The Logistical File System: A Network File System Designed for Scalable Resource Sharing

Alex Bassi, Micah Beck, Erika Fuentes, Terry Moore, James S. Plank
Logistical Computing and Internetworking Laboratory
University of Tennessee

Abstract —This paper describes the technologies and design ideas that underlie the *Logistical File Systems* (*LoFS*). LoFS is close to a traditional distributed file system in structure and in the class of operations it supports, but it is designed to preserve the easy deployability and scalability across the administrative boundaries that have been the pillars of the Web's success. The leading idea behind the design of LoFS is that in order to implement a real file system that nonetheless preserves the strengths of the Internet of model resource sharing, one has to apply that model to the storage resources needed to implement file system operations, so that they are exposed and shareable to the global network. Systems that do not expose the underlying resources used to implement file system operations can implement remote access to file system operations, but they cannot distribute many important functions of the file system itself.

1. Introduction

After more than a decade of unprecedented growth, the World Wide Web has transformed the landscape of networked information systems. It has easily dominated the arena of wide area information sharing that seemed, at the beginning of the 90's, to be the natural domain of more capable and mature file sharing technologies, such as the Andrew File System [1]. This development was all the more surprising since comparison shows that the weaknesses of the Web's design (e.g. limited *ad hoc* caching capabilities, little or no security, lack of scalability in important dimensions, tendencies to performance degradation, etc.) were significant, and many of its strengths (e.g. free distribution, intuitive user interface) could be (and have been) duplicated by more capable competitors [2]. But what could not have been duplicated, and the key, we believe, to the Web's unparalleled success, was the *Internet model of resource sharing* at the base of its design

As evidenced in structure the IP stack, the Internet Model was created to facilitate the sharing of network bandwidth for the purpose universal communication among an international community of indefinite size, and is therefore designed to be as open (i.e. lightly controlled) and easy to use as possible. It's very weaknesses, according to some established criteria, have had the compensating virtue of making it easy to deploy and highly scalable across the administrative boundaries of traditional systems. The Web builds on this same foundation. The fundamental contribution of the Web over previous networked information systems was not the notion of the URL as a filename with globally uniform semantics (AFS and other distributed file systems have that feature), but the use of such names in hyperlinks that were globally valid across *administrative boundaries*. When the Web was introduced users found that if they could put up a Web server (and this was very easy to do), *anyone* could link to the files they wanted to share and vice versa. It seemed as if the Web's "fuzzy pointers" could point anywhere.

The universality and generality that this model to resource sharing achieves has a power that is undeniable. But in order to achieve it, the Web architecture makes a series of compromises that from the traditional systems point of view seem drastic:

1. The default semantics of URLs are that they represent unprotected, read-only data. The protection mechanisms that have been introduced are based on passwords and certificates, and so are not inherently limited in administrative scope.
2. Local interpretation of URLs exposes a specific portion of the host directory structure to the network.
3. The Web defines a restricted set of operations on URLs that can be implemented in a globally scalable network.

The thesis of this paper is that a different set of design choices can be made that results in a distributed file system that can scale up to the global Internet, but without accepting the compromises accepted by the Web. We believe that the file system that results, which we call the *Logistical File System* (*LoFS*) is closertotraditionaldistributedfilesystemsins tructureandintheclassofoperationsitsupports ,but preservestheeasydeployabilityandscalabilityac rosstheadministrativeboundariesthathavebeent he pillarsoftheWeb'ssuccess.Wecallthisfilesys tem"logistical"becauseitbuildsonourworkin *logisticalnetworking* ,whichisdescribedelsewhere[3].

TheleadingideabehindthedesignofLoFSisthat inordertoimplementarealfilesystemthat nonethelesspreservesthestrengthsoftheInternet model,onehastoapplytheInternetmodelto *the resources(primarilystorage)neededtoimplementf ilesystemoperations,sothattheyareexposedand shareabletotheglobalnetwork* .Systemsthatdonotexposetheunderlyingresourc esusedtoimplement filesystemoperationscanimplementremoteaccess tofilesystemoperations,buttheycannotdistribu te manyimportantfunctionsofthefilesystemitself.

Accordingly,indesigningLoFSwehavefolloweda" bottomup"designphilosophythattodayismore familiarindesignofnetworkprotocolstacksthan inthedesignofoperatingsystems.Atthebottomo fthe "networkstoragestack"isthe *InternetBackplaneProtocol (IBP)*,whichisamechanismcreatedtoenable sharingofexposedstorageresourcesacrossthenet workontheInternetparadigm.Since,aswedescrib e below,IBPusesamodelofstoragewithweaksemant ics,inordertosupportthekindofstrongfile abstractionthatLoFSrequires,wehavedevelopeda datastructurerepresents *aggregate*storageresources andallowsustolayerafileabstractionwithstro ngsemanticpropertiesontopofaweakunderlying storageresourcethatdoesnotgenerallyprovideth em.Wecallthisdatastructurethe *exNode*becauseitis analogoustotheUnix *inode*,butscopedforthewideareanetwork.Finally,th etoplayerofthenetwork storagestackisLoFS,whichisdesignedtobealo g-basedfilesystemthatcanleveragethepowerof Internetresourcesharingthatthelowerlayersmak eavailable.Thediscussionbelowfollowsthesame bottomupdesignasthetechnologybeingdescribed, explainingeachlayerinturnandprovidingdetail sof applicationsandexperienceswiththesetechnologie s.

2. Background:TheInternetProtocolandtheInternetBackpl aneProtocol

TheuniquecapabilitiesofLoFSwillderivefromth efoundationofexposedresourcesharingonwhichi t builds,i.e.fromthe InternetBackplaneProtocol.IBPisamechanismdev elopedforthepurposeof sharingstorageresourcesacrossnetworksrangingf romrack-mountedclustersinasinglemachineroom toglobalnetworks.[3-5]Toapproximatetheopenn essoftheInternetparadigmforthecaseofstorag e, thedesignofIBPparallelskeyaspectsofthedesi gnofIP,inparticularIPdatagramdelivery.This service isbasedonpacketdeliveryatthelinklevel,but withmorepowerfulandabstractfeaturesthatallow itto scaleglobally.Itsleadingfeatureistheindepend enceofIPdatagramsfromtheattributesofthepar ticular linklayer,whichisestablishedasfollows:

- Aggregationoflinklayerpacketsmasksitslimits onpacketsize;
- Faultdetectionwithasingle,simplefailuremodel (faultydatagramsaredropped)masksthe varietyofdifferentfailuremodes;
- Globaladdressingmasksthedifferencebetweenloca lareanetworkaddressingschemesand masksthelocalnetwork'sreconfiguration.

