



---

# ORIFS

---

Verteiltes Dateisystem mit ORIFS



MARCH 19, 2015

TGM 5AHITT  
Mair, Vogt

## Table of Contents

Aufgabenstellung.....	2
Installation.....	3
Downloads.....	3
SSH.....	3
ORI .....	3
API .....	6
replicate.....	6
snapshot .....	6
checkout .....	6
graft .....	6
filelog.....	7
list .....	7
log .....	7
merge .....	8
newfs .....	8
pull.....	8
remote .....	8
removefs.....	8
show .....	9
status .....	9
tip.....	9
varlink .....	9
Fehlerberichte .....	10
Quellen .....	11

## Aufgabenstellung

### 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

### Quellen

[1] Ori File System, Stanford Website, online: <http://ori.scs.stanford.edu/>, visited: 2015-03-02

[2] Ori File System, Bitbucket Wiki, online: <https://bitbucket.org/orifs/ori/wiki/Home>, visited: 2015-03-02

[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: 2015-03-02

[5] GlusterFS, <http://www.gluster.org/documentation/howto/HowTo/>, visited: 2015-03-02

## Installation

### Downloads

```
apt-get install scon build-essential pkg-config
apt-get install libboost-dev uuid-dev libfuse-dev libevent-dev libssl-dev
apt-get install libedit-dev
git clone https://bitbucket.org/orifs/ori-orisyncng.git[6]
```

### SSH

Um verbindungen ohne einem passwort über ssh zu ermöglichen werden folgende Befehle benötigt.

Zuerst erstellt man einen Key mittels diesem Befehl:

```
ssh-keygen -t rsa
```

Hier muss man bei den Eingabeaufforderungen einfach den Default nehmen (Keine passphrase).

Als nächstes wollen wir mittels ssh ein Directory erstellen.

```
ssh b@B mkdir -p .ssh
```

Bevor man jetzt sich ohne Passwordeingabe mit einem anderen Computer verbinden kann muss man noch den public key in die `authorized_keys` schreiben.

```
cat .ssh/id_rsa.pub | ssh b@B 'cat >> .ssh/authorized_keys'
```

Hier muss man noch ein letztes mal das Passwort eingeben.

Ab jetzt kann man mittels ssh sich automatisch ohne Passwordeingabe verbinden. [5]

### ORI

Hier werden die Voreinstellungen von ORI angewendet.

```
cd ori-orisyncng
scons
vim SConstruct
--> change PREFIX to /usr/local/ instead of /usr/local/bin
sudo scons install
```

exchange `id_rsa.pub` to `authorized_keys` (look into the SSH headline)

```
orisync init
```

Ab hier wird ORI und die Repositories konfiguriert.[6]

----- Wenn kein externes repository existiert nimm diesen Teil -----

```
ori newfs MyRepo          #Definieren eines neuen File Systems
orisync add /home/user/.ori/MyRepo.ori #Hinzufügen des Repos
orisync                  #Vorbereiten der .ori Datei
orisync list              # Überprüft ob du bereits gemounted bist oder nicht
```

```
mkdir ori/MyRepo          #Erstellen eines neuen Ordners
orifs /home/user/ ori-orisyncng /ori/MyRepo #Definiert den Ort als File System
mount                      #Versucht dich auf das Repo zu Mounten[6]
```

```
tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)
none on /run/lock type tmpfs (rw,nodev,noexec,nosuid,size=5242880)
none on /run/shm type tmpfs (rw,nosuid,nodev)
none on /run/user type tmpfs (rw,nodev,noexec,nosuid,size=104857600,mode=0755)
none on /sys/fs/pstore type pstore (rw)
systemd on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,noexec,nodev,none,name=systemd)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,user=schueler)
/dev/sr0 on /media/schueler/CDROM type iso9660 (ro,nodev,nosuid,uid=1000,gid=1000,icharset=utf8,mode=0400,dmode=0500,uhelper=udisks2)
/dev/sr1 on /media/schueler/Ubuntu 14.10 amd64 type iso9660 (ro,nodev,nosuid,uid=1000,gid=1000,icharset=utf8,mode=0400,dmode=0500,uhelper=udisks2)
orifs on /home/schueler/ori/MyRepo type fuse.orifs (rw,nosuid,nodev,user=schueler)
/dev/fd0 on /media/floppy0 type vfat (rw,nosuid,nodev,utf8,user)
orifs on /home/schueler/ori-orisyncng/ori/MyRepo type fuse.orifs (rw,nosuid,nodev,user=schueler)
schueler@ubuntu:~/ori-orisyncng/ori/MyRepo$ orisync list
Repo                                Mounted                                Peers

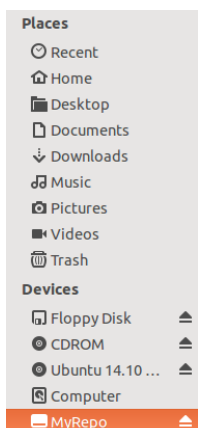
/home/schueler/.ori/MyRepo.ori      true

schueler@ubuntu:~/ori-orisyncng/ori/MyRepo$
```

orisync list #Überprüft ob du bereits gemounted bist oder nicht

```
wolf@ubuntu:~/ori-orisyncng$ orisync list
Repo                                Mounted                                Peers

/home/wolf/.ori/MyRepo.ori          true
```



Nach dem Mounten sollte im Arbeitsplatz das File System erscheinen und auswählbar sein. [6]

----- Wenn bereits ein externes repository existiert nimm diesen Teil -----

ori replicate mike@vmxubuntu:MyRepo #Herunterladen der .ori Datei eines externen Repos

```
schueler@ubuntu:~$ cd ori-orisyncng/
schueler@ubuntu:~/ori-orisyncng$ ori replicate wolf@10.0.104.169:MyRepo
Cloning from wolf@10.0.104.169:MyRepo to /home/schueler/.ori/MyRepo.ori
```

orisync add /home/mike/.ori/MyRepo.ori #Hinzufügen des Repos

orisync #Vorbereiten der .ori Datei

orisync list #Überprüft ob du bereits gemounted bist oder nicht

mkdir ori/MyRepo #Erstellen eines neuen Ordners

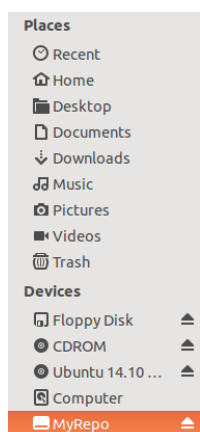
orifs /home/user/ ori-orisyncng /ori/MyRepo #Definiert den Ort als File System

mount #Versucht dich auf das Repo zu Mounten[6]

```
tmpfs on /run type tmpfs (rw,noexec,nosuid,size=10%,mode=0755)
none on /run/lock type tmpfs (rw,nodev,noexec,nosuid,size=5242880)
none on /run/shm type tmpfs (rw,nosuid,nodev)
none on /run/user type tmpfs (rw,nodev,noexec,nosuid,size=104857600,mode=0755)
none on /sys/fs/pstore type pstore (rw)
systemd on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,noexec,nodev,none,name=systemd)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,user=schueler)
/dev/sr0 on /media/schueler/CDROM type iso9660 (ro,nodev,nosuid,uid=1000,gid=1000,ioccharset=utf8,mode=0400,dmode=0500,uhelper=udisks2)
/dev/sr1 on /media/schueler/Ubuntu 14.10 amd64 type iso9660 (ro,nodev,nosuid,uid=1000,gid=1000,ioccharset=utf8,mode=0400,dmode=0500,uhelper=udisks2)
orifs on /home/schueler/ori/MyRepo type fuse.orifs (rw,nosuid,nodev,user=schueler)
/dev/fd0 on /media/floppy0 type vfat (rw,nosuid,nodev,utf8,user=schueler)
orifs on /home/schueler/ori-orisyncng/ori/MyRepo type fuse.orifs (rw,nosuid,nodev,user=schueler)
schueler@ubuntu:~/ori-orisyncng/ori/MyRepo$ orisync list
Repo Mounted Peers
/home/schueler/.ori/MyRepo.ori true
schueler@ubuntu:~/ori-orisyncng/ori/MyRepo$
```

orisync list #Überprüft ob du bereits gemounted bist oder nicht

```
wolf@ubuntu:~/ori-orisyncng$ orisync list
Repo Mounted Peers
/home/wolf/.ori/MyRepo.ori true
```



Nach dem Mounten sollte im Arbeitsplatz das File System erscheinen und auswählbar sein. [6]

## API

Mithilfe des Befehles "help" welches man auch indem man kein Command angibt aufruft, zeigt alle möglichen Befehle an. Diese Befehle ermöglichen eine aktive Verwaltung des File Systems.

```
Available commands:
filelog      Display a log of change to the specified file
help         Show help for a given topic
list         List local file systems
log          Display a log of commits to the repository
merge        Merge two heads
newfs        Create a new file system
pull         Pull changes from a repository
remote       Remote connection management
removefs     Remove a local replica
replicate     Create a local replica
show         Show repository information
snapshot     Create a repository snapshot
snapshots    List all snapshots available in the repository
status       Scan for changes since last commit
tip          Print the latest commit on this branch
varlink      Get, set, list varlink variables
```

Die Befehle die laut dem help Befehl zur Verfügung stehen.

### replicate

Replicate ermöglicht das Klonen eines bereits existierenden File Systems welches sich nicht auf dem lokalen Rechner befindet. Im weiteren Verlauf muss das so heruntergeladene .ori File noch eingebunden werden.

```
schueler@ubuntu:~$ cd ori-orisyncng/
schueler@ubuntu:~/ori-orisyncng$ ori replicate wolf@10.0.104.169:MyRepo
Cloning from wolf@10.0.104.169:MyRepo to /home/schueler/.ori/MyRepo.ori
```

### snapshot

Snapshots speichern den derzeitigen Zustand des File Systems und laden die Änderungen auch auf den lokalen bzw. remote Speicher hoch. Es ist auch möglich Snapshots Namen zu geben um sie später einfacher wieder laden zu können. [3]

```
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori status
M /Secret.txt
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori snapshot
Committed 853ee23299f3c21ababc7a68d5cd600ba52abcd4b6b14429fdbcb471aed93b62d
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori status
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$
```

### checkout

Nicht in der Befehlsliste von ORI vorhanden.

### graft

Nicht in der Befehlsliste von ORI vorhanden.

## filelog

Filelog gibt den User Informationen über die Änderungen einer einzigen Datei. In diesem Fall ist es nur log da es nur eine einzige Datei gibt die geloggt werden konnte. Aufgerufen wäre der Befehl mittels `ori filelog Secret.txt`

```
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori log
Commit:      853ee23299f3c21ababc7a68d5cd600ba52abcd4b6b14429fdbcb471aed93b62d
Parents:     a64e4ca067f19e5360e82fc10a7738441972e2e1ed45d9f37f8025d05bacddd8
Tree:        a61b89097ffad3fcc9c7873940c407c0c82218c9a3a6455f2b3ccac1d794adf1
Author:      wolf,,,
Date:        Thu Mar 19 16:17:43 2015

No message.

Commit:      a64e4ca067f19e5360e82fc10a7738441972e2e1ed45d9f37f8025d05bacddd8
Parents:     3d6b1f36e5700e0408db3954881ad097632df32928221a226216dbc883b986ec
Tree:        3d6b1f36e5700e0408db3954881ad097632df32928221a226216dbc883b986ec
Author:      wolf,,,
Date:        Thu Mar 19 16:15:51 2015

No message.
```

## list

Die List Befehl listet alle File Systeme auf mit denen er verbunden ist. Er zeigt also sowohl interne als auch externe ori Repositorien an.

```
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori list
Name                               File System ID
MyRepo                             b72fb375-f333-491e-8ddd-0ab9c2896f26
```

Ein lokales Repository.

```
schueler@ubuntu:~/ori-orisyncng$ ori list
Name                               File System ID
MyRepo                             b72fb375-f333-491e-8ddd-0ab9c2896f26
```

Ein externes Repository.

## log

Log gibt den User eine Übersicht über alle Commits die in dem Repository ausgeführt worden sind.

```
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori log
Commit:      853ee23299f3c21ababc7a68d5cd600ba52abcd4b6b14429fdbcb471aed93b62d
Parents:     a64e4ca067f19e5360e82fc10a7738441972e2e1ed45d9f37f8025d05bacddd8
Tree:        a61b89097ffad3fcc9c7873940c407c0c82218c9a3a6455f2b3ccac1d794adf1
Author:      wolf,,,
Date:        Thu Mar 19 16:17:43 2015

No message.

Commit:      a64e4ca067f19e5360e82fc10a7738441972e2e1ed45d9f37f8025d05bacddd8
Parents:     3d6b1f36e5700e0408db3954881ad097632df32928221a226216dbc883b986ec
Tree:        3d6b1f36e5700e0408db3954881ad097632df32928221a226216dbc883b986ec
Author:      wolf,,,
Date:        Thu Mar 19 16:15:51 2015

No message.
```



## merge

Ich mag keine merges und werde sie wahrscheinlich auch nie mögen. Aufgrund von eigener Faulheit und des Aufwandes welcher meist für einen erfolgreichen merge benötigt wird hier kein example Bild (Ich möchte wenigstens noch das Gefühl haben einen eigenen Willen zu haben).

## newfs

```
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori newfs stuff
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori list
Name                               File System ID
MyRepo                             b72fb375-f333-491e-8ddd-0ab9c2896f26
stuff                              a9091710-b651-421f-b5ca-8ad4c6268d95
```

## pull

Pull ermöglicht das Updaten der lokal gespeicherten temporären Daten die von einem remoten File System kommen. Man wird nicht durch ori status über die neueren Daten auf dem remote File System informiert.

```
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ nano Secret.txt
Error reading /home/wolf/.nano_history: Permission denied
Press Enter to continue starting nano.

wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori snapshot
Committed 032577ceceaef24a3d3663de11c2a4140ef36893891eace5280abc3ac5ee4a6b
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori status
```

Das hochladen einer Änderung auf dem remote FS.

```
schueler@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori pull
Pulled up to 032577ceceaef24a3d3663de11c2a4140ef36893891eace5280abc3ac5ee4a6b
```

Das "pullen" oder auch herunterladen der neuen Daten.

## remote

Nicht getestet, ehrlicherweise wirklich vergessen ... Nachholen der Tests erweist sich als schwierig da sich mein Kollege bereits zuhause auf ein Referat vorbereitet.

## removefs

Removefs erlaubt das löschen lokaler File Systeme. Diese sind dann nicht mehr zugänglich und müssen vorher erst neu erstellt werden.

```
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori newfs stuff
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori list
Name                               File System ID
MyRepo                             b72fb375-f333-491e-8ddd-0ab9c2896f26
stuff                              a9091710-b651-421f-b5ca-8ad4c6268d95
```

Erstellen eines neuen File Systems namens Stuff.

```
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori removefs stuff
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori list
Name                               File System ID
MyRepo                             b72fb375-f333-491e-8ddd-0ab9c2896f26
```

Löschen des neuen File Systems.

### show

Show ist ein Befehl welches dem User derzeitige Informationen über das Repository ausgibt. In diesem Fall die Version, der Root, die UUID und den HEAD.

```
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori show
--- Repository ---
Root: /home/wolf/.ori/MyRepo.ori
UUID: b72fb375-f333-491e-8ddd-0ab9c2896f26
Version: ORI1.1
HEAD: 853ee23299f3c21ababc7a68d5cd600ba52abcd4b6b14429fdb471aed93b62d
```

### status

Zeigt dem User welche Dateien noch committed werden müssen.

```
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori status
M /Secret.txt
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori snapshot
Committed 853ee23299f3c21ababc7a68d5cd600ba52abcd4b6b14429fdb471aed93b62d
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori status
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$
```

### tip

Tip zeigt dem User den letzten Commit den er durchgeführt hat. Allerdings wird nur die Referenz angezeigt ohne jegliche weitere Informationen über den Commit.

```
wolf@ubuntu:~/ori-orisyncng/ori/MyRepo$ ori tip
853ee23299f3c21ababc7a68d5cd600ba52abcd4b6b14429fdb471aed93b62d
```

### varlink

Aus Angst vor Änderung einer falschen Variable nicht durchgeführt. Dieses Programm wurde bereits oft genug Installiert und gelöscht.

## Fehlerberichte

Kommen wir zu einem sehr schönen Teil der HÜ. Nämlich die Fehlerbehebung die wir alle unterlaufen mussten. Erfreulicherweise ist dieses Programm, freundlich gesagt, sehr "Sensibel".

Jegliche Schