file system commands as well and then they refer to this shared memory and then pass them to the Venus process in the client computer so essentially like I mean the Venus intercepts these or submits to the Y's and then that Y is posited to the Venus of the client and people add the information and then some mix back to the process that is looking for this information.
(Refer Slide Time: 11:04)

Case Study: The Andrew File System (AFS)

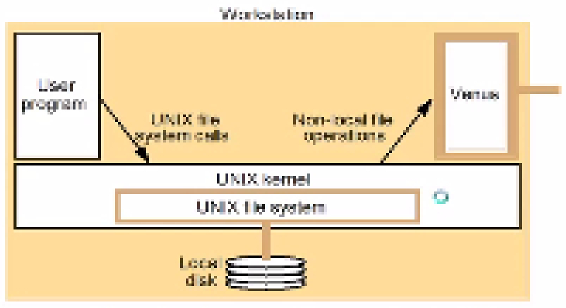


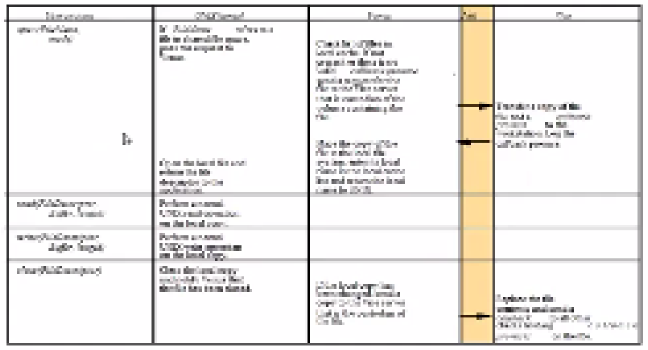
Figure 13. System call interception in AFS

74

So here, a quick setup basically like I mean in the picture of the system called intersection itself so here the user program calls it generates a UNIX file system Bob goes into the UNIX kernel it knows that actually it is not the local disk access but exactly remote finances so those non-local file operations are sent to Venus which actually captures that then communicates to wise and then goes to the server and then gets the that mission brings it back to the user temple user program so let us look at this process in little bit detail.

(Refer Slide Time: 11:51)

Case Study: The Andrew File System (AFS)



[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]		[illegible]
[illegible]	[illegible]			
[illegible]	[illegible]			
[illegible]	[illegible]	[illegible]		[illegible]

Figure 14. implementation of file system calls in AFS

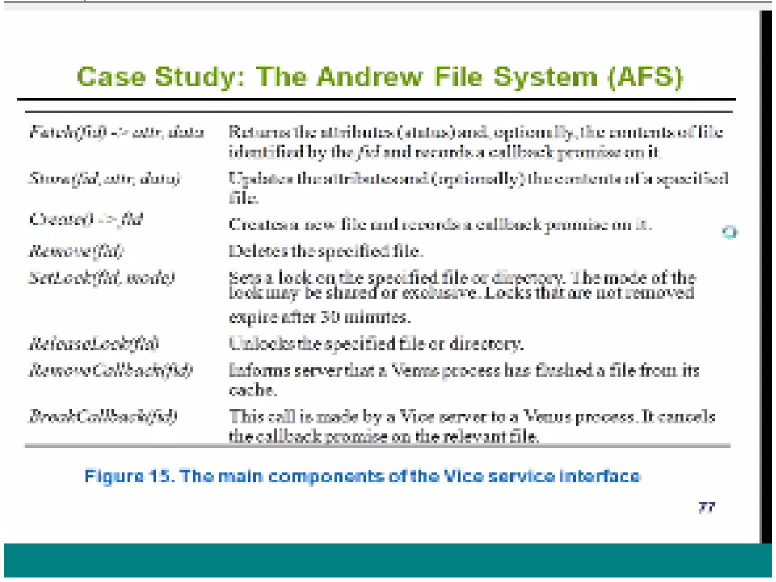
So here, the user process is open and we know this command to open the file name the more once the user process initiates with form command the eunuchs actually has this score which is the if the file name refers to the file in the shared file space as the request to beans otherwise let

us open the local file and return the file to the user process and so the result of this operation will be descriptive.
So now let us look at what happens when this file system or the file that is finding that it is requesting is not in the local file but it is actually a shared file so now if the request is sent to Venus the Venus actually check list of files in the local cache if it is not present or there is no valid all back promise send the request of the file to the white server.
That is a custodian of volume containing this oil so and if it finds that file either fall back from the wise or otherwise basically the cities in the local cache or reset all back promise then the place the copy of the file in the local pawn system enter its name in the local cache list and return the local name back to the unit so it goes back the kernel and then looks back now when the DNS makes that all device essentially wise reads this call and says okay.
I have this file in my PI system so I will transfer that file and the callback point is to the workstation which is requesting this log that callback bonus and then basically existence at that point.
So once that file gets transferred back then this is basically like the copies in the local cache so that now it can be serve so now let us look at a reed pointer essentially so for reading a file descriptor the buffer and the length this is not an issue basically now does not go to Venus otherwise because the venues is already placed or the yeah the Venus already plays that file in the UNIX system as a local point in the cache so now the read operation is just reading that uh-huh then it is a normal operation like this normal read off mission on a localized PI system now if it is a write operation now again it does not go into the venous.
Otherwise but it test does the operation in the local file system so the local copy now the interesting thing is when you close that particular file when you close the file it closes the local copy and it also notifies the venous that the file has been closed so that is when now the venous will take aim and then essentially looks for the local copies change if it is changed then that it will send that copy to the Y server and Y server.
Now puts it in its file system it replaces the 500 and send the call back to all the climbs that are holding that that file that is called backbone is saying that okay this file is modified now you need to modify your local cache file so you can think of this system.
Basically the venues the network and the Y's and as processes running monitoring what is going on in the file and communicating back and forth then the files change or if some other venous processes with the same file it causes that file so think of this there are two systems here one is the user process and user tunnel the other one is the Venus and the white for no systems so and

then they both have a shared local cache section even though here the shared memory is activated in wise you can think of them as updating local cache that essentially legend that is your area where both of these processes can gain access.