ThishigherlevelofabstractionallowsauniformI Pmodeltobeappliedtonetworkresourcesglobally , anditiscrucialtocreatingthemostimportantdi fferencebetweenlinklayerpacketdeliveryandIP datagramservice.Namely,

> *AnyparticipantinaroutedIPnetworkcanmakeuse ofanylinklayerconnectioninthe networkregardlessofwhoownsit.Routersaggregat eindividuallinklayerconnections tocreateaglobalcommunicationservice.*

ThisIP-basedaggregationoflocallyprovisioned,l inklayerresourcesforthecommonpurposeof universalconnectivityconstitutestheformofshar ingthathasmadetheInternetthefoundationfora globalinformationinfrastructure.

IBPisdesignedtoenablethesharingofstoragere sourceswithinacommunityinmuchthesamemanner. JustasIPisamoreabstractservicebasedonlink -layerdatagramdelivery,IBPisamoreabstractse rvice basedonblocksofdata(ondisk,tapeorothermed ia)thataremanagedas“bytearrays.”The independenceofIBPbytearraysfromtheattributes oftheparticular *accesslayer* (whichisourtermfor storageserviceatthelocallevel)isestablished asfollows:

- Aggregationofaccesslayerblocksmasksthefixed blocksize;
- Faultdetectionwithaverysimplefailuremodel(f aultybytearraysarediscarded)masksthe varietyofdifferentfailuremodes;
- GlobaladdressingbasedonglobalIPaddressesmask sthedifferencebetweenaccesslayer addressingschemes.

Thishigherlevelofabstractionallowsauniform I BPmodeltobeappliedtostorageresourcesgloball y, andthisisessentialtocreatingthemostimportan tdifferencebetweenaccesslayerblockstorageand IBP bytearrayservice:

> *AnyparticipantinanIBPnetworkcanmakeuseofa nyaccesslayerstorageresourceinthe networkregardlessofwhoownsit.TheuseofIPn etworkingtoaccessIBPstorageresources createsaglobalstorageservice.*

WhateverthestrengthsofthisapplicationoftheI Pparadigm,however,itleadsdirectlytotwoprobl ems. First,inthecaseofstorage,thechronicvulnerab ilityofIPnetworkstoDenialofUse(DoU)attacks is greatlyamplified.Thefreesharingofcommunicatio nwithinaroutedIPnetworkleaveseverylocal networkopentobeingoverwhelmedbytrafficfromt hewideareanetwork,andconsequentlyopentothe unfortunatepossibilityofDoUfromthenetwork.Wh ileDoUattacksintheInternetcanbedetectedand corrected,theycannotbeeffectivelyavoided.Yet thisproblemisnotdebilitatingfortworeasons: onthe onehand,eachdatagramsentoveralinkusesonly atinyportionofthecapacityofthatlink,sotha tDoU attacksrequireconstantsendingfrommultiplesour ces;ontheotherhand,monopolizingremote communicationresourcescannotprofittheattacker inanyway,itcanonlyharmthevictim. Unfortunatelyneitherofthesefactorsholdtruefo raccesslayerstorageresources.Onceadatablock is writtentoastoragemedium,itoccupiesthatporti onofthemediumuntilitisdeallocated,sonocon stant sendingisrequired.Moreoveritisclearthatmon opolizingremotestorageresourcescanbevery profitableforanattackerandhisapplications.

Thesecondproblemwithsharingstoragenetwork-sty leisthattheusualdefinitionofastorageservic eis basedonprocessor-attachedstorage,andsoitincl udesstrongsemantics(near-perfectreliabilityand availability)thataredifficulttoimplementinth ewideareanetwork.Evenin“storagearea”orloc alarea networks,thesestrongsemanticscanbedifficultt oimplementandareacommoncauseoferror conditions.Whenextendedtothewidearea,itbeco mesimpossibletosupportsuchstrongguaranteesfo r storageaccess.

Wehaveaddressedbothoftheseissuesthroughspec ialcharacteristicsofthewayIBPallocatesstorag e:

- *AllocationsofstorageinIBPcanbetimelimited.* Whentheleaseonanallocationexpires,the storageresourcecanbereusedandalldatastructu resassociatedwithitcanbedeleted.AnIBP allocationcanberefusedbyastorageresourcein responsetoover-allocation,muchasrouterscan droppackets,andsuch“admissiondecisions”canbe basedonbothsizeandduration.Forcing timelimitsputstransienceintostorageallocation ,givingitsomeofthefluidityofdatagram delivery.
- *ThesemanticsofIBPstorageallocationareweaker thanthetypicalstorageservice* .Chosento modelstorageaccessedoverthenetwork,itisassu medthatanIBPstorageresourcecanbe transientlyunavailable.Sincetheuserofremotes torageresourcesisdependingonsomany uncontrolledremotevariables,itmaynecessaryto assumethatstoragecanbepermanentlylost. Thus, *IBPisa“besteffort”storageservice* .Toencouragethesharingofidleresources,IBP even

supports“volatile”storageallocationsemantics,whereallocatedstoragecanberevokedatany
time.Inallcasessuchweaksemanticsmeanthatthelevelofservicemustbecharacterized
statistically.

BecauseofIBP’slimitationsonthesizeanddurationofallocationanditsweakallocationsemantics, IBP
doesnotdirectlyimplementreliablestorageabstractionssuchasconventionalfiles.Insteadthesemustbe
builtontopofIBPusingtechniquessuchasredundantstorage,muchasTCPbuildsonIP’sunreliable
datagramdeliveryinordertoprovidereliabletransport.

3. TheInternetBackplaneProtocol

3.1 TheIBPAPI

TheIBPAPIisinmanyrespectstypicalofnetworkfilesystems,withcallsforallocation,accessand
managementassummarizedinTable1belowanddiscussedinsomedetailintheAPIdocumentation[5].
TheuniqueaspectsofIBParereflectedmostdirectlyintheIBP_allocatecall.

Inmostconventionalfilesystems,filecreationentailsthecreationofanentryinafiledirectory undera
client-suppliedname.Thedirectoryentryrepresentstheabilitytoindefinitelyallocatedataspacethrough
writeoperations,generallyuptosomelimitimposedeitherbythesystemoronaper-userbasis,and to
laterreadfromthatdataspace.Insomespecializedfilesystems,fileattributessuchasphysicallayoutor
stagingpolicybetweendiskandtapecanbespecified[6-8].

TheIBP_allocatecalldiffersinanumberofways. Firstofall,itismuchlikeatypicalmemoryallocation
operationsuchastheClibrarymallocinthatitallocateswritablestoragespacebutdoesnotcreate a
visibledirectoryentry.Itreturnsasetofcapabilitiesthatcanbeusedasanopaquecredentialforlater
read,writeandmanagementoperationsontheallocatedspacefromanyInternet-connectedclient.Most
important,itallowsthespecificationofattributesthatmodelanumberofusesforstorageotherthan
conventionalfilespace.

- Anumberofdifferentwritesemanticsareavailable:append,truncate,FIFOqueue,circular
queue.Notethatthereiscurrentlynosupportforoverwritingotherthanbytruncatingtheentire
filetozerolength,asthisrequiresexternalsynchronizationwhensharedamongmultiplewriters.
Thesewritesemanticssupportnon-fileapplicationssuchastheimplementationofUnix-like
pipesinthenetwork.Toreflectthisgenerality,werefertothespaceallocatedasabytearrayto
reflectthatfactthatitismoregeneralthaneitherafileoracommunicationbuffer.
- Allocationscanbeweakenedinanumberofwaysreflectingamorelightweightapproachto
permanenceandreliabilitythanistypicalfornetworkfilesystems.Inparticular,allocationsmay
betime-limited,representingaleasethatexpiresatsomeknownpointinthefuture,andtheymay
bevolatile,meaningthattheyrepresentanallocationoffreespacethattheservercanrevokeat
anypointinthefuture.Itisintendedthatbyallowingdepotstograntweakerallocationswhile
remainingwithintheboundsofcorrectfunctioning, IBPwillenablethesharingofresourcesthat
wouldotherwisebeheldforprivateuseonly.

TheotherIBPAPIcallsfallintotwogroups:thosethatoperateonanIBPallocation,andonethat
operatesonanIBPdepot.

1. **Operationsonallocations** . **`IBP_allocate`**returnsasetofthreecapabilities:areadcapability,
awritecapabilityandamanagecapability.Eachofthesearerequiredfordifferentsubsequent
APIcalls.
    a. **`IBP_store, IBP_load, IBP_copy, IBP_mcopy.`** The **`IBP_store`**and **`IBP_load`**
    aresyncrhonousreadandwriteoperationsthatreturnwhentherequesteddatatransfer
    hasoccurredorwhenaspecifiedtimeoutexpires. Theytakeawriteandreadcapability
    asargumentrespectively. **`IBP_copy`**allowsathird-partycopybetweendepotswithout
    requiringthatthedataberetrievedbytheclient, andtakesbothareadandawrite

capabilityasarguments.The **IBP_mcopy**callmodelsgeneralizedpoint-to-multipoint communication.Ittakesasargumentasingleread capability(source)andasetofwrite capabilities(destinations)aswellasan“operatio n”parameterandasetofoperation-specificarguments.Theseoperationsareimplement edbydepot“plug-in”modules called“datamovers”(seesection3)andareintend edtosupportflexibleexplorationof newandnon-standardwaysoftransferringdatabetw eenendpoints[9].

   b. **IBP_manage** takesthemanagecapabilityasanargument,andimp lementsincrementand decrementoperationsontworeferencecountsmainta inedforeachallocations:aread countandawritecount.Ifthewritecountreache szero,theallocationcanbetreatedas read-only,andifthereadcountreacheszero,the allocationcanbedeleted.Thiscallis alsousedtoquerythestateofanallocation,and torequestmodificationofsomeofits basiccharacteristics(suchasextendingthelease) .

2. **Depotmanagement** callsrequirenocapability,butcanbeprotectedb ypassword.
   a. **IBP_status**hastwosub-commands: **inquire**and **change**. **Inquire**allowsaclient toqueryadepotaboutitstotalstableandvolatil estorage,theamountofbothstorage categoriesused,andthemaximumallowedd.duration. **Change**allowstheclienttochange theseparameters.

StorageManagement	DataTransfer	DepotManagement
IBP_allocate, IBP_manage	IBP_store, IBP_load IBP_copy, IBP_mcopy	IBP_status

Table 1

3.2 IBPImplementation

Depot—ThemainIBPdepotarchitecturegoalsareflexib ility,reliabilityandperformance.Thecurrent implementation(1.0)ismulti-threadedforperforma nce,withapoolofthreadscreatedatboottime. The codebaseissharedbetweenUnix/Linux/OSXandWin 32versions,withOS-specificI/Ocalls encapsulatedinalibrarythathastwoimplementati ons(win-libandunix-lib).

ClientLibrary —TheIBPClientLibrary,offeredinafewdifferen tversionsandsystems,wasdesigned tobeflexible,toeasetheimplementationoffutur echangestoboththeAPIandtheprotocol,tobev ery maintainablecodeandtobeextremelyrobustandfa ult-tolerant.Thelibraryissepeartedintotwo differentmodules:theAPI2PModuleandtheCommuni cationModule(ComModule).API2Ptranslates theAPIcommandintoCommunicationUnits,whichare abstractdatatypesthatspecifythe communicationanditscharacteristics(direction,s emanticsofthemessage,themessageitselforthe expectedmessage).Then,theComModuleallowsthee xecutionofthecommunication.Noanalysisofthe messageismadeatthislevel,theAPI2Pmodulebei ngresponsibletointerpretthemessageandtotake theappropriateaction.Thisdesignallowseasych angestotheAPI(asit’sseenasasequenceof communicationunits)andtotheprotocol.Ontopof it,thecommunicationmodule,beingcompletely independent,willbere-usedinfuturedepotimplem entations,cuttingdevelopingtime.

Protocol—Thecurrentversionofourprotocolisaverydir ectencodingoftheAPIasanarchitecture-independentRCPusingaper-callTCPconnection.A smorechallengingrequirementsarise,weare consideringaredesignofthisprotocol,perhapsus ingsomeoftheprotocolencapsulationtools,such as BXXP,thatarecurrentlyunderdiscussionforstand ardizationwiththeIETF[10].

Testing—Aflexibletestlanguageandinterpreterweredev elopedtoenableextensivetestingand performancemeasurementofthedepotbybothdevelo persandusers.Thesetools[11]allowacomplete andextensivetestofoursoftwareandoftheproto colsemantics.

4. DataMovers

SincetheprimaryintentofIBPistoprovideacom monabstractionofstorage,itisarguablethatthi rd partytransferofdatabetweendepotsisunnecessar y.Indeed,itislogicallypossibletobuildanex ternal serviceformovingdatabetweendepotsthataccess IBPallocationsusingonlythe IBP_loadand IBP_storecalls.However,suchaservicewouldhavetoacta saproxyforclients,andthisimmediately raisestrustandperformanceconcerns.The IBP_copyand IBP_mcopydatamovementcallswere providedinordertoallowasimpleimplementation thatavoidstheseconcerns.However,software architecturesbasedonexternaldatamovementopera tionsarestillofgreatinteresttous.

Theintentofthebasic IBP_copycallistoprovideaccesstoasimpledatatransfe roveraTCPstream betweendepots. IBP_mcopyisamoregeneralfacility,andcanprovideaccess tooperationsthatrange fromsimplevariantsonsimpleTCP-baseddatatrans fertohighlycomplexprotocolsusingmulticastor otheradvancednetworkfacilities.Inallcases,t hecallerisresponsiblefordeterminingwhetherth e requestedoperationisappropriatetothedepot’sn etworkenvironment,andforanyerrorstrategyshou ld thedatamovementcallreturninfailure.

Thedatamoverisaplug-inmoduletoanIBPdepot thatisactivatedeitherbyan **IBP_mcopy**callorby an **IBP_datamover**call.ThesecondcallisnotanAPIcall,butani nternalcallmadebythesendingIBP depot.Thesendingdepotisresponsibleforinvoki ngaDataMoverplug-inonthereceivingdepot,and it accomplishesthisby **fork**ingadatamovercontrolprocessthatsendsan **IBP_datamover** request, causingthereceivingdepotto **fork**asymmetricdatamovercontrol.Sendingandreceiv ingcontrol processesthen **exec**theappropriateDataMoverplug-insforthereques tedoperationandthesecooperate toperformtheoperation,thentheplug-inatthes endingdepotrepliestotheclientandthenbothpl ug-ins terminate.Thefigure1illustratesthisprocess.

TheDataMoversoftwarearchitecturecansupporta widevarietyofoperations,including:

- Point-to-multipointthroughsimpleiteratedTCPuni casttransfers
- Point-to-multipointthroughsimultaneousthreadedT CPunicasttransfers.
- UnreliableUDPpoint-to-mulitpointutilizingnative IPmulitcast
- Reliablepoint-to-multipointutilizingnativeIPmu lticast
- Fast,reliableUDPdatatransferoverprivatenetwo rklinks[12]

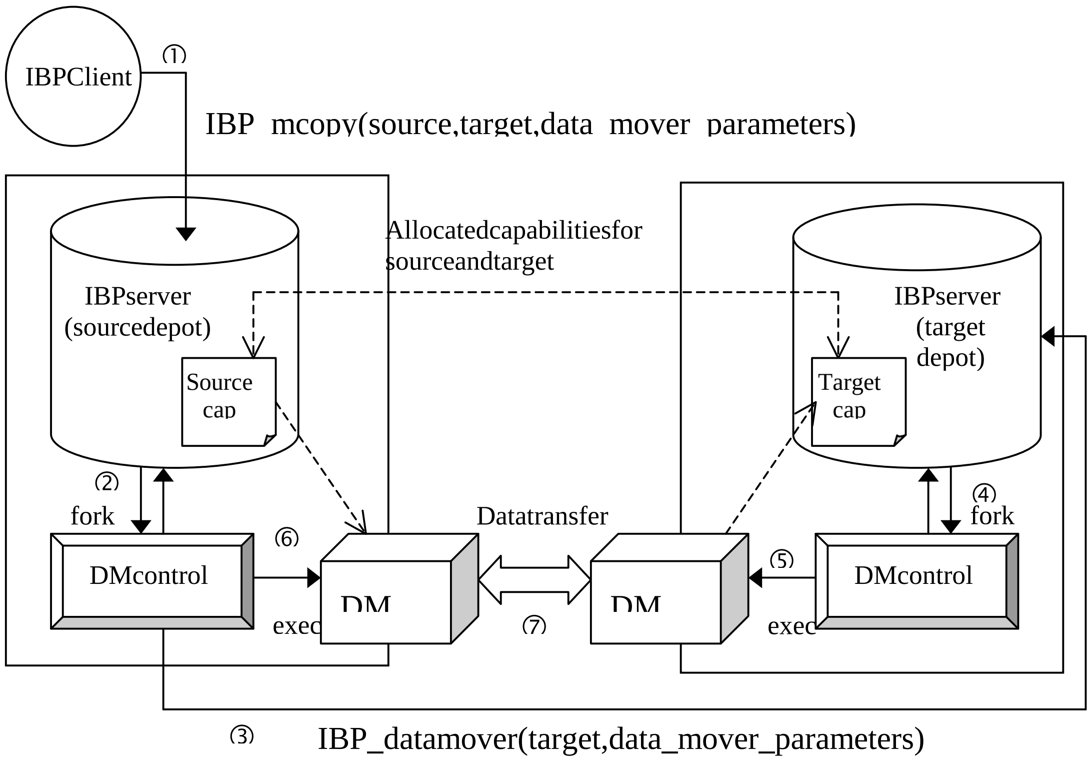

Figure1:DataMoverControlandTransferCoordinat ion.

5. ExperienceandApplications

OurmethodindevelopingtheInternetBackplanePro tocolisbasedonimplementationand experimentation.Anumberofsimpleapplicationsf romwithinourownresearchgrouphaveformedthe basisoftheexperiencethathasguidedourwork. Inaddition,afewexternalapplicationgroupshave pickeduponearlyimplementationsofIBPandcontr ibutebybothtrackingourreleasesandgivingus neededfeedbackforfuturedevelopments.However, itisonlywiththeupcomingreleaseoftheexNode libraryandserialization(Section6)andultimatel ytheLogisticalFileSystem(Section7)thatwebe lievea wideapplicationcommunitywillfindthesupporting toolsnecessarytoadoptanIBP-basedmethodology.

- IBP-mailisasystemthatusesIBPtotransmitand delivermailattachmentsthatrequirestorage resourcesbeyondthecapacityofstandardmailserv ers[13].TouseIBP-Mail,thesenderfirst uploadstheattachmentintoasuitableIBPservera ndthenforwardsthereadcapabilitytothe receiver,whocanuseittodownloadtheattachment (Figureqqq).AninitialversionofIBP-mail usedcustomCGIscriptstochosethedepotandimpl ementtheuploadanddownloadfunctions. Thecapabilitywastransmittedbetweensenderandr eceiverembeddedinaformthatinvokedthe receivingCGI.Thiscustomarchitecturehasbeenr eplacedbyagenericmechanismforstoring IBPcapabilities(seesection6ontheexNode)whic hallowsthemtobesentbetweensenderand receiverasasimplefileattachmentinastandard serializedformat.Themanipulationoffiles storedinIBPformatisnowhandledthroughgeneric toolsoperatingontheserializedexNode datastructure,andsotodayIBP-Mailisnotsomuc hanapplicationasawayofusingthose standardtoolstogetherwiththestandardMIMEatta chmentfacilityine-mail.
- NetSolveisadistributedcomputationtoolcreated byCassanovaandDongarra[14]toprovide remoteinvocationofnumericallibrariesforscient ificcode.Oneshortcomingoftheinitial NetSolvearchitectureisthatitisstateless:ase riesofcallstothesameservercannot,forinstan ce,

cacheargumentsorresultsinordertoavoidunnece ssarydatamovementinsubsequentcalls.One oftheapproachestakentooptimizeNetSolvewasto useanIBPdepotattheservertoimplement acacheunderthecontroloftheclient.Suchshor t-liveduseofIBPforvaluecachingisableto makegooduseofevenvolatileandtime-limitedall ocations[15].

- TAMANOIR[16]isaprojectdevelopedbytheRESAMl aboftheLyonIUniversityinthefield ofActiveNetworking.Itisaframeworkthatallows userstoeasilydeployandmaintain distributedactiveroutersinawideareanetwork. IBPdepotswillbeamongthestandardtools availabletoservicesimplementedwithintheTAMANO IRframework,alongwithothebasictools suchastheroutingmanagerandstreammonitoringt ools.Itwillalsobeusedtoimplement distributionandcachingofservices(distributeda sJavabyte-codemodules)thatareloadedby TAMANOIRon-demand,freeingTAMANOIRtomanageonly isowninternalcacheofservices.

6. TheexNode:AggregatingIBPStorageResourcestoProvideFile Services

Ourapproachtocreatingastrongfileabstraction ontheweakmodelofstorageofferedbyIBPcontinu es toparallelthedesignparadigmofthetraditional networkstack.Intheworldofend-to-endpacket delivery,ithaslongbeenunderstoodthatTCP,ap rotocolwithstrongsemanticproperties(e.g.,reli ability andin-orderdelivery)canbelayeredontopofIP, aweakdatagramdeliverymechanism.Inspiteofth e weakpropertiesofIPdatagramdelivery,strongerp ropertieslikereliabilityandin-orderdeliveryof packetscanbeachievedthroughthefundamentalmec hanismofretransmittingIPpackets. Retransmissioncontrolledbyahigherlayerprotoco l,combinedwithprotocolstatemaintainedatthe endpoints,overcomesnon-deliveryofpackets.Alln on-transientconditionsthatinterruptthereliable ,in-orderflowofpacketscanthenbereducedtonon-de livery.Weviewretransmissionasan *aggregation*of weakIPdatagramdeliveryservicestoimplementas trongerTCPconnection.

Thesameprincipleofaggregationcanbeappliedin ordertolayerastorageservicewithstrongseman tic propertiesontopofaweakunderlyingstoragereso urcethatdoesnotgenerallyprovidethem,suchas an IBPdepot.Examplesofaggregatingweakerstorages ervicesinordertoimplementstrongeronesinclude thefollowing:

- Reliability—Redundantstorageofinformationonre sourcesthatfailindependentlycan implementreliability(e.g.RAID,backups).
- Fastaccess—Redundantstorageofinformationonre sourcesindifferentlocalitiescan implementhighperformanceaccessthroughproximity (e.g.caching)orthroughtheuseof multipledatapaths(e.g.RAID[6]).
- Unboundedallocation—Fragmentationofalargeallo cationacrossmultiplestorageresources canimplementallocationsofunboundedsize(e.g.f ilesbuiltoutofdistributeddiskblocks, databasessplitacrossdisks).
- Unboundedduration—Movementofdatabetweenresour cesasallocationsexpirecanimplement allocationsofunboundedduration(e.g.migrationo fdatabetweengenerationsoftapearchive).

Inthisexposed-resourceparadigm,implementingaf ileabstractionwithstrongpropertiesinvolves creatingaconstructatahigherlayerthataggrega tesmoreprimitiveIBPbyte-arraysbelowit.Toapp ly theprincipleofaggregationtoexposedstorageser vices,however,itisnecessary *tomaintainstatethat representssuchanaggregationofstorageallocatio ns*,justassequencenumbersandtimersare maintainedtokeeptrackofthestateofaTCPsess ion.Fortunatelywehaveatraditional,well-unders tood modeltofollowinrepresentingthestateofaggreg atestorageallocations.IntheUnixfilesystem,t hedata structureusedtoimplementaggregationofunderlyi ngdiskblocksisthe *inode*(*intermediatenode*). UnderUnix,afileisimplementedasatreeofdisk blockswithdatablocksattheleaves.Theinterme diate nodesofthistreearetheinodes,whicharethemse lvesstoredondisk.TheUnixinodeimplementsonly theaggregationofdiskblockswithinasingledisk volumetocreatelargefiles;otherstrongpropert iesare sometimesimplementedthroughaggregationatalowe rlevel(e.g.RAID)orthroughmodificationstothe filesystemoradditionalsoftwarelayersthatmake redundantallocationsandmaintainadditionalstat e (e.g.AFS,HPSS)[1,8].

Followingtheexampleoftheinode,wehavechosen toimplementasinglegeneralizeddatastructure, whichwecallan *externalnode,* or *exNode*,inordertomanageofaggregateallocationsthat canbeused inimplementingnetworkstoragewithmanydifferent strongsemanticproperties(Figure2).Rathertha n aggregatingblocksonasinglediskvolume,theexN odeaggregatesstorageallocationsontheInternet, andtheexposednatureofIBPmakesIBPbyte-arrays exceptionallywelladaptedtosuchaggregations.I n thepresentcontextthekeypointaboutthedesign oftheexNodeisthatithasallowedustocreatea n abstractionofanetworkfiletolayeroverIBP-bas edstorageinawaythatiscompletelyconsistentw ith theexposedresourceapproach.

WeplantousetheexNodeasthebasisforasetof generictoolsforimplementingfileswitharange of characteristics.BecausetheexNodemustprovidein teroperabilitybetweenheterogeneousnodesona diverseInternet,wehavechosennottospecifyit asalanguage-specificdatastructure,butasanab stract datatypewithanXMLserialization.Thebasisof theexNodeisasingleallocation,representedbya n Internetresource,whichinitiallywillbeeithera nIBPcapabilityoraURL.Otherclassesofunderl ying storageresourcescanbeaddedforextensibilityan dinteroperability.

Theimportantelementstobedevelopedarelibrarie sthatimplementgenericrequirementssuchaslarge size(throughfragmentation),fastaccess(through caching),andreliability(throughreplication). Applicationsrequiringthesecharacteristicsshould beabletoobtainthemevenwithouthavingavailab le individualIBPdepotsthatimplementthosespecific characteristics–simplyusingtheAPIsshouldbe sufficientifaggregateresourcesthatareavailabl eforusesomewhereonthenetwork.TheexNodedata structurewillbeabasisforinteroperabilitywith inthelogisticalnetworkingAPI,andtheXML serializationwillbethebasisofinteroperability betweennetworknodes.

SinceourintentistousetheexNodefileabstract ioninanumberofdifferentapplications,wehave chosentoexpresstheexNodeconcretelyasanencod ingofstorageresources(URLsorIBPcapabilities) andassociatedmetadatainXML.IftheexNodeispl acedinadirectory,thefileitimplementscanbe imbeddedinanamespace.ButiftheexNodeissent asamailattachment,thereneednotbeacanonica l locationforit.TheuseoftheexNodebyvarying applicationswillprovideinteroperabilitysimilar to beingattachedtothesamenetworkfilesystem.

TheexNodemetadatamustbecapableofexpressinga tleastthefollowingrelationshipsbetweenthefil eit implementsandthestorageresourcesthatconstitut ethedatacomponentofthefilesstate:

- Theportionofthefileextentimplementedbyapar ticularresource(startingoffsetandending offsetinbytes)
- Theserviceattributesofeachconstituentstorage resource(e.g.reliabilityandperformance metrics,duration)
- Thetotalsetofstorageresourceswhichimplement thefileandtheaggregatingfunction(e.g. simpleunion,paritystoragescheme)

Despiteouremphasisonusinganexposed-resourcea pproach,itisnaturaltohavetheexNodesupport accesstostorageresourcesviaURLs,bothforthe sakeofbackwardcompatibilityandbecausethe Internetissoprodigiouslysuppliedwiththem.It isimportanttonote,however,thattheflexibility ofafile implementedbytheexNodeisafunctionoftheflex ibilityoftheunderlyingstorageresources.Theva lue ofIBPdoesnotconsistinthefactthatitisthe onlystorageresourcethatcanbeaggregatedinan exNode, butratherthatitisbyfarthemostflexibleand mosteasilydeployed.

6.1 TheexNodeAPI

TheexNodeAPIisastandardinterfaceforcreating ,communicatingandmanipulatingtheexNodedata structure.

- **xnd_create**, **xnd_destroy** arestandarddatastructureconstructor/destructor operations.
- **xnd_serialize**, **xnd_deserialize** write/readthestandardXMLserializationoftheex Nodedata structureto/fromafiledescriptor(seesection6. 2).
- **xnd_add_mapping**, **xnd_delete_mapping** add/deleteamappingfromtheexNode.
- **xnd_query**, **xnd_enum_next**, **xnd_enum_end**, **xnd_build_exNode**arequeryoperations. **xnd_query** returnstheaenumerationdatastructurerepresenti ngthesetofmappingswhoserange intersectswithaspecifiedtargetrange.Theenum erationcanbetraversedusingxnd_enum_next ordestroyedusing **xnd_enum_end**. **xnd_build_exNode** createsanewexNodefromthesetof mappingsthatcompriseanenumeration.
- **xnd_size**returnstheaggregateextentofallthemappingsi nanexNode.

ThisminimalexNodeAPIcanbeextendedinanumber ofwaysthathavebeenleftoutofthisaccount for the sake of clarity, and to keep from having to introduce additional structure. Some of these extensionsinclude:

- Queriescanbemuchmorecomplex,specifyingranges ofdataandtime,andreturningsetsof storageresourceswithassociatedmetadatatodirec ttheprocessofretrievingdata.
- Mappingscanbeannotatedtospecifyread-onlyorw rite-onlydata.
- Asstorageallocationsexpireorbecomeunavailable itwillbenecessarytomanagetheexNode byfindinganddeletingmappings,andthiswillreq uireadditionalmappingmanagementcalls.
- Byassociatingamappingwithasetofstoragespec ifiersandanaggregationfunction,itis possibletomodelgroupallocationssuchasRAID-li keerrorcorrection.
- Bydefiningmetricsonthelocationorperformance orothercharacteristicsofdifferentstorage allocationsitispossibletoinformtheuserofth eexNodewhichofmultiplealternativesto choose.

6.2 TheexNodeXMLSerialization

ThemobilityoftheexNodeisbasedontwopremises :

1. itispossibletopopulatetheexNodeexclusivel ywithnetwork-accessiblestorageresources
2. theexNodecanbeencodedinaportablewaythat canbeinterpretedatanynodeinthenetwork

Today,XMListhestandardtoolusedtoimplementp ortableencodingofstructureddata,andsoweare definingastandardXMLserializationoftheexNode .TheserializationisbasedontheabstractexNod e datastructure,andsoallowseachnodeorapplicat iontodefineitsownlocaldatastructure.

7. LoFS:TheLogisticalFileSystem

AsimpleIBP-basedfilesystemthatimplementsadi rectorystructureanddatastoragecompletelywithi n IBPhasbeendeveloped[17],usingan *adhoc* modifiedApachewebservertoactasthetrialapp lication thataccessesitssourcefilesthroughit.Inther ealmofthisexperiment,werestrictedupdatestoc omplete replacementofafile,allowingatomicupdatestob eimplementedthroughthedirectory.Averyinteres ting featureofthisprojectcanbeidentifiedinthepo ssibilityofhavingalocalIBPdepotasdatacache ,to improveperformance.Thepreliminarytestsresults showagoodpotential,butmoretestsneedtobe conductedinordertohavevalidresults.

OurApache-basedfilesystemwasaprototypeonwhi chtotesttheIBPimplementationandrobustness. Atruedistributedfilesystem,builtonwide-area resources,musthaveatitscoretheabilitytodea lwith thecommonplaceoccurencesoffailuresandunbounde dnetworklatencies.Ourimplementationwillbe basedontwocorefunctionalities:

- TheexNodeasthemainmetadatatype.
- Alog-structuredapproachtostoringdata

TheexNodeisexplainedabove.Log-structuredfile systems(LFS's)wereinventedinthelate80'sas a waytoimprovetheperformanceoffilesystemwrite s[18,19].Insteadofoverwritingdatainplace, whichresultsindiskwritesthatcanbescattered acrossthedisk,dataisappendedtologs,whichar e flushedtodisk *enmasse* ,resultinginamoreefficientuseofcontiguousd iskblocks.Thisofcourse resultsindeaddataspreadthroughoutthefilesyst em,whichmustbereclaimedbyaseparate *gargabe collector*process.ManyimplementationsofLFS'sdemonstrate dimprovedoverallfilesystem performance[20,21].

Asanunexpectedbenefit,LFS'swerefoundtohave otherdesirableproperties,suchasextremely efficientfailurerecovery[19],theabilitytodeal smoothlywithcompressiononthestoragesubstrate [22], andtheabilitytoeasesynchronizationworrieswh enreplicationisaddedtothefilesystem[23].T his latterpropertyiswhatmakesLFS'sattractivetoL oFS.

Recently,astoragesystemcalledSwarmhasbeende velopedattheUniversityofArizona[24],which implementsa *storageserver* layer,intendedtobealayerbetweenrawornetwo rk-attachedstorage,and filesystemclients.WithSwarm,clientsproducea ppend-onlylogsasinaLFS,whichthenbecome stripedacrossmultiplestorageserverswithRAIDL evel5parityencodingtotoleratethelossofany serverinthecollection.LikeinatypicalLFS,fi leupdatesresultinlogrecordswhichareappended to newlogs,andtheupdateddataeventuallybecomesc leanedbyacleanerthread.

Thisstructurehasthebasicelementsneededtoimp lementfilesystemsonthewidearea,sinceitavoi ds thesynchronizationproblems(typicallysolvedbyh oldinglocks)ofupdatingdatainplace,andthelo g-basedstructureeasesthetaskofrebuildingstate followingafailure.Itisastorageservice,howe ver,and notafull-blownfilesystembecauseitleavesissu esofnaming,sharingandsecurityuptotheclient s.The Swarmresearchershavebuiltprototypefilesystems forSwarmonalocalareanetwork[25].

WeplantostartwiththemethodologyofSwarmaso urbasedesignforLoFS,usingtheexNodedata structureasthebasicmetadatablock.Beneaththi sfoundation,wewilluseIBPserverstoprovideth e basicstorageservices,takingtheplaceofstandar ddisksandnetworkattachedstoragedevices.Ont opof thisfoundation,wemustincorporatethefollowing majorschanges:

1. Unlikerawandnetwork-attacheddisks,IBPbyte arrayshavetimelimits.Asimplewayof incorporatingthisintothefilesystemistohave anotherthread,muchlikethecleanerthread,that refreshestimelimits,andifthatisnotpossible, copiesthedatatoanotherIBPbytearray.
2. Manyfilesystemsimplementstripingand/orrepl icationforperformanceimprovementsonmore tightlycouplednetworksthanthewidearea.Onth ewidearea,cache-basedreplication,and logisticallyscheduledreplication(asisperformed inarudimentaryfashionbyIBPMail)willbea necessity,andwillhavetobeincorporatedintoLo FS.
3. Assuch,filesinLoFSwillconsistofreplicate dblocks,andcodingblocks(likeparityblocksin RAID)thatenableclientstorebuildfileblockswh entheyareunavailable.Thecodingwillbe basedonReed-Solomoncoding[26],whichallowsfor asystemtoprovide$m$codingblocksfor *n*blocksofdata,andthentotoleratethefailureo f{\emany} *m*blocks(note,RAIDLevel5 parityisequivalenttoReed-Solomoncodingwith *m*equalto1).

Oncethesechangesareinplace,wemayexperiment withtheuseoftime-limitedIBPstorageonthe wide-areaasthebasisforanetwork-widefilesyst em.

8. RelatedWork

IBPoccupiesanarchitecturalnichesimilartothat ofnetworkfilesystemssuchasAFS[1]andNetwor k AttachedStorageappliances[27],butitsmodelof storageismoreprimitive,makingitsimilarinsom e waystoStorageAreaNetworking(SAN)technologies developedforlocalnetworks.IntheGrid community,projectssuchasGASS[28]andtheSDSC StorageResourceBroker[29]arefilesystem overlaysthatimplementauniformfileaccessinter faceandalsoimposeuniformdirectory,authenticat ion andaccesscontrolframeworksontheirusers.

9. Conclusions

Validation of the claimed scalability of the logistical model of file system design based on IBP requires a research program based on implementation and extensive experimentation and experience through deployment. For this reason, we are commited to an aggressive schedule of code release, and all released code can be freely downloaded from our project website `http://icl.cs.utk.edu/ibp`. Today, the depot software runs on a variety of Unix-based operating systems and the Win32 platform and the client library runs on these platforms as well as Java. Porting to other systems is being pursued as interested user communities or interesting experimental opportunities are identified. The test suite and IBP application-building tools are also available. A specification of the exNode serialization and a preliminary version of the exNode library are scheduled for release before publication of this paper. The L-Bone is a service available to any IBP depot administrator, and the Logistical File System is currently under development, and the design of content distribution systems based on IBP is being studied.

An underlying thesis of our research program in the confluence of networking and storage technologies is that community-based resource sharing is one of the important factors that distinguishes the Internet from other communication networks. In formulating the Internet Backplane Protocol as a common mechanism for the sharing of storage resources, we are making a conscious attempt to emulate key aspects of IP networking while generalizing them to a new domain. While the challenges are considerable, and success is far from guaranteed, the potential reward is a new phase in the Internet revolution involving not just end-to-end communication but also the management of distributed state ranging from communication buffers to distributed files with the myriad policies and algorithms used by distributed applications of all sorts applied to a common underlying infrastructure. The ultimate goal is not simply to enable distributed file systems to scale across administrative domains in the manner of the Web, but to achieve a similar level of deployability for distributed systems of all sorts, ultimately integrating distributed computational resources (process cycles) to create a Logistical Computing and Internetworking infrastructure provisioned with the fundamental troika of distributed resources: bandwidth, storage and computation.

10. References

[1] J. H. Morris, M. Satyanarayan, M. H. Conner, J. H. Howard, D. S. H. Rosenthal, and F. D. Smith, "Andrew: A Distributed Personal Computing Environment," Communications of the ACM, vol. 29, no. 3, pp. 184-201, 1986.

[2] M. Satyanarayanan and M. Spasojevic, "AFS and the Web: Competitors or Collaborators?," Operating Systems Review, vol. 31, no. 1, pp. 18-23, 1997.

[3] M. Beck, T. Moore, J. Plank, and M. Swany, "Logistical Networking: Sharing More Than the Wires," in Active Middleware Services, vol. 583, The Kluwer International Series in Engineering and Computer Science, S. Hariri, C. Lee, and C. Raghavendra, Eds. Boston: Kluwer Academic Publishers, 2000.

[4] J. Plank, M. Beck, W. Elwasif, T. Moore, M. Swany, and R. Wolski, "The Internet Backplane Protocol: Storage in the Network," presented at NetStore99: The Network Storage Symposium, Seattle, WA, 1999.

[5] A. Bassi, M. Beck, J. Plank, and R. Wolski, "Internet Backplane Protocol: API 1.0," Department of Computer Science, University of Tennessee, Knoxville, TN, CS Technical Report, ut-cs-01-455, March 16, 2001 2001. http://www.cs.utk.edu/~library/2001.html.

[6] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson, "RAID: High-performance, reliable secondary storage," ACM Computing Surveys, vol. 26, pp. 145-185, 1994.

[7] H. Garcia-Molina and K. Salem, "Disk striping," in 2nd International Conference on Data Engineering: IEEE, 1986, pp. 336-342.

[8] R. W. Watson and R. A. Coyne, "The Parallel I/O Architecture of the High-Performance Storage System (HPSS)," presented at IEEE Mass Storage Systems Symposium, 1995.

[9] M.Beck,T.Moore,andJ.S.Plank,"Exposedvs .EncapsulatedApproachestoGridService Architecture,"presentedat2ndInternationalWorks hoponGridComputing,Denver,CO,Nov. 12,2001,2001.

[10] M.T.Rose,"TheBlocksExtensibleExchangePr otocolCore,"InternetEngineeringTaskForce, IETFRFC,RFC3080,March2001.http://www.ietf.org/ rfc/rfc3080.txt.

[11] A.BassiandX.Lee,"InternetBackplaneProto cal-TestLanguage1.0,"Departmentof ComputerScience,UniversityofTennessee,Knoxvill e,TN,CSTechnicalReport,ut-cs-01-454, June2001.http://www.cs.utk.edu/~library/2001.html.

[12] M.BeckandE.Fuentes,"AUDP-BasedProtocol forFastFileTransfer,"Departmentof ComputerScience,UniversityofTennessee,Knoxvill e,TN,CSTechnicalReport,ut-cs-01-456, June2001.http://www.cs.utk.edu/~library/2001.html.

[13] W.Elwasif,J.Plank,M.Beck,andR.Wolski, "IBP-Mail:ControlledDeliveryofLargeMail Files,"presentedatNetStore99:TheNetworkStorag eSymposium,Seattle,WA,1999.

[14] H.CasanovaandJ.Dongarra,"ApplyingNetSolv e'sNetworkEnabledServer,"IEEE ComputationalScience&Engineering,vol.5,no.3, pp.57-66,1998.

[15] D.C.Arnold,S.S.Vahdiyar,andJ.Dongarra, "OntheConvergenceofComputationalandData Grids,"ParallelProcessingLetters,forthcoming.

[16] J.GelasandL.Lefevre,"TAMANOIR:AHighPe rformanceActiveNetworkFramework,"in ActiveMiddlewareServices,vol.583,TheKluwerIn ternationalSeriesinEngineeringand ComputerScience,S.Hariri,C.Lee,andC.Raghave ndra,Eds.Boston:KluwerAcademic Publishers,2000.

[17] A.BurtonandM.Beck,"CreatingaFileSystem withCachingUsingInternetBackplane Protocol,"DepartmentofComputerScience,Universi tyofTennessee,Knoxville,TN,CS TechnicalReport,ut-cs-01-465,July192001.http: //www.cs.utk.edu/~library/2001.html.

[18] J.K.OusterhoutandF.Douglis,"Beatingthe I/OBottleneck:ACaseforLog-structuredFile Systems,"OperatingSystemsReview,vol.23,no.1, pp.11-27,1989.

[19] M.RosenblumandJ.K.Ousterhout,"TheDesign andImplementationfoaLog-StructuredFile System,"OperatingSystemsReview,vol.25,no.5, pp.1-15,1991.

[20] M.Seltzer,K.Bostic,M.K.McKusick,andC. Staelin,"AnImplementationofaLog-Structured FileSystemforUNIX,"inConferenceProceedings,U senixWinter1993TechnicalConference. SanDiego,1993,pp.307-326.

[21] J.T.Kohl,C.Staelin,andM.Stonebraker,"H ighLight:UsingaLog-structuredFileSystemfor TertiaryStorageManagement,"inConferenceProceed ings,UsenixWinter1993Technical Conference.SanDiego,1993,pp.435-447.

[22] M.Burrows,C.Jerian,B.Lampson,andT.Mann ,"On-lineDataCompressioninaLog-structuredFileSystem,"inFifthInternationalCon ferenceonArchitecturalSupportfor ProgrammingLanguagesandOperatingSystems:ACM,1 992,pp.2-9.

[23] J.H.HartmanandJ.K.Ousterhout,"TheZebra StripedNetworkFileSystem,"Operating SystemsReview--14thACMSymposiumonOperatingS ystemPrinciples,vol.27,no.5,pp.29-43,1993.

[24] I.MurdockandJ.H.Hartman,"Swarm:ALog-St ructuredStorageSystemforLinux,"in ProceedingsofFREENIXTrack:2000USENIXAnnualTe chnicalConference.SanDiego,2000.

[25] J.H.Hartman,I.Murdock,andT.Spalink,"Th eSwarmScalableStorageSystem,"in19th InternationalConferenceonDistributedComputingS ystems(ICDCS),1999,pp.74-81.

[26] J.S.Plank,"AtutorialonReed-Solomoncodin gforfault-toleranceinRAID-likesystems," Software-Practice&Experience,vol.27,pp.995- 1012,1997.

[27] G.GibsonandR.V.Meter,"NetworkAttachedS torageArchitecture,"Communicationsofthe ACM,vol.43,no.11,pp.37-45,2000.

[28] J.Bester,I.Foster,C.Kesselman,J.Tedesco ,andS.Tuecke,"GASS:ADataMovementand AccessServiceforWideAreaComputingSystems,"pr esentedatSixthWorkshoponI/Oin ParallelandDistributedSystems,May5,1999,1999 .

[29] C.Baru,R.Moore,A.Rajasekar,andM.Wan," TheSDSCStorageResourceBroker,"presented atCASCON'98,Toronto,Canada,1998.