
DEZSYS

Verteiltes Dateisystem mit ORIFS

**Systemtechnik DEZSYS
5BHITT 2015/16**

Burkhard Hampl, Simon Wortha

Note:

Betreuer: Borko/Micheler

Version 1

Begonnen am 1. April 2016

Beendet 2016

Inhaltsverzeichnis

1	Einführung.....	3
	Installation und Implementierung	3
	Gegenüberstellung.....	3
	Info	3
2	Ergebnisse.....	4
2.1	Installation	4
2.2	Tests.....	4
2.3	Ori API.....	4
	replicate	4
	Master	4
	Slave	5
	Snapshot.....	6
	checkout	6
	Master	6
	Slave	7
	graft	7
	filelog.....	7
	list	7
	log	8
	merge	8
	newfs.....	8
	Pull	9
	Master	9
	Slave	9
	remote.....	9
	removefs	9
	show.....	10
	status	10
	tip	10
	varlink	10
2.4	Andere Verteile Datei Systeme	11
	HDFS	11
	GlusterFS	11
3	Gegenüberstellung.....	12
4	Zeitaufzeichnung	12
5	Quellen	13

1 Einführung

Installation und Implementierung

"Ori is a distributed file system built for offline operation and empowers the user with control over synchronization operations and conflict resolution. We provide history through light weight snapshots and allow users to verify the history has not been tampered with. Through the use of replication instances can be resilient and recover damaged data from other nodes." [1]

Installieren Sie Ori und testen Sie die oben beschriebenen Eckpunkte dieses verteilten Dateisystems (DFS). Verwenden Sie dabei auf jeden Fall alle Funktionalitäten der API von Ori um die Einsatzmöglichkeiten auszuschöpfen. Halten Sie sich dabei zuallererst an die Beispiele aus dem Paper im Kapitel 2 [3]. Zeigen Sie mögliche Einsatzgebiete für Backups und Roadwarriors (z.B. Laptopbenutzer möchte Daten mit zwei oder mehreren Servern synchronisieren). Führen Sie auch die mitgelieferten Tests aus und kontrollieren Sie deren Ausgaben (Hilfestellung durch Wiki [2]).

Gegenüberstellung

Wo gibt es Überschneidungen zu anderen Implementierungen von DFS? Listen Sie diese auf und dokumentieren Sie mögliche Entscheidungsgrundlagen für mindestens zwei unterschiedliche Einsatzgebiete. Verwenden Sie dabei zumindest HDFS [4] und GlusterFS [5] als Gegenspieler zu Ori. Weitere Implementierungen sind möglich aber nicht verpflichtend. Um aussagekräftige Vergleiche anstellen zu können, wäre es von Vorteil die anderen Systeme ebenfalls - zumindest oberflächlich - zu testen.

Info

Gruppengröße: 2 Mitglieder

Gesamtpunkte: 16

- Installation und Testdurchlauf von Ori: 2 Punkte
- Einsatz/Dokumentation der Ori API (replicate, snapshot, checkout, graft, filelog, list, log, merge, newfs, pull, remote, removefs, show, status, tip, varlink): 8 Punkte
- Gegenüberstellungstabelle: 4 Punkte
- Einsatz der Gegenspieler: 2 Punkte

2 Ergebnisse

2.1 Installation

OS: Ubuntu 14.04.4

```
sudo apt-get install scons build-essential pkg-config libboost-dev uuid-dev libfuse-dev libevent-
dev libssl-dev libedit-dev
git clone http://bitbucket.org/orifs/ori.git
cd ori
scons
sudo scons PREFIX="/usr/local/" install
# generate a ssh key without a password
ssh-keygen
cat ~/.ssh/id_rsa.pub > ~/.ssh/authorized_keys
Danach VM gecloned und die Hostnamen (und des gleichen) ausgetauscht
```

2.2 Tests

Nun kann man die Tests auf einem Host ausführen (oder auch nicht ;))

```
cd ori
cat > runtests_config.sh <<END
# Required for Mac OS X and FreeBSD only (comment out on Linux machines)
# We commented this out due to running it on Debian
#export UMount="umount"

# Not updated to new CLI
HTTP_CLONE="no"
HTTP_PULL="no"
MERGE="no"
MOUNT_WRITE="no"
MOUNT_WRITE_PYTHON_MT="no"
END
```

2.3 Ori API

replicate

Die "replicate"-Funktion bietet die Möglichkeit mittels orisync zwischen einem Master und mehreren Slaves ein Orifs-Dateisystem zu synchronisieren.

Master

```
user@orifs-ubuntu1:~/testsync$ orisync init
Is this the first machine in the cluster (y/n)? y
Enter the cluster name: testsync
```

Use the following configuration for all other machines:

```
Cluster Name: testsync
Cluster Key: t8jhfkm
```

Now use 'orisync add' to register repositories.

```
user@orifs-ubuntu1:~/testsync$ ori newfs testfs
user@orifs-ubuntu1:~/testsync$ orisync add /home/user/.ori/testfs.ori
user@orifs-ubuntu1:~/testsync$ orisync
OriSync started as pid 1074
user@orifs-ubuntu1:~/testsync$ orisync list
```

```

Repo                               Mounted                               Peers
/home/user/.ori/testfs.ori         false
user@orifs-ubuntu1:~/testsync$ mkdir testfs
user@orifs-ubuntu1:~/testsync$ orifs testfs
user@orifs-ubuntu1:~/testsync$ mount
/dev/mapper/orifs--ubuntu1--vg-root on / type ext4 (rw,errors=remount-ro)
[...]
orifs on /home/user/testsync/testfs type fuse.orifs (rw,nosuid,nodev,user=user)
user@orifs-ubuntu1:~/testsync$ orisync list
Repo                               Mounted                               Peers
/home/user/.ori/testfs.ori         true
user@orifs-ubuntu1:~/testsync$ cd testfs/
user@orifs-ubuntu1:~/testsync/testfs$ ll
total 6
drwxr-xr-x 2 user user 512 Apr 15 08:28 ./
drwxrwxr-x 3 user user 4096 Apr 15 08:28 ../
-rw----- 1 user user 26 Jan 1 1970 .ori_control
drwxr-xr-x 2 user user 512 Jan 1 1970 .snapshot/
user@orifs-ubuntu1:~/testsync/testfs$ mkdir testdir
user@orifs-ubuntu1:~/testsync/testfs$ touch testfile
user@orifs-ubuntu1:~/testsync/testfs$ echo "Hello" > testfile
user@orifs-ubuntu1:~/testsync/testfs$ cat testfile
Hello
user@orifs-ubuntu1:~/testsync/testfs$ ll
total 6
drwxr-xr-x 3 user user 512 Apr 15 08:28 ./
drwxrwxr-x 3 user user 4096 Apr 15 08:28 ../
-rw----- 1 user user 26 Jan 1 1970 .ori_control
drwxr-xr-x 2 user user 512 Jan 1 1970 .snapshot/
drwxrwxr-x 2 user user 512 Apr 15 08:31 testdir/
-rw-rw-r-- 1 user user 6 Apr 15 08:31 testfile
user@orifs-ubuntu1:~/testsync/testfs$ echo -e "Hallo Welt\nTest 123" > testfile
user@orifs-ubuntu1:~/testsync/testfs$ cat testfile
Hallo Welt
Test 123

```

Slave

```

user@orifs-ubuntu2:~/testsync$ orisync init
Is this the first machine in the cluster (y/n)? n
Enter the cluster name: testsync
Enter the cluster key: t8jhfhkm

```

Use the following configuration for all other machines:
Cluster Name: testsync
Cluster Key: t8jhfhkm

```

Now use 'orisync add' to register repositories.
user@orifs-ubuntu2:~/testsync$ ori replicate user@10.0.105.27:testfs
Cloning from user@10.0.105.27:testfs to /home/user/.ori/testfs.ori
The authenticity of host '10.0.105.27 (10.0.105.27)' cant be established.
ECDSA key fingerprint is 7a:f6:b4:13:fb:66:82:49:3c:2e:4d:2e:5e:7b:af:2f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.105.27' (ECDSA) to the list of known hosts.
user@orifs-ubuntu2:~/testsync$ orisync add /home/user/.ori/testfs.ori
user@orifs-ubuntu2:~/testsync$ orisync
OriSync started as pid 1141
user@orifs-ubuntu2:~/testsync$ orisync list
Repo                               Mounted                               Peers
/home/user/.ori/testfs.ori         false                                10.0.105.27
user@orifs-ubuntu2:~/testsync$ mkdir testfs
user@orifs-ubuntu2:~/testsync$ orifs testfs
user@orifs-ubuntu2:~/testsync$ ori list
Name                               File System ID

```

```

testfs                                f5537585-7d35-4099-831f-4d28951c6d9f
user@orifs-ubuntu2:~/testsync$ cd testfs/
user@orifs-ubuntu2:~/testsync/testfs$ ll
total 7
drwxr-xr-x 3 user user  512 Apr 15 08:32 ./
drwxrwxr-x 3 user user 4096 Apr 15 08:30 ../
-rw----- 1 user user   26 Jan  1 1970 .ori_control
drwxr-xr-x 2 user user  512 Jan  1 1970 .snapshot/
drwxrwxr-x 2 user user  512 Apr 15 08:31 testdir/
-rw-rw-r-- 1 user user    6 Apr 15 08:31 testfile
user@orifs-ubuntu2:~/testsync/testfs$ cat testfile
Hallo Welt
Test 123

```

Snapshot

Mittels der "snapshot"-Funktion kann man Snapshots von Änderungen machen, ähnlich wie commits bei git. Dabei darf das Filesystem nicht mittels orisync synchronisiert werden, da orisync automatisch committed.

```

user@orifs-ubuntu1:~/testsync$ ori newfs NoRepo
user@orifs-ubuntu1:~/testsync$ mkdir NoRepo
user@orifs-ubuntu1:~/testsync$ orifs NoRepo
user@orifs-ubuntu1:~/testsync$ cd NoRepo/
user@orifs-ubuntu1:~/testsync/NoRepo$ ori status
user@orifs-ubuntu1:~/testsync/NoRepo$ mkdir TEST
user@orifs-ubuntu1:~/testsync/NoRepo$ ori status
user@orifs-ubuntu1:~/testsync/NoRepo$ ori snapshot FIRST
Committed fa05a4d10b818fb4ce647c4eb0bb209c4a8fc8386cb546e67a3612571ea81b6d
user@orifs-ubuntu1:~/testsync/NoRepo$ ori status
user@orifs-ubuntu1:~/testsync/NoRepo$ touch MYFILE
user@orifs-ubuntu1:~/testsync/NoRepo$ ori status
A  /MYFILE
user@orifs-ubuntu1:~/testsync/NoRepo$ ori snapshot SECOND
Committed ab692ac9dfc319db41866cd6dcec86e014a52bb31a17f3bd67deb9c9ab42bb33
user@orifs-ubuntu1:~/testsync/NoRepo$ ori status
Die Daten der Snapshots werden unter .snapshot/<Snapshotname> gespeichert

```

```

user@orifs-ubuntu1:~/testsync/NoRepo$ ori status
user@orifs-ubuntu1:~/testsync/NoRepo$ rm MYFILE
user@orifs-ubuntu1:~/testsync/NoRepo$ ori status
D  /MYFILE
user@orifs-ubuntu1:~/testsync/NoRepo$ cp .snapshot/SECOND/MYFILE .
user@orifs-ubuntu1:~/testsync/NoRepo$ ori status
M  /MYFILE

```

checkout

Mittels checkout ist es möglich, auf einen snapshot zurückzusetzen

Master

```

user@orifs-ubuntu1:~/testsync$ ori newfs testfs
user@orifs-ubuntu1:~/testsync$ orifs testfs
user@orifs-ubuntu1:~/testsync$ cd testfs/
user@orifs-ubuntu1:~/testsync/testfs$ ori status
user@orifs-ubuntu1:~/testsync/testfs$ mkdir testdir
user@orifs-ubuntu1:~/testsync/testfs$ echo "test123" > testfile
user@orifs-ubuntu1:~/testsync/testfs$ cat testfile
test123
user@orifs-ubuntu1:~/testsync/testfs$ ori snapshot testcommit
Committed 5217bf9930a4a08e946436711e60b8a319eda6ebbe3266d0af588135444a6c4b
user@orifs-ubuntu1:~/testsync/testfs$ ori status
user@orifs-ubuntu1:~/testsync/testfs$ ori snapshots
testcommit

```

Slave

```

user@orifs-ubuntu2:~/testsync$ ori newfs testfs
user@orifs-ubuntu2:~/testsync$ mkdir testfs
user@orifs-ubuntu2:~/testsync$ orifs newfs
user@orifs-ubuntu2:~/testsync$ orifs testfs
user@orifs-ubuntu2:~/testsync$ cd testfs/
user@orifs-ubuntu2:~/testsync/testfs$ ori pull user@10.0.105.43:testfs
Pulled up to 5217bf9930a4a08e946436711e60b8a319eda6ebbe3266d0af588135444a6c4b
user@orifs-ubuntu2:~/testsync/testfs$ ori checkout
5217bf9930a4a08e946436711e60b8a319eda6ebbe3266d0af588135444a6c4b
Checkout success!
Checkout success!
user@orifs-ubuntu2:~/testsync/testfs$ ls -la
total 7
drwxr-xr-x 3 user user 512 Apr 21 10:24 .
drwxrwxr-x 3 user user 4096 Apr 21 10:57 ..
-rw----- 1 user user 26 Jan 1 1970 .ori_control
drwxr-xr-x 2 user user 512 Jan 1 1970 .snapshot
drwxrwxr-x 2 user user 512 Apr 21 10:20 testdir
-rw-rw-r-- 1 user user 8 Apr 21 10:23 testfile
user@orifs-ubuntu2:~/testsync/testfs$ cat testfile
test123
user@orifs-ubuntu2:~/testsync/testfs$ ori snapshots
testcommit

```

graft

Kopiert, aber mit Metadaten und "Skelett"

```

user@orifs-ubuntu2:~/testsync/repo1$ mkdir test1
user@orifs-ubuntu2:~/testsync/repo1$ ls
test1
user@orifs-ubuntu2:~/testsync/repo1$ cd ../repo2/
user@orifs-ubuntu2:~/testsync/repo2$ ls
user@orifs-ubuntu2:~/testsync/repo2$ ori graft ~/testsync/repo1/ ~/DezSys/GraftB
.ori_control .snapshot/ test1/
user@orifs-ubuntu2:~/testsync/repo2$ ori graft ~/testsync/repo1/test1/ ~/testsync/repo2
Warning: source or destination is not a repository.
user@orifs-ubuntu2:~/testsync/repo2$ ls
test1

```

filelog

Gibt die Snapshots aus, welche dieses File betreffen

```

user@orifs-ubuntu2:~/testsync/testfs$ ori filelog testfile
Commit: ac8c87db77b5a43cbbb004055b34ab30a53b3d5c6c4e5be08607dce0b1ce2323
Parents: 5217bf9930a4a08e946436711e60b8a319eda6ebbe3266d0af588135444a6c4b
Author: User,,,
Date: Thu Apr 21 12:37:18 2016

```

Created snapshot 'commit2'

```

Commit: 5217bf9930a4a08e946436711e60b8a319eda6ebbe3266d0af588135444a6c4b
Parents: 0000000000000000000000000000000000000000000000000000000000000000
Author: User,,,
Date: Thu Apr 21 10:24:04 2016