So this that may be a better way to understand how the whole thing works in the podcasting the NFS would have seen that actually the processes are pretty much the same the file systems themselves have a separate basically so the there is a localized file system and then there is a remote PI system and basically like the common link is through the mounting and un mounting. So now let us look at some more important components of the wife service.

(Refer Slide Time: 17:32)



Case Study: The Andrew File System (AFS)

Fetch(fid) -> attr, data	Returns the attributes (status) and, optionally, the contents of file identified by the *fid* and records a callback promise on it.
Store(fid, attr, data)	Updates the attributes and (optionally) the contents of a specified file.
Create() -> fid	Creates a new file and records a callback promise on it.
Remove(fid)	Deletes the specified file.
SetLock(fid, mode)	Sets a lock on the specified file or directory. The mode of the lock may be shared or exclusive. Locks that are not removed expire after 30 minutes.
ReleaseLock(fid)	Unlocks the specified file or directory.
RemoveCallback(fid)	Informs server that a Venus process has flushed a file from its cache.
BreakCallback(fid)	This call is made by a Vice server to a Venus process. It cancels the callback promise on the relevant file.

Figure 15. The main components of the Vice service interface

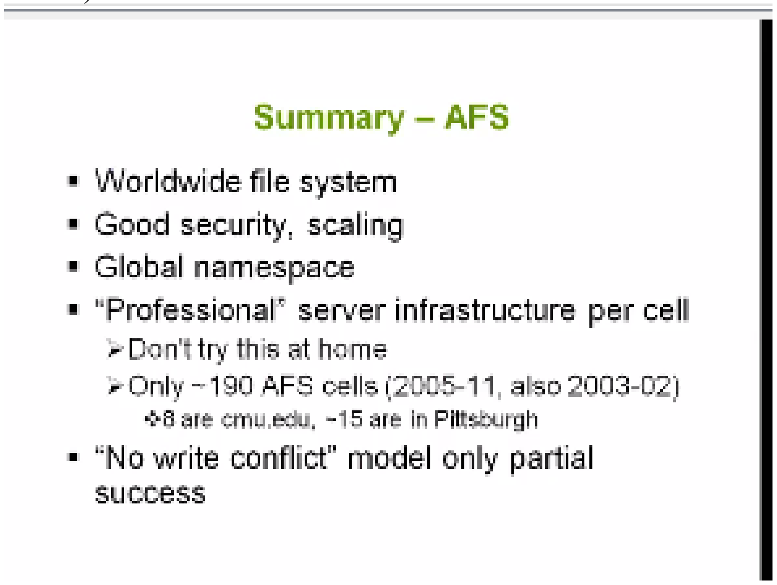
77

Debate this is basically like the server side essentially we can walk through what is shown here so there is a fetch demand which returns an attribute and the data or a given file ID so this essentially like I mean returns the attribute and also the progeny are back promises from on that particular file from that the centralized files so back to the plant back to the Venus program the second command is the store buy lady attribute data this updates the attributes essentially an and the contents of a specific this is.

What the Venus we give that the man once there is a closed file it is basically like an end the store command to add the wife to make sure that the contents are updated then there is another command pole create that returns the file ID this creates a new file and the report with all back promise on it and then sends this pile ID to the requesting process the remove file ad removes a specified file and then we have set lock file ID with mode this sets the lock on the specified PI and the mode of the lock may be shared or exclusive the locks.

That are not removed will expire after 30 minutes and release lock again with the file ID unlocks the specified file or the directory and remove callback or pie lighting also informs the server that the Venus process has flushed by from the cache. And break callback highlighting this is made by the Y server to the various process it cancels a call back from this one at 11:00 time. So in summary the AFS is a worldwide file system.

(Refer Slide Time: 20:02)

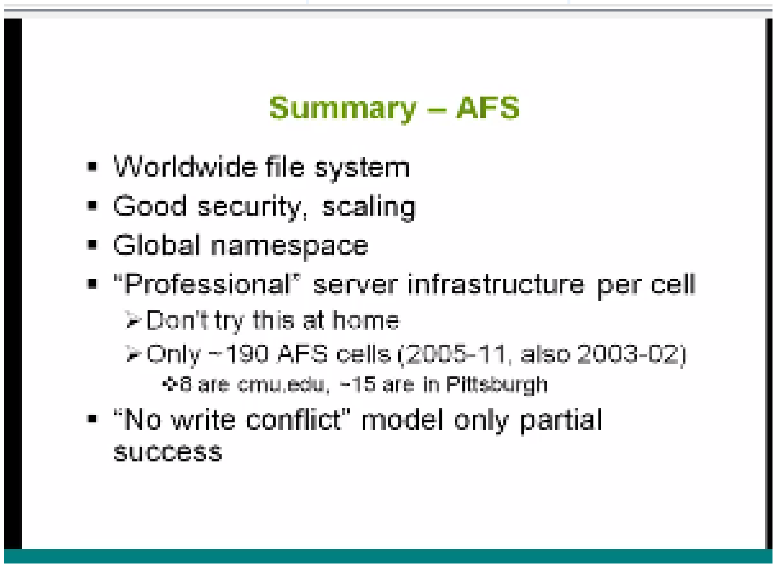


Summary – AFS

- Worldwide file system
- Good security, scaling
- Global namespace
- "Professional" server infrastructure per cell
  - ➢Don't try this at home
  - ➢Only ~190 AFS cells (2005-11, also 2003-02)
    - ❖8 are cmu.edu, ~15 are in Pittsburgh
- "No write conflict" model only partial success

So if you look at this structure here the services hold this file system here. And you can pretty much any other files that are there it is mirrored under that this particular file system so you can almost think of this overall worldwide pie system.

(Refer Slide Time: 20:45)



Summary – AFS

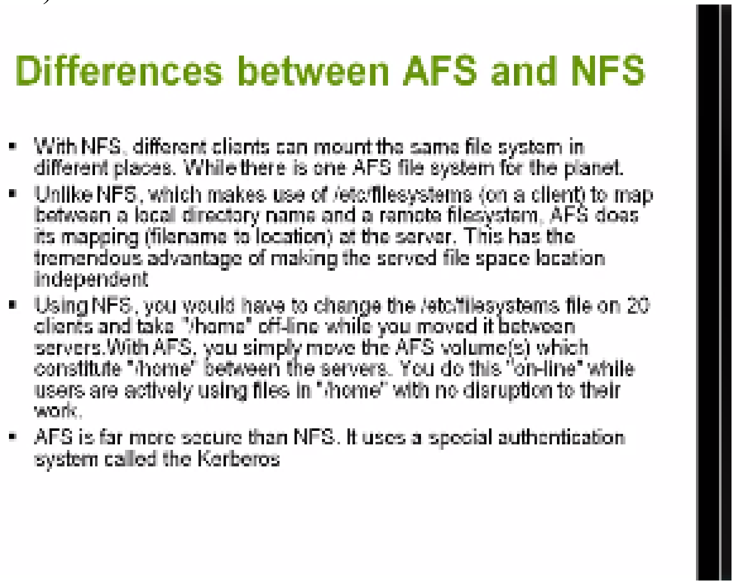
- Worldwide file system
- Good security, scaling
- Global namespace
- "Professional" server infrastructure per cell
  - ➢Don't try this at home
  - ➢Only ~190 AFS cells (2005-11, also 2003-02)
    - ❖8 are cmu.edu, ~15 are in Pittsburgh
- "No write conflict" model only partial success

So NFS may be like one per network here the AFS is just only one that is needed and also like since all these callbacks and all of them are tagged and basically like all the file management is done internally it is it has a very good security as the life good scaling and as you see basically the names themselves they do not have to polite basically it is actually it is a global namespace essentially because the files themselves are copied back and forth into various clients as a management.

So again you can think of this as basically like the server still keeps track of the state of the overall network at all times so this is not a static so as we have seen in the NFS system. And it is a professional server infrastructure Purcell so essentially like I mean you have to be trained on this so you cannot just set it up this like the NFS where in UNIX server can be set up on phone or any UNIX line can be paid upon.

To follow there are only like about 194 for between 2005 and 2011 the eight are in CMU and 15 are in Pittsburgh again in this one again it's the no right conflict essentially like I am kind of that is a protocol that is implied because one and one particular client checks out a file the others may not get the right axis this is only like a partial success because you have like there is a time lag between then the server updates the file. And then it how many things to all the clients to also like change the sign so there are still some issues with respect to how we can create the file and bottle the pipe.

(Refer Slide Time: 22:58)

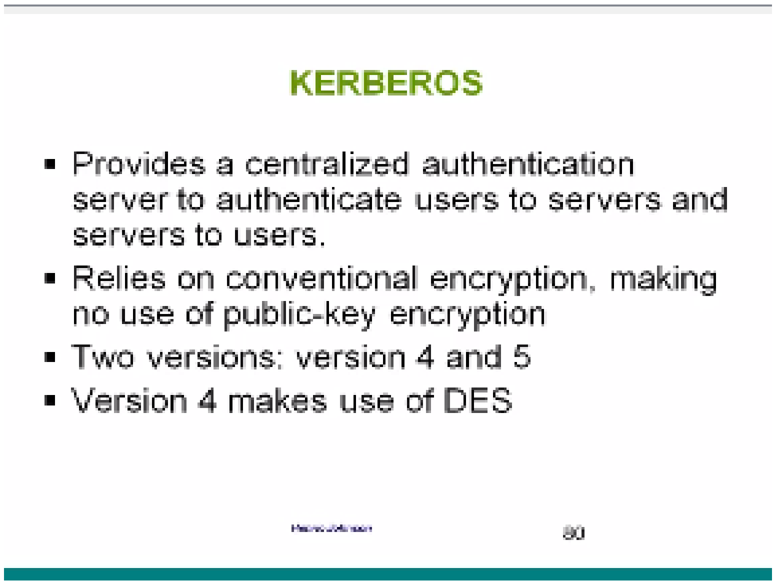


Differences between AFS and NFS

- With NFS, different clients can mount the same file system in different places. While there is one AFS file system for the planet.
- Unlike NFS, which makes use of /etc/filesystems (on a client) to map between a local directory name and a remote filesystem, AFS does its mapping (filename to location) at the server. This has the tremendous advantage of making the served file space location independent
- Using NFS, you would have to change the /etc/filesystems file on 20 clients and take "/home" off-line while you moved it between servers.With AFS, you simply move the AFS volume(s) which constitute "/home" between the servers. You do this "on-line" while users are actively using files in "/home" with no disruption to their work.
- AFS is far more secure than NFS. It uses a special authentication system called the Kerberos

So the key differences between the AFS and NFS essentially with NFS the different clients can mount the same file system in different places there as AFS is just one system or the entire planet so again this goes back to the basic difference is basically like client every client see is the file system has its own unique file system in a NSS amendment truly by replicating the file monk points whereas in an AFS system.

The centralized server stores all the files basically the processes themselves communicate to this sense log over in order to gather those files so that is one of the key differences between the annotation with this and the unlike NFS the NFS actually uses the etcetera.
Five systems as a mount point and this is where there is a map between the local directory name and the remote PI system the AFS does all the mapping at the server so this is a tremendous advantage this so holds tremendous advantage of making the third file system patient completely independent so everything is controlled by the server for the clients have seldom you need anything to do with that anything to do changing a namespace so there is no like namespace collision and also the this file space location is also completely independent.
Because of server is the one box temples they are the particular grossing where the particular file goes into so you can actually move it around and still you would not see in issues the server still has a mapping of their the types of whereas in an NFS which is the next one basically the then when you change a file essentially the in an NFS.
So essentially you need to change the EDC file systems file on the 20 clients even if you are taking like this taking the slash home of time or while you move it between the service with the AFS you have to just simply move the FS volume this constitutes a slash home between the service you do this online.
While the others are actively using time and without no disruption to their work because the changing whatever you are doing you are going to send a call back notice to those circles kind we are the Y's and the Venus to disk change the notification basically so in fact when you're changing a file you do not even have to notify the Y server because in their access form to either close the file or open a new file in that directory the mapping is already in place in the server to go to that particular thing um and then the regarding the security is far more secure than NFS it uses this special authentication called Kerberos. We talked about this Kerberos briefly in the last lecture.
(Refer Slide Time: 27:07)



KERBEROS

- Provides a centralized authentication server to authenticate users to servers and servers to users.
- Relies on conventional encryption, making no use of public-key encryption
- Two versions: version 4 and 5
- Version 4 makes use of DES

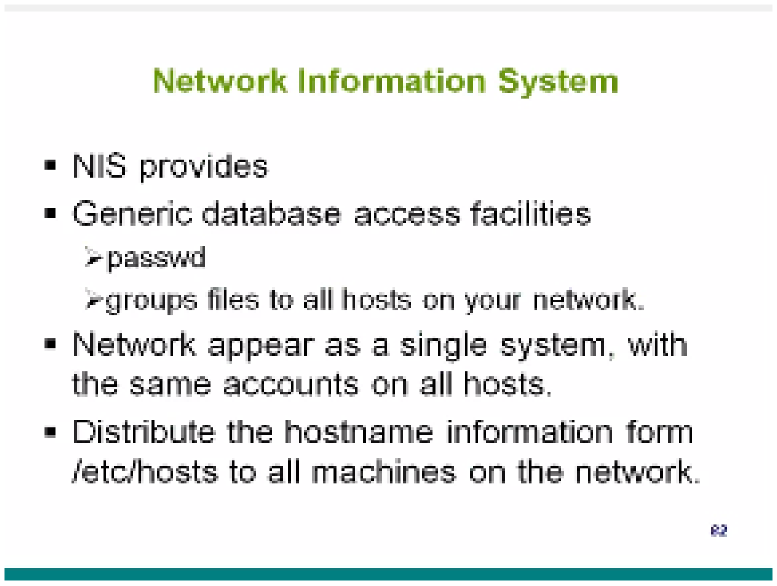
80

I just wanted to give you some more details on books some of you may have already known both rows Kerberos actually provides a centralized authentication server to authenticate users to service in service to users and this relies on the conventional encryption and it does not make use of the public key encryption so there are some additional things with there are some other infection services that it uses so the basic idea behind Kerberos is essentially there is a server where you send the just so the as a client or a user the user data is sent to that server.

Which keeps track of which users are authorized to use the particular business so and then it sends back the authentication information so once you receive the authentication moving with that certificate you can go into a server or for a given service and while you are doing that even the server is also authenticated both games so by the same tubules process which says that ok that server is authenticated to actually now receive your service request and deliver the service.

So the server gets the same thing so this is what you see like I mean in most local websites when they say basically the certificate is expired you want to redo the certificate essentially like in the search here. It is the authentication forwarded by the third or so with that we conclude the distributed file systems now let us look at the network information system which is essentially an application of all the things that we talked about which is both the DSS DNS everything put together.

(Refer Slide Time: 28:58)


Network Information System

- NIS provides
- Generic database access facilities
  - ➢passwd
  - ➢groups files to all hosts on your network.
- Network appear as a single system, with the same accounts on all hosts.
- Distribute the hostname information form /etc/hosts to all machines on the network.

62

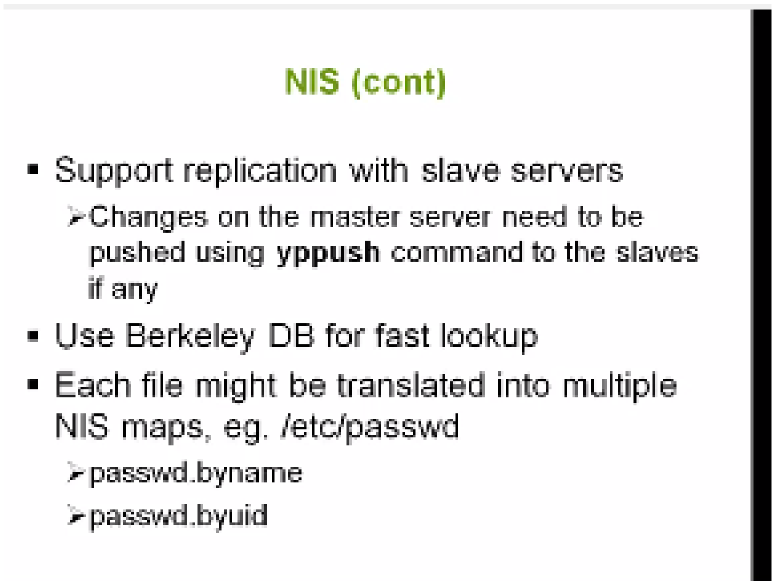
So a network information system provides a generic database access facilities such as password and group files to all the posts in a given network so that the network will appear as a single system with the same account all the hosts so you log in from any of the holes so today like I mean you have a machine and you are logging in as yourself in the machine tomorrow you go into your friends office and you try to log in as your I your authentication information what if the computer does not let you access and it happens differently since all the computers are connected in the same network.

Do not you think that that also needs to take in your information then give you the access I think that is what the NIS voids so it basically distributes the hostname information from flashy TV goes to all the machines on them so from for a given user the client everything including the network and all going to put access points everything appears as a single network so it does not matter whether it is a subnet or possibly local subnet and model subnets are grouped together the NIS provides single active or uniform access across these people.

The network information structure is based on remote procedure calls this we saw already it comprises server at client-side library and several administrative tools originally the NIS was started by warm Sun as Yellow Pages system so it used to be all yellow pages but then British Telecom had exclusively it was a white form this the yellow pages.

So Sun had discontinued the UC go for the Olympics so then they named it has no so that is the origin of the network information system but the YP is still there inaudible commands such as YT serve might be buying etc these are the basic commands for building in a token system. So the NIH supports replication with these lasers.

(Refer Slide Time: 31:43)

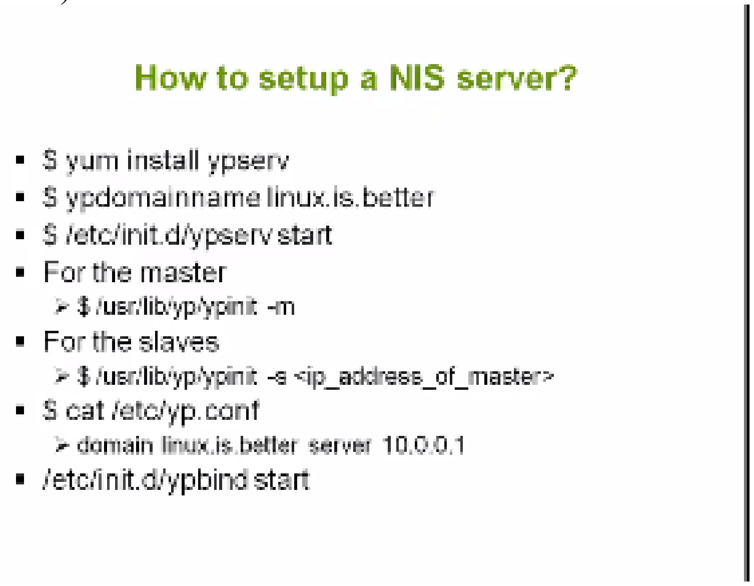


NIS (cont)

- Support replication with slave servers
  - Changes on the master server need to be pushed using **yppush** command to the slaves if any
- Use Berkeley DB for fast lookup
- Each file might be translated into multiple NIS maps, eg. /etc/passwd
  - passwd.byname
  - passwd.byuid

We thought that big box for it does and essentially like any the changes in NASA server need to be pushed using YP pushed man to the slave end and if you this the Berkeley database or passing companies all the very first thing is into the database based system or authenticating password and the group bikes and each file might be translated in the multiple set of information system box so this is using the UPC password so you can have like password by name password or user ID things like that so various forms of authentication information and we provide it. So now let us look at how to set up an NIS server.

(Refer Slide Time: 32:45)

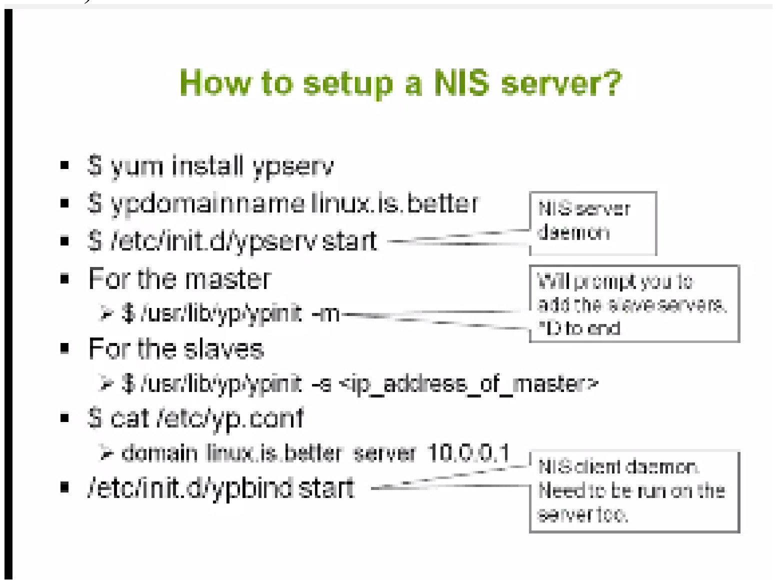


How to setup a NIS server?

- $ yum install ypserv
- $ ypdomainname linux.is.better
- $ /etc/init.d/ypserv start
- For the master
  - $ /usr/lib/yp/ypinit -m
- For the slaves
  - $ /usr/lib/yp/ypinit -s <ip_address_of_master>
- $ cat /etc/yp.conf
  - domain linux.is.better server 10.0.0.1
- /etc/init.d/ypbind start

So essentially like I am going to use these to mount yum install YP so then we provide a weepy domain name with YP domain name then access better then we start the YP services within the fashion TP slash index dot d and if you want to set up a master process then we say basically YP

unit - M and for the slaves we say basically so I be in it - s and provide masters so I get this then we essentially like read from the white kid or Quantico essentially ligament that is where we say to go pass with domain.
So this is what the outcome message in is on where we set up this domain Linux is better and basically like and now you have the alarm but the configuration define. Now you can actually start the process which is using the YT bind command and we just say basically like to bind start and then start the overall boss.
(Refer Slide Time: 34:12)

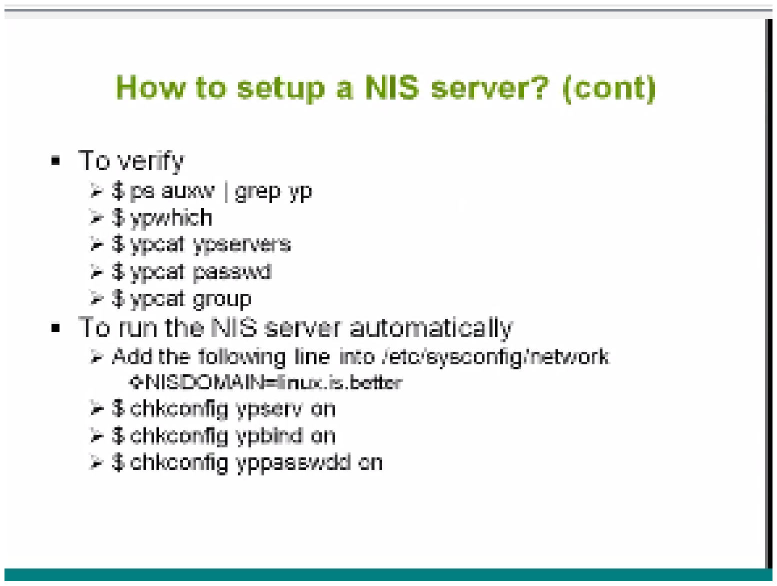


How to setup a NIS server?

- $ yum install ypserv
- $ ypdomainname linux.is.better
- $ /etc/init.d/ypserv start
- For the master
  - $ /usr/lib/yp/ypinit -m
- For the slaves
  - $ /usr/lib/yp/ypinit -s <ip_address_of_master>
- $ cat /etc/yp.conf
  - domain linux.is.better server 10.0.0.1
- /etc/init.d/ypbind start

NIS server daemon
Will prompt you to add the slave servers. ^D to end
NIS client daemon. Need to be run on the server too.

So the weepy serve start essentially starts the NASL the wind and even when you view the master command that will prompt you to add the save command and control D is to end and then finally the NA s client demon itself is started by the wiping bind command and this needs to be run both in the server side as well as the contact.
(Refer Slide Time: 34:44)



How to setup a NIS server? (cont)

- To verify
  - $ ps auxw | grep yp
  - $ ypwhich
  - $ ypcat ypservers
  - $ ypcat passwd
  - $ ypcat group
- To run the NIS server automatically
  - Add the following line into /etc/sysconfig/network
    - NISDOMAIN=linux.is.better
  - $ chkconfig ypserv on
  - $ chkconfig ypbind on
  - $ chkconfig yppasswdd on

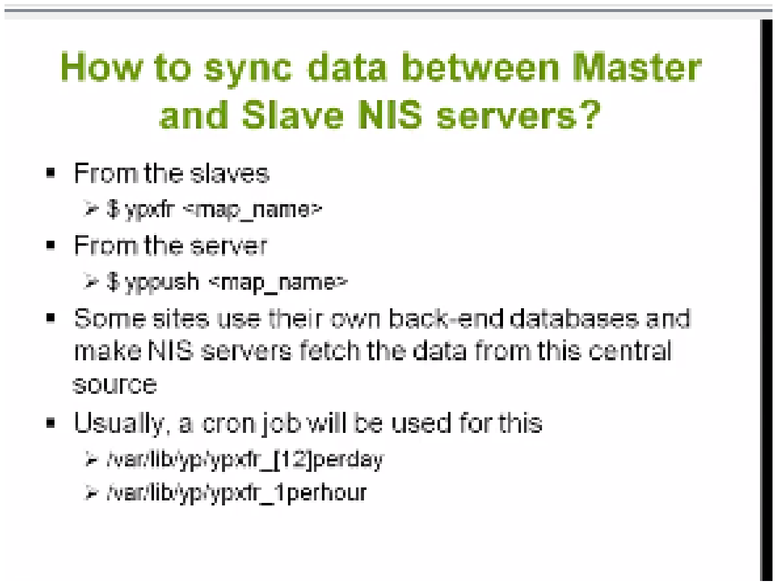
And then to verify use the pH - a UA W which is essentially like we want to make sure that the YP processes are running and then YP which is another command Y peak at Y P servers essentially and then Y P get password and maybe get group. So these are all essentially commands to get the information regarding the inner NIS system.

I would like you to actually use some of these comments in your own workstation and see what is the result and also you can also type in man or each of these commands essentially the YP which by the at etcetera to find out more information on out increments and to run the NA server automatically.

The slash ETC slash Visconti slash network is to be updated with the domain so basically this or DNA is the main equal to domain name and then the you give these commands check on why this is on check on TTIP bind go again instead of the register do the binding and then essentially like if want to do like the other services to be offered so for example the YP password is one of them it is you can do the checked on pick a password make that also on.

So that you can give the users to do internet and IP CH SH is also again the one this is a change shall command if you want to provide the users with that facility then you can basically turn on that is also you so now the other thing is how do you think data between the master and the slave in the Emperors. So from the server we can just provide the weepy transfer command.

(Refer Slide Time: 37:01)



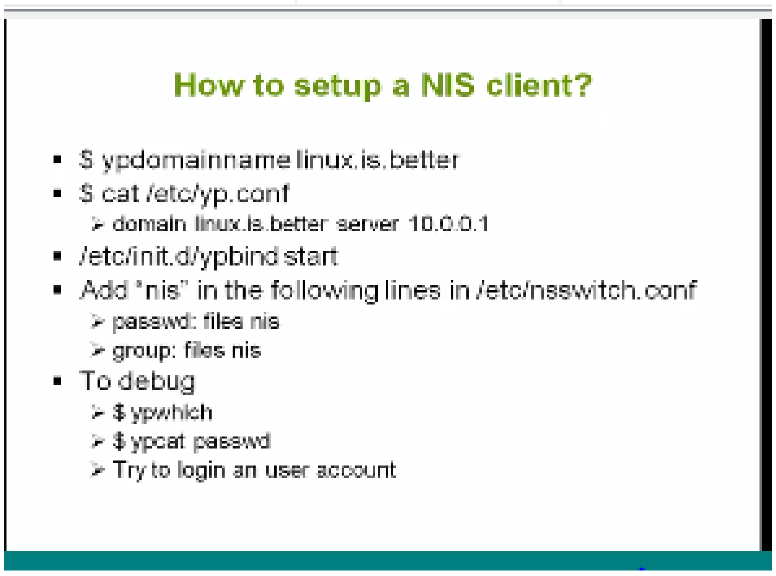
How to sync data between Master and Slave NIS servers?

- From the slaves
  - $ ypxfr <map_name>
- From the server
  - $ yppush <map_name>
- Some sites use their own back-end databases and make NIS servers fetch the data from this central source
- Usually, a cron job will be used for this
  - /var/lib/yp/ypxfr_[12]perday
  - /var/lib/yp/ypxfr_1perhour

With the math team from the server side so from the slave side we just provide the weepy transfer form and from the server side we do the weepy push man we already saw this way to be pushed essentially to push the dimension back to go the all the slaves so any new map changes are pushed to these two commands on either from the slaves to the server or service to the sleigh and some sites use their back-end databases and make a nice service fetch the data from this central.

So we have a some back-end database is already defined and then it basically like the NASL fetch the data from the post essentially like they use some cron job to do this. One for example this is a transfer twelve per day which is basically we allow basically transfers that or we can use a con job which is transfer wiper control one per day per hour which is every hour it transfers them you so here essentially like I mean this is the sequence of demand.

(Refer Slide Time: 38:37)

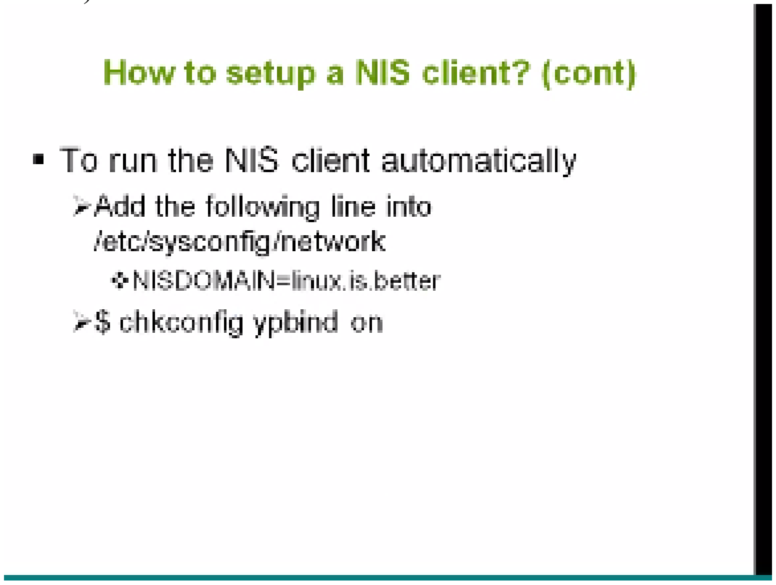


How to setup a NIS client?

- $ ypdomainname linux.is.better
- $ cat /etc/yp.conf
  - domain linux.is.better server 10.0.0.1
- /etc/init.d/ypbind start
- Add "nis" in the following lines in /etc/nsswitch.conf
  - passwd: files nis
  - group: files nis
- To debug
  - $ ypwhich
  - $ ypcat passwd
  - Try to login an user account

That you can try it in your machines if you set up the weepy domain name and called in Linux dot e dot better and then essentially you configure the look for the comfit file it is why P dot Pontiac and then you basically do the YP fine start then the N is to the each of these lines the password and the group essentially and then use the various commands to debug with the YP which we pick at password and then try to login into a user account.

(Refer Slide Time: 39:25)

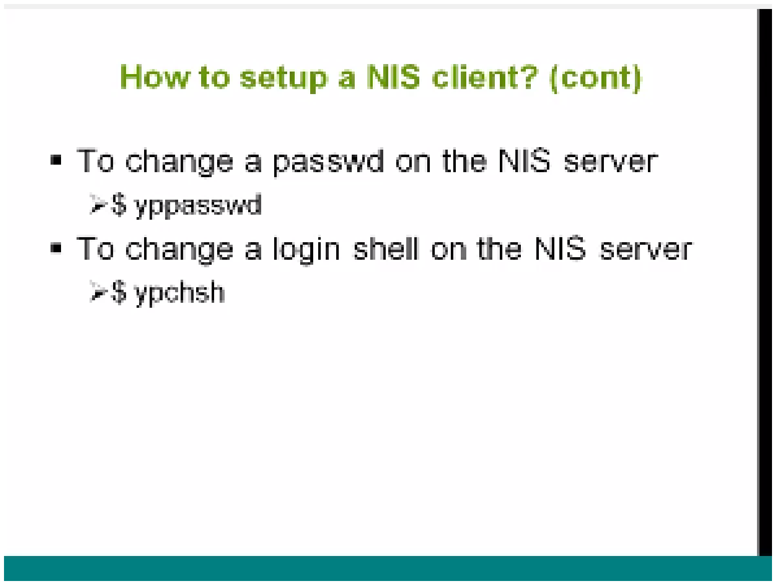


How to setup a NIS client? (cont)

- To run the NIS client automatically
  - Add the following line into /etc/sysconfig/network
    - NISDOMAIN=linux.is.better
  - $ chkconfig ypbind on

And then this is something that we thought they should really talk on automatically.

(Refer Slide Time: 39:39)

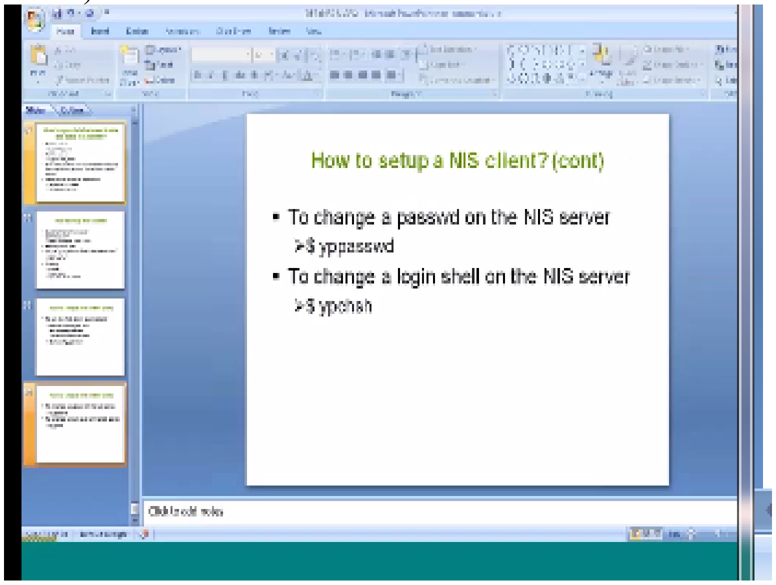


How to setup a NIS client? (cont)

- To change a passwd on the NIS server
  - $ yppasswd
- To change a login shell on the NIS server
  - $ ypchsh

And then to Kane the password on the NA server it wills YP password the man similar to password. So this essentially changes the password across all the clan so that that is true that why the transfer process it pushes into the schools and then to change the login shell change shell command with YP in front so this actually concludes the topic for today.

(Refer Slide Time: 40:14)



How to setup a NIS client? (cont)

- To change a passwd on the NIS server
  - $ yppasswd
- To change a login shell on the NIS server
  - $ ypchsh

Click to add notes

I think we will begin the programming classes the next one so today in the server in the Linux networking class we saw two items one is the Android platform we now know what is an enterprise system and out the contrast between an to PI system and the NFS that we learnt last class and then also we learnt about how to set up in s and also like what are the key points about NIS and why do we need it again thank you very much for attending and thank you um see you in the next class Thanks you.

Hello sir yeah hey Cindy okay so done with the last I said yeah so today there are only like about 40 minutes way around 41 minutes okay, so let me think let me send you the sides I the day in today yes like within four to five hours okay so starting the base always right yeah okay Shane it set up by the way like so by sending that particular thing descending my contact number to you go okay.

So like if anything happen like you can just call me on this all the way with it sure and you know what then even next class let us start at 10:30 when we can sample same thing right now we go with like Wednesday or Sunday.