

---

# **Laborprotokoll**

## **DezSys13: Verteiltes Dateisystem mit OriFS**

---

**Systemtechnik Labor  
5BHIT 2015/16**

**Stefan Erceg & Martin Kritzl**

**Version 1.0**

**Note:**

**Betreuer: Prof. Borko & Micheler**

**Begonnen am 1. April 2016**

**Beendet am 21. April 2016**

## Inhaltsverzeichnis

1	Aufgabenstellung .....	3
1.1	Installation und Implementierung .....	3
1.2	Gegenüberstellung .....	3
1.3	Info .....	3
1.4	Quellen .....	4
2	Ergebnisse .....	5
2.1	Installation von OriFS .....	5
2.2	Testdurchlauf von OriFS .....	6
2.3	Einsatz der Ori API .....	8
2.3.1	Filesystem erstellen (newfs) .....	8
2.3.2	Filesystem löschen (removefs) .....	8
2.3.3	Anzeige vorhandener Filesysteme (list) .....	8
2.3.4	Anzeige von vorgenommenen Änderungen (status) .....	9
2.3.5	Anlegen von Snapshots (snapshot) .....	9
2.3.6	Anzeigen der History (log) .....	10
2.3.7	Replikation eines entfernten Filesystems (replicate) .....	11
2.3.8	Herunterladen eines entfernten Filesystems (pull) .....	11
2.3.9	Umsteigen auf einen anderen Snapshot (checkout) .....	12
2.3.10	Fusionierung zweier Versionen (merge) .....	12
2.3.11	Einmaliges Replizieren eines Filesystems (graft) .....	13
2.3.12	Anzeigen der History eines bestimmten Files (filelog) .....	13
2.3.13	Verwaltung von entfernten Verbindungen (remote) .....	14
2.3.14	Anzeigen von Repository-Informationen (show) .....	15
2.3.15	Anzeigen der aktuellen Commit-ID (tip) .....	15
2.3.16	Anzeigen der von OriFS verwendeten Variablen (varlink) .....	16
2.4	Gegenüberstellung der Dateisysteme .....	17
3	Github-Link .....	18
4	Quellen .....	18
5	Befehlsverzeichnis .....	21

# 1 Aufgabenstellung

## 1.1 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]).

## 1.2 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.

## 1.3 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

## 1.4 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 [<http://sigops.org/sosp/sosp13/>] , November 2013. Paper [<http://bit.ly/1TFBnWo>].
- [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

## 2 Ergebnisse

### 2.1 Installation von OriFS

Die Installation wurde auf einer VM durchgeführt, die als Betriebssystem „Ubuntu 14.04.4“ besitzt. Folgende 3 Befehle, welche auf der offiziellen Seite von OriFS einzusehen sind [1], wurden zu Beginn ausgeführt:

```
add-apt-repository ppa:ezyang/ppa
apt-get update
apt-get install ori
```

*Listing 1: Installation von OriFS über Packetmanager*

Mit diesen Befehlen holt man sich die letzte Stable-Version von OriFS (= 0.8.1), welche im Januar 2014 veröffentlicht wurde. Als Problem stellte sich hierbei heraus, dass „orisync“ nicht wie gewünscht funktionierte.

Aus diesem Grund wurde das Bitbucket-Repository [20] geklont und mittels „scons“ gebuildet. Somit war es möglich, sich die letzte Developer-Version zu holen. Folgende Schritte wurden hierbei ausgeführt:

```
# notwendige Packages installieren
sudo apt-get install scons build-essential pkg-config libboost-dev
                        uuid-dev libfuse-dev libevent-dev libssl-dev
                        libedit-dev

# Bitbucket-Repository klonen
git clone https://bitbucket.org/orifs/ori-orisyncng.git
cd ori-orisyncng

# Ori mittels "scons" builden
scons
```

*Listing 2: Repository klonen & Ori builden*

Daraufhin wird das File „SConstruct“ geöffnet und der Pfad folgender Zeile

```
PathVariable("PREFIX", "Installation target directory", "/usr/local/bin/", ...)
```

auf /usr/local/ geändert.