```

Created snapshot 'testcommit'

list

Listet alle Dateien auf

```
user@orifs-ubuntu2:~/testsync/testfs$ ori list
Name                               File System ID
repo1                             2c99c200-fcd8-4f2a-b9a7-6aa1b52663c8
repo2                             f5307b0a-1884-4ce6-80cf-58b933798343
testfs                            d64a46e8-9a23-4a9d-b58b-1698b9188c8b
```

log

Zeigt die Snapshots an (ähnlich wie git log)

```
user@orifs-ubuntu2:~/testsync/testfs$ ori log
Commit:    ac8c87db77b5a43cbbb004055b34ab30a53b3d5c6c4e5be08607dce0b1ce2323
Parents:   5217bf9930a4a08e946436711e60b8a319eda6ebbe3266d0af588135444a6c4b
Tree:      e8668435831d8141a7b1beba751a68a3dc1f5c5aebfe364a6360cf511561ee2a
Author:    User,,,
Date:      Thu Apr 21 12:37:18 2016
```

Created snapshot 'commit2'

```
Commit:    5217bf9930a4a08e946436711e60b8a319eda6ebbe3266d0af588135444a6c4b
Parents:   3cebc29890756dec403dcfb1fc8f106218c0543e8b2736d27291ad3ab3a89c2f
Tree:      3cebc29890756dec403dcfb1fc8f106218c0543e8b2736d27291ad3ab3a89c2f
Author:    User,,,
Date:      Thu Apr 21 10:24:04 2016
```

Created snapshot 'testcommit'

merge

Mittels des merge-Befehles kann man verschiedene Versionen/Dateien zusammenfügen.

```
user@orifs-ubuntu2:~/testsync/testfs$ ll
total 6
drwxr-xr-x 3 user user 512 Apr 21 21:52 ./
drwxrwxr-x 5 user user 4096 Apr 21 12:29 ../
-rw----- 1 user user 26 Jan 1 1970 .ori_control
drwxr-xr-x 2 user user 512 Jan 1 1970 .snapshot/
drwxrwxr-x 2 user user 512 Apr 21 21:52 TESTDIR/
-rw-rw-r-- 1 user user 7 Apr 21 21:52 testfile
user@orifs-ubuntu2:~/testsync/testfs$ ori pull user@10.0.0.14:testfs
Pulled up to 999a7cffe5c9d35665e19a996ebcc5c87636a052cfffec8ed1e862902ac1417dc
user@orifs-ubuntu2:~/testsync/testfs$ ori checkout
999a7cffe5c9d35665e19a996ebcc5c87636a052cfffec8ed1e862902ac1417dc
Checkout success!
user@orifs-ubuntu2:~/testsync/testfs$ ori merge
999a7cffe5c9d35665e19a996ebcc5c87636a052cfffec8ed1e862902ac1417dc
Merge success!
user@orifs-ubuntu2:~/testsync/testfs$ ll
total 7
drwxr-xr-x 3 user user 512 Apr 21 21:27 ./
drwxrwxr-x 5 user user 4096 Apr 21 12:29 ../
-rw----- 1 user user 26 Jan 1 1970 .ori_control
drwxr-xr-x 2 user user 512 Jan 1 1970 .snapshot/
drwxrwxr-x 2 user user 512 Apr 21 21:23 testdir/
-rw-rw-r-- 1 user user 5 Apr 21 21:23 testfile
```

newfs

Damit wird ein neues Filesystem hinzugefügt

```
user@orifs-ubuntu2:~/testsync$ ori newfs therepo
user@orifs-ubuntu2:~/testsync$ ori list
Name                               File System ID
therepo                           06db6225-8500-481d-a8ea-346ef37bb09b
```


Pull

Mittels pull kann man ein remote Filesystem pullen (wie git pull)

Master

```
user@orifs-ubuntu1:~/testsync$ ori newfs testfs
user@orifs-ubuntu1:~/testsync$ orifs testfs
user@orifs-ubuntu1:~/testsync$ cd testfs/
user@orifs-ubuntu1:~/testsync/testfs$ ori status
user@orifs-ubuntu1:~/testsync/testfs$ mkdir testdir
user@orifs-ubuntu1:~/testsync/testfs$ echo "test123" > testfile
user@orifs-ubuntu1:~/testsync/testfs$ cat testfile
test123
user@orifs-ubuntu1:~/testsync/testfs$ ori snapshot testcommit
Committed 5217bf9930a4a08e946436711e60b8a319eda6ebbe3266d0af588135444a6c4b
user@orifs-ubuntu1:~/testsync/testfs$ ori status
user@orifs-ubuntu1:~/testsync/testfs$ ori snapshots
testcommit
```

Slave

```
user@orifs-ubuntu2:~/testsync$ ori newfs testfs
user@orifs-ubuntu2:~/testsync$ mkdir testfs
user@orifs-ubuntu2:~/testsync$ orifs newfs
user@orifs-ubuntu2:~/testsync$ orifs testfs
user@orifs-ubuntu2:~/testsync$ cd testfs/
user@orifs-ubuntu2:~/testsync/testfs$ ori pull user@10.0.105.43:testfs
Pulled up to 5217bf9930a4a08e946436711e60b8a319eda6ebbe3266d0af588135444a6c4b
user@orifs-ubuntu2:~/testsync/testfs$ ls -la
total 7
drwxr-xr-x 3 user user 512 Apr 21 10:24 .
drwxrwxr-x 3 user user 4096 Apr 21 10:57 ..
-rw----- 1 user user 26 Jan 1 1970 .ori_control
drwxr-xr-x 2 user user 512 Jan 1 1970 .snapshot
drwxrwxr-x 2 user user 512 Apr 21 10:20 testdir
-rw-rw-r-- 1 user user 8 Apr 21 10:23 testfile
user@orifs-ubuntu2:~/testsync/testfs$ cat testfile
test123
user@orifs-ubuntu2:~/testsync/testfs$ ori snapshots
testcommit
```

remote

Damit kann man remote Filesysteme hinzufügen

```
user@orifs-ubuntu2:~/testsync/testfs$ ori remote add origin user@10.0.0.14:testfs
user@orifs-ubuntu2:~/testsync/testfs$ ori remote
Name          Path
ID
origin         user@10.0.0.14:testfs
```

```
user@orifs-ubuntu2:~/testsync/testfs$ ori pull
Pulled up to 999a7cffe5c9d35665e19a996ebcc5c87636a052cfffec8ed1e862902ac1417dc
```

removefs

Damit ist es möglich, eine mit newfs hinzugefügtes Filesystem zu löschen (unmounten muss man es manuell)

```
user@orifs-ubuntu2:~/testsync$ ori removefs therepo
user@orifs-ubuntu2:~/testsync$ ori list
Name          File System ID
```

show

Dieser Befehl zeigt Informationen des Repositories an

```
user@orifs-ubuntu1:~/testsync/testfs$ ori show
--- Repository ---
Root: /home/user/.ori/testfs.ori
UUID: c848af70-3eff-453a-ac3b-fa66d586d82c
Version: ORI1.1
HEAD: 999a7cffe5c9d35665e19a996ebcc5c87636a052cffe8ed1e862902ac1417dc
```

status

Zeigt den aktuellen Status an (wie git status)

```
user@orifs-ubuntu1:~/testsync/testfs$ ori status
A   /thefile
```

tip

Zeigt den Hash des aktuellen "HEAD" an

```
user@orifs-ubuntu1:~/testsync/testfs$ ori tip
999a7cffe5c9d35665e19a996ebcc5c87636a052cffe8ed1e862902ac1417dc
```

varlink

Zeigt Variablen der Machine an

```
user@orifs-ubuntu1:~/testsync/testfs$ ori varlink
Variable      Value
machtype      unknown
osname        unknown
domainname    (none)
hostname      orifs-ubuntu1
```

2.4 Andere Verteile Datei Systeme

HDFS

„Apache Hadoop is an open-source software framework developed in Java that allows distributed processing of large data sets across clusters of computers using simple programming models. It is composed of several modules such as Hadoop Yarn and Hadoop MapReduce for cluster resource management and parallel processing, Hadoop Distributed File System (HDFS) that provides high-throughput access to application data and other related sub-projects such as Cassandra, HBase, Zookeeper, etc. Several big companies use Hadoop for their services; Yahoo! has by far the most number of nodes in its massive Hadoop clusters at over 42,000 nodes as of July 2011, while Facebook stores more than 100 PetaByte of data on HDFS. Many other famous companies use Hadoop such as Amazon, E-Bay, LinkedIn, etc" [6]

Einsatz: Yahoo!

GlusterFS

GlusterFS ist ein verteiltes Dateisystem, das von RedHat entwickelt und ist open source. Der größte Teil des Codes steht unter der GNU GPL, aber auch andere open source Lizenzen sind für einige Code-Teile verwendet. GlusterFS ist in der Lage Data-Files von mehreren Petabytes und Tausenden an Clients zu nutzen. Es wird ein einzelner globaler Namespace genutzt und GlusterFS „clustert“ eine benutzerdefinierte Menge an Storage-Blocks über TCP/IP oder Infiniband RDMA.

Die Daten in diesen Storage-Blocks können von dem GlusterFS Server gemanagt werden und via GlusterFS Clients, Windows oder SAMBA Clients, welche CIFS nutzen, kann darauf zugegriffen werden.

Es ist nicht möglich GlusterFS zu benutzen um Dateien zwischen Server-Nodes zu teilen aber mehrere Clients können auf die Daten zugreifen und neue Dateien erstellen, beziehungsweise alte löschen. Normalerweise speichert jeder Server-Node seinen Teil der Daten als volle Dateien aber er ist durch „striping“ möglich, dass mehrere Server nur einen Teil einer Datei speichern. Die meisten Funktionalitäten sind als „translators“ verfügbar, wie zum Beispiel Striping, Replikation, Scheduling und Load Balancing.

Verzeichnisse werden voll exportiert und Daten müssen auf der Client-Seite „Strukturiert“ werden. [7, 8]

Einsatz: bei den meisten Open Source Projekten. Vor allem RedHat-Linux.

3 Gegenüberstellung

	oriFS	HDFS	GlusterFS
Dokumentation	-	+	+
Useability	-	~	-
Community	-	+	~
Fehlertoleranz	+	+	+
Hochverfügbarkeit	+	+	~
Replikation	+	+	+
Skalierbarkeit	+	+	+
Bewertung	~	+	~

[9]

4 Zeitaufzeichnung

Hampl

Wortha

Datum	Zeit	Datum	Zeit
01.04.2016	2 Stunden	01.04.2016	2 Stunden
14.04.2016	4 Stunden	14.04.2016	1 Stunde
21.04.2016	10 Stunden	22.04.2016	5 Stunden
Gesamt:	16 Stunden	Gesamt:	8 Stunden

5 Quellen

[1] Ori File System, Stanford Website, online: <http://ori.scs.stanford.edu/>, visited: 2016-04-01

[2] Ori File System, Bitbucket Wiki, online: <https://bitbucket.org/orifs/ori/wiki/Home>, visited: 2016-04-01

[3] Ali José Mashtizadeh, Andrea Bittau, Yifeng Frang Huang, David Mazières. Replication, History, and Grafting in the Ori File System. In Proceedings of the 24th Symposium on Operating Systems Principles, November 2013. Paper.

[4] Apache Hadoop FileSystem, <http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html>, visited: 2016-04-01

[5] GlusterFS, <http://gluster.readthedocs.org/en/latest/>, visited: 2016-04-01

[6] G Marzulli G Donvito and D Diacono. Testing of several distributed filesystems (hdfs, ceph and glusterfs) for supporting the hep experiments analysis. http://iopscience.iop.org/1742-6596/513/4/042014/pdf/1742-6596_513_4_042014.pdf.

[7] Dsimic. Glusterfs. <http://en.wikipedia.org/wiki/GlusterFS>

[8] Red Hat. About. http://www.gluster.org/documentation/About_Gluster/.

[9] oriFS; Aly Ahmed, Vennesa Belinic; <https://github.com/vbelinic-tgm/orifs/blob/master/doc/Protocol.pdf>