Folgende Befehle wurden schlussendlich noch zur Installation von OriFS durchgeführt:

```
# Ori mittels "scons" installieren
sudo scons install

# Public-SSH-Key generieren
ssh-keygen -t rsa

# Public-SSH-Key in das "authorized_keys"-File kopieren
cd ~/.ssh
cat id_rsa.pub >> authorized keys
```

*Listing 3: Installation von OriFS & Verwaltung des Public-SSH-Keys*

## 2.2 Testdurchlauf von OriFS

Um die Tests zum Laufen zu bringen, wurde zu Beginn eine Datei „runtests\_config.sh“ im Ordner „ori-orisyncng“ erstellt, welche folgenden Inhalt besitzt [20]:

```
HTTP_CLONE="no"
HTTP_PULL="no"
MERGE="no"
MOUNT_WRITE="no"
MOUNT_WRITE_PYTHON_MT="no"
```

*Listing 4: Inhalt des "runtest\_config.sh"-Files*

Daraufhin musste in den Files „53-mount-write-wget.sh“ und „61-mount-write-wget-mt.sh“, welche sich im Ordner „ori\_tests“ befinden, die Zeile

```
PKG_NAME="wget-1.14"
```

auf die aktuelle Version von „wget“, welche auf der VM installiert ist, geändert werden (bei uns daher wget-1.15).

Sobald die Änderungen vorgenommen wurden, kann das File „runtests.sh“ ausgeführt werden. Folgender Output wird daraufhin angezeigt und ist später auch im File „ori\_test\_results.txt“ einzusehen:

```
runtests.sh test results
Mon Apr 18 14:59:11 CEST 2016

01-new-remove-fs
02-local-clone
03-local-clone-empty
04-local-pull
05-local-clone-uds
11-ssh-clone
12-ssh-clone-empty
14-ssh-pull
15-ssh-clone-uds
SKIPPED: 21-http-clone
SKIPPED: 22-http-pull
30-local-clone-shallow
SKIPPED: 35-merge
SKIPPED: 51-mount-write
52-mount-write-zlib
53-mount-write-wget
60-mount-write-zlib-mt
61-mount-write-wget-mt
SKIPPED: 62-mount-write-python-mt
```

*Listing 5: Output beim Durchführen der OriFS-Tests*

## 2.3 Einsatz der Ori API

### 2.3.1 Filesystem erstellen (newfs)

Um ein Filesystem zu erstellen, ist es notwendig, ein neues Filesystem zu definieren und ihn auf einen spezifischen Ordner zu mounten. Dieser Ordner muss vorher manuell erstellt werden.

```
ori newfs <Filesystemname>  
mkdir <Filesystemname>  
orifs <Filesystemname>
```

*Listing 6: Befehle zur Erstellung eines Filesystems*

### 2.3.2 Filesystem löschen (removefs)

Ein bestehendes Filesystem kann auf Wunsch ebenfalls wieder gelöscht werden. Dazu muss zu Beginn der Ordner des Filesystems ungemounted und anschließend sowohl das Filesystem als auch der Ordner gelöscht werden.

```
umount <Filesystemname>  
ori removefs <Filesystemname>  
rm -rf <Filesystemname>
```

*Listing 7: Befehle zum Löschen eines Filesystems*

### 2.3.3 Anzeige vorhandener Filesysteme (list)

Um alle Filesysteme anzuzeigen, welche lokal existieren, ist folgender Befehl zu verwenden:

```
ori list
```

*Listing 8: Befehl zur Anzeige vorhandener Filesysteme*

Bei uns konkret wurde daraufhin das Filesystem „DezSysRepo“ angezeigt, jedoch war es aufgrund der Ausgabe nicht ersichtlich, ob es sich beim Filesystem um einen Master oder einen Slave handelt:

Name	File System ID
DezSysRepo	c3de0388-b98e-406c-9274-3471f25f0ed8

*Listing 9: Output aller vorhandenen Filesysteme*



### 2.3.4 Anzeige von vorgenommenen Änderungen (status)

Ähnlich wie bei dem bereits bekannten Befehl „git status“ können bei OriFS einzelne Änderungen an Files mittels folgendem Befehl eingesehen werden:

```
ori status
```

*Listing 10: Befehl zum Anzeigen vorgenommener Änderungen*

Im Folgenden sind die verschiedenen Änderungen angeführt, die mit „ori status“ angezeigt werden können:

```
A    /<neues File>
D    /<gelöschtes File>
M    /<geändertes File>
```

*Listing 11: Anzeige der verschiedenen Änderungen*

### 2.3.5 Anlegen von Snapshots (snapshot)

Snapshots dienen zum Festhalten eines bestimmten Zeitpunkts des Filesystems. Dabei werden die geänderten Daten gespeichert und können von replizierten Filesystemen verwendet werden. Bei OriFS betrachtet man den Befehl zum Anlegen eines Snapshots genauso wie bei git ein „git commit“ + „git push“.

```
ori snapshot <name>
```

*Listing 12: Befehl zum Anlegen eines Snapshots*

Sobald ein Snapshot angelegt wurde, ist dieser im Ordner „.snapshot“ vorzufinden. Dieser Ordner ist immer zugänglich und die Files darin können bei Bedarf wiederhergestellt oder zurückgesetzt werden.

Beispielsweise wurde bei uns nach der Erstellung eines Snapshots folgender Output angezeigt:

```
Committed
743fda1e039f9790d878d0df264999d3b2a342d2a67a123928ba6175dbfc9374
```

*Listing 13: Output nach einer erfolgreichen Erstellung eines Snapshots*

### 2.3.6 Anzeigen der History (log)

Möchte man die History aller durchgeführten Änderungen einsehen, so kann folgender Befehl verwendet werden:

```
ori log
```

*Listing 14: Befehl zum Anzeigen der History*

Um dies zu testen, wurden zuvor zwei Snapshots angelegt. Die Ausgabe nach der Ausführung des oben erwähnten Befehls erfolgt chronologisch mit dem neuesten Snapshot zu Beginn. Folgender Output wurde bei uns angezeigt:

```
Commit:
c03f71d21626bc6db67bad70d46c497ae571f5a65de9a0148403d75cb3495d25
Parents:
8df2210b6299e4bf086eae0cf473c8926ef49a4a6a3e4eed4656e3832660307a
Tree:
245cb1d68d6434b60564b546ef4b2c3134b7c3a3b7a23f4856d54280b484d278
Author:
Date: Mon Apr 18 16:38:34 2016

Created snapshot test2

Commit:
8df2210b6299e4bf086eae0cf473c8926ef49a4a6a3e4eed4656e3832660307a
Parents:
b2a342d2a67a123928ba6175dbfc93742b90b37b63a84bd25655798c85da6892
Tree:
5fb9eee38217b68ed9b106df1413e75fc74990482eeebc0fc72c2f8b0f0655aa
Author:
Date: Mon Apr 18 16:31:47 2016

Created snapshot test1
```

*Listing 15: Output nach der Ausführung von log*

### 2.3.7 Replikation eines entfernten Filesystems (replicate)

Ori ermöglicht das Replizieren eines Filesystems von einem entfernten Rechner. Dazu werden der Host, der Username und der Name des Filesystems des entfernten Rechners angegeben. Ebenfalls ist es hierbei erneut notwendig, einen neuen Ordner zu erstellen und das Filesystem in diesen zu mounten.

```
ori replicate <username>@<host>:<Filesystemname>  
mkdir <Filesystemname>  
orifs <Filesystemname>
```

*Listing 16: Befehle zum Replizieren eines entfernten Filesystems*

### 2.3.8 Herunterladen eines entfernten Filesystems (pull)

Um auf dem neuesten Stand eines entfernten Filesystems zu kommen, ist die Verwendung des Befehls „ori pull“ notwendig. Bei diesem werden genauso wie beim Replizieren der Host, der Username und der Name des Filesystems beim erstmaligen Pullen angegeben. Bei der weiteren Ausführung des Befehls sind diese Angaben nicht mehr notwendig.

```
ori pull <username>@<host>:<Filesystemname>
```

*Listing 17: Befehl zum Herunterladen eines entfernten Filesystems*

Dieser Befehl lädt jedoch nur die Änderungen herunter, ohne das aktuelle Filesystem zu ändern bzw. zu aktualisieren. Möchte man dies jedoch durchführen, so ist der Befehl „ori checkout“ notwendig. Somit kann der Befehl „ori pull“ eher mit „git fetch“ als „git pull“ verglichen werden.

### 2.3.9 Umsteigen auf einen anderen Snapshot (checkout)

Um auf einen anderen Snapshot umzusteigen, ist der Befehl „ori checkout“ notwendig. Ebenso ist es essentiell den Checkout-Befehl nach dem Pullen durchzuführen, da OriFS diesen beim Pull-Befehl nicht inkludiert.

Folgender Befehl ist notwendig, wenn man auf den letzten Snapshot zurückgreifen möchte:

```
ori checkout
```

*Listing 18: Befehl zum Umsteigen auf den letzten Snapshot*

Möchte man jedoch auf einen bestimmten Snapshot umsteigen, ist es zusätzlich notwendig die Commit-ID anzugeben:

```
ori checkout <commit-id>
```

*Listing 19: Befehl zum Umsteigen auf einen bestimmten Snapshot*

### 2.3.10 Fusionierung zweier Versionen (merge)

Um die Fusionierung zweier Versionen mit dem Befehl „ori merge“ durchführen zu können, ist es davor notwendig den Checkout-Befehl anzuwenden. Falls dies vorher nicht ausgeführt wird, wird eine wenig aussagende Fehlermeldung angezeigt:

```
ori merge <commit-id>  
merge failed with an unknown error!
```

*Listing 20: Fehlermeldung bei Durchführung des "merge"-Befehls*

Eine erfolgreiche Anwendung des „merge“-Befehls sieht dementsprechend folgendermaßen aus:

```
ori checkout <commit-id>  
ori merge <commit-id>
```

*Listing 21: Befehle zur Fusionierung zweier Versionen*

### 2.3.11 Einmaliges Replizieren eines Filesystems (graft)

Im Gegensatz zum „replicate“-Befehl kann mittels „ori graft“ ein einmaliges Replizieren eines Filesystems erreicht werden. Falls Änderungen existieren, kann bei Bedarf der „graft“-Befehl erneut durchgeführt werden. Somit stellt „graft“ eine höhere Stufe des Kopierens dar, da ein einfacheres Vergleichen der verschiedenen Versionen ermöglicht wird.

```
ori graft <source-repo> <destination-repo>
```

*Listing 22: Befehl zum einmaligen Replizieren eines Filesystems*

### 2.3.12 Anzeigen der History eines bestimmten Files (filelog)

Möchte man einsehen, welche Änderungen an einem bestimmten File vorgenommen wurden, ist der Befehl „ori filelog“ zu verwenden. Nach der Ausführung des Befehls werden alle Commits, die Änderungen des angegebenen Files beinhalten, angezeigt. Der Befehl ist mit dem „log“-Befehl vergleichbar, jedoch bezieht sich die Ausgabe auf ein einziges File.

```
ori filelog <filename>
```

*Listing 23: Befehl zum Anzeigen der History eines bestimmten Files*

Um dies zu testen, wurde ein File angelegt und dieses committed. Daraufhin wurde der Inhalt des Files aktualisiert und erneut der „snapshot“-Befehl angewendet. Bei der Ausführung des „filelog“-Befehls wurde folgender Output bei uns angezeigt:

```
Commit: 83f2a0921626bc6db67bad70d46c497ae571f5a65de9a0148403d75cb3495d25
Parents:
56d7e90a6299e4bf086eae0cf473c8926ef49a4a6a3e4eed4656e3832660307a
Author:
Date: Thu Apr 21 12:11:27 2016

Changed content of testfile

Commit: 56d7e90a6299e4bf086eae0cf473c8926ef49a4a6a3e4eed4656e3832660307a
Parents:
Author:
Date: Thu Apr 21 12:08:53 2016

Created testfile
```

*Listing 24: Output nach der Ausführung von filelog*

### 2.3.13 Verwaltung von entfernten Verbindungen (remote)

Um Verbindungen zu anderen Servern anzuzeigen, hinzuzufügen und/oder zu löschen, wird der Befehl „ori remote“ angeboten. Dieser ist vergleichbar mit dem gleichnamigen Befehl von Git.

Möchte man alle bestehenden entfernten Verbindungen angezeigt bekommen, ist die Verwendung des folgenden Befehls notwendig:

```
ori remote
```

*Listing 25: Befehl zum Anzeigen aller verfügbaren entfernten Verbindungen*

Nach der Ausführung des oben angeführten Befehls kam bei uns folgender Output:

Name	Path
ID	
dezsyst	ubuntu@10.0.104.168:DezSysRepo

*Listing 26: Output nach Ausführung vom "remote"-Befehl*

Beim Hinzufügen einer neuen Verbindung ist es notwendig den Namen und den Endpoint anzugeben:

```
ori remote add <connection-name> <remote-repository>
```

*Listing 27: Befehl zum Hinzufügen einer entfernten Verbindung*

Zuletzt besteht noch die Möglichkeit, eine bestehende Verbindung zu löschen:

```
ori remote del <connection-name>
```

*Listing 28: Befehl zum Löschen einer bestehenden entfernten Verbindung*

### 2.3.14 Anzeigen von Repository-Informationen (show)

Um die Informationen des aktuellen Repositories anzuzeigen, wird der Befehl „ori show“ angeboten:

```
ori show
```

*Listing 29: Befehl zum Anzeigen der Repository-Informationen*

Nach der Ausführung des Befehls wurde folgendes ausgegeben:

```
--- Repository ---  
Root: /home/ubuntu/.ori/DezSysRepo.ori  
UUID: f8321b7a-48e2-764f-e1ec-edb7aac85dc2  
Version: ORI1.1  
HEAD:  
566f4a6f22b563df7ad1f7e566f4acb42dac1cf227ef8c045623fc867a83b867
```

*Listing 30: Output nach der Ausführung vom "show"-Befehl*

Der Output gibt den Pfad zum Repository, eine eindeutige UUID, die verwendete ORI-Version und den aktuellen Commit aus.

### 2.3.15 Anzeigen der aktuellen Commit-ID (tip)

Möchte man die aktuelle Commit-ID des Repositories erfahren, ist folgender Befehl zu verwenden:

```
ori tip
```

*Listing 31: Befehl zum Anzeigen der aktuellen Commit-ID*

### 2.3.16 Anzeigen der von OriFS verwendeten Variablen (varlink)

Die Anzeige aller Variablen, die OriFS verwendet, erfolgt über folgenden Befehl:

```
ori varlink
```

*Listing 32: Befehl zum Anzeigen der von OriFS verwendeten Variablen*

Folgender Output wurde nach der Ausführung des Befehls angezeigt:

Variable	Value
machtype	unknown
osname	unknown
domainname	(none)
hostname	ubuntu

*Listing 33: Output nach der Ausführung vom "varlink"-Befehl*



## 2.4 Gegenüberstellung der Dateisysteme

Kriterien	OriFS	HDFS	GlusterFS
<i>unterstützte Betriebssysteme</i>	FreeBSD, Mac OS X, Linux [1]	Linux, Windows, BSD, Mac OS X, OpenSolaris [2]	Linux, OS X, NetBSD, OpenSolaris [3]
<i>Dokumentation</i>	wenig zufriedenstellend [6]	sehr detailliert (inkl. Beispielen) [7]	zufriedenstellend (inkl. Troubleshooting) [8]
<i>Architektur</i>	Peer to peer [1]	Master / Slave [4]	Master / Slave [5]
<i>Sicherheitsaspekte</i>	SSH [1]	Verwendung von Kerberos-Sicherheitsaspekten [22]	erlaubt Verwendung des „Transport Layer Security“-Standards (SSL) [21]
<i>Fehlertoleranz</i>	automatische Wiederherstellung von korrupten Dateien [6]	hoch, da Replikate auf verschiedene Racks verteilt werden [11]	RAID5-ähnliches System [15]
<i>verwendete Art von Replikation</i>	Polling (alle 5 Sekunden) [6]	Erstellung der Replikation beim Schreibvorgang [9]	keine vorhanden [9]
<i>Skalierbarkeit</i>	viele Nodes, die nicht unbedingt miteinander kommunizieren [6]	tausende Nodes [10]	Skalierung auf mehrere Peta-Bytes möglich [14]
<i>Snapshots</i>	Read-Only Point-in-time Kopien vom Filesystem [6]	Read-Only Point-in-time Kopien vom Filesystem [12]	Point-in-time Kopien vom GlusterFS Volume [13]
<i>Anwendungsgebiet 1</i>	als Version Control System [6]	zum Managen von „Big Data“ [16]	für Network Attached Storage (NAS) [18]
<i>Anwendungsgebiet 2</i>	geeignet für Offline Operationen [1]	für Write-Once & Read-Many Applikationen [17]	für Media Streaming (große Bandbreiten) [19]

### 3 Github-Link

[https://github.com/serceg-tgm/DezSys13\\_OriFS](https://github.com/serceg-tgm/DezSys13_OriFS)

### 4 Quellen

- [1] Stanford (2016). Ori – A Secure Distributed File System [Online]. Available at: <http://ori.scs.stanford.edu/> [zuletzt abgerufen am 14.04.2016]
- [2] The Apache Software Foundation (2016). What platforms and Java versions does Hadoop run on? [Online]. Available at: [https://wiki.apache.org/hadoop/FAQ#What\\_platforms\\_and\\_Java\\_versions\\_does\\_Hadoop\\_run\\_on.3F](https://wiki.apache.org/hadoop/FAQ#What_platforms_and_Java_versions_does_Hadoop_run_on.3F) [zuletzt abgerufen am 14.04.2016]
- [3] Gluster, Inc (2011). GlusterFS General FAQ [Online]. Available at: [http://www.gluster.org/community/documentation/index.php/GlusterFS\\_General\\_FAQ](http://www.gluster.org/community/documentation/index.php/GlusterFS_General_FAQ) [zuletzt abgerufen am 14.04.2016]
- [4] Dhruba Borthakur (2008). HDFS Architecture Guide [Online]. Available at: [https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html#NameNode\\_and\\_DataNodes](https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html#NameNode_and_DataNodes) [zuletzt abgerufen am 14.04.2016]
- [5] Gluster Docs (2016). Getting started with GlusterFS [Online]. Available at: <https://gluster.readthedocs.org/en/latest/Quick-Start-Guide/Architecture/> [zuletzt abgerufen am 14.04.2016]
- [6] Ali Jose Mashtizadeh, Andrea Bittau, Yifeng Frank Huang, David Mazieres (2013). Replication, History, and Grafting in the Ori File System [Online]. Available at: <http://sigops.org/sosp/sosp13/papers/p151-mashtizadeh.pdf> [zuletzt abgerufen am 14.04.2016]
- [7] The Apache Software Foundation (2016). HDFS User Guide [Online]. Available at: <https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html> [zuletzt abgerufen am 14.04.2016]
- [8] Gluster Docs (2016). Gluster Documentation [Online]. Available at: [http://www.gluster.org/community/documentation/index.php/Main\\_Page](http://www.gluster.org/community/documentation/index.php/Main_Page) [zuletzt abgerufen am 14.04.2016]

- [9] Yuduo Zhou. Large Scale Distributed File System Survey [Online]. Available at: <http://grids.ucs.indiana.edu/ptliupages/publications/Large%20Scale%20Distributed%20File%20System%20Survey.pdf> [zuletzt abgerufen am 14.04.2016]
- [10] The Apache Software Foundation (2016). HDFS User Guide - Security [Online]. Available at: <http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html#Scalability> [zuletzt abgerufen am 14.04.2016]
- [11] The Apache Software Foundation (2016). HDFS User Guide – Rack Awareness [Online]. Available at: [https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html#Rack\\_Awareness](https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html#Rack_Awareness) [zuletzt abgerufen am 14.04.2016]
- [12] The Apache Software Foundation (2016). HDFS Snapshots [Online]. Available at: <https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-hdfs/HdfsSnapshots.html> [zuletzt abgerufen am 14.04.2016]
- [13] Gluster, Inc (2014). Features/Gluster Volume Snapshot [Online]. Available at: [http://www.gluster.org/community/documentation/index.php/Features/Gluster\\_Volume\\_Snapshot](http://www.gluster.org/community/documentation/index.php/Features/Gluster_Volume_Snapshot) [zuletzt abgerufen am 14.04.2016]
- [14] Todd Hoff (2008). Product: GlusterFS [Online]. Available at: <http://highscalability.com/product-glusterfs> [zuletzt abgerufen am 14.04.2016]
- [15] Gluster, Inc (2014). Features/disperse [Online]. Available at: <http://www.gluster.org/community/documentation/index.php/Features/disperse> [zuletzt abgerufen am 14.04.2016]
- [16] Margaret Rouse. Hadoop Distributed File System (HDFS) [Online]. Available at: <http://searchbusinessanalytics.techtarget.com/definition/Hadoop-Distributed-File-System-HDFS> [zuletzt abgerufen am 14.04.2016]
- [17] Jim Scott (2014). Get Real with Hadoop: Read-Write File System [Online]. Available at: <https://www.mapr.com/blog/get-real-hadoop-read-write-file-system> [zuletzt abgerufen am 14.04.2016]

- [18] The University of Pennsylvania (2015).  
Use Case of GlusterFS Distributed File System [Online]. Available at:  
<https://www.isc.upenn.edu/events/use-case-glusterfs-distributed-file-system> [zuletzt abgerufen am 14.04.2016]
- [19] Red Hat, Inc (2016). GlusterFS – Storage for your Cloud [Online].  
Available at: <https://www.gluster.org/>  
[zuletzt abgerufen am 14.04.2016]
- [20] Bitbucket Wiki (2015). Ori File System [Online].  
Available at: <https://bitbucket.org/orifs/ori/wiki/Home>  
[zuletzt abgerufen am 18.04.2016]
- [21] Gluster, Inc (2016). Setting up GlusterFS with SSL/TLS [Online].  
Available at:  
<http://www.gluster.org/community/documentation/index.php/SSL>  
[zuletzt abgerufen am 21.04.2016]
- [22] The Apache Software Foundation (2016).  
Hadoop in Secure Mode [Online]. Available at:  
<https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-common/SecureMode.html> [zuletzt abgerufen am 21.04.2016]

## 5 Befehlsverzeichnis

Listing 1: Installation von OriFS über Packetmanager .....	5
Listing 2: Repository klonen & Ori builden .....	5
Listing 3: Installation von OriFS & Verwaltung des Public-SSH-Keys.....	6
Listing 4: Inhalt des "runtest_config.sh"-Files .....	6
Listing 5: Output beim Durchführen der OriFS-Tests .....	7
Listing 6: Befehle zur Erstellung eines Filesystems .....	8
Listing 7: Befehle zum Löschen eines Filesystems .....	8
Listing 8: Befehl zur Anzeige vorhandener Filesysteme .....	8
Listing 9: Output aller vorhandenen Filesysteme .....	8
Listing 10: Befehl zum Anzeigen vorgenommener Änderungen .....	9
Listing 11: Anzeige der verschiedenen Änderungen .....	9
Listing 12: Befehl zum Anlegen eines Snapshots .....	9
Listing 13: Output nach einer erfolgreichen Erstellung eines Snapshots.....	9
Listing 14: Befehl zum Anzeigen der History .....	10
Listing 15: Output nach der Ausführung von log.....	10
Listing 16: Befehle zum Replizieren eines entfernten Filesystems .....	11
Listing 17: Befehl zum Herunterladen eines entfernten Filesystems .....	11
Listing 18: Befehl zum Umsteigen auf den letzten Snapshot .....	12
Listing 19: Befehl zum Umsteigen auf einen bestimmten Snapshot.....	12
Listing 20: Fehlermeldung bei Durchführung des "merge"-Befehls .....	12
Listing 21: Befehle zur Fusionierung zweier Versionen .....	12
Listing 22: Befehl zum einmaligen Replizieren eines Filesystems .....	13
Listing 23: Befehl zum Anzeigen der History eines bestimmten Files .....	13
Listing 24: Output nach der Ausführung von filelog .....	13
Listing 25: Befehl zum Anzeigen aller verfügbaren entfernten Verbindungen..	14
Listing 26: Output nach Ausführung vom "remote"-Befehl .....	14
Listing 27: Befehl zum Hinzufügen einer entfernten Verbindung .....	14
Listing 28: Befehl zum Löschen einer bestehenden entfernten Verbindung ....	14
Listing 29: Befehl zum Anzeigen der Repository-Informationen .....	15
Listing 30: Output nach der Ausführung vom "show"-Befehl .....	15
Listing 31: Befehl zum Anzeigen der aktuellen Commit-ID.....	15
Listing 32: Befehl zum Anzeigen der von OriFS verwendeten Variablen .....	16
Listing 33: Output nach der Ausführung vom "varlink"-Befehl .....	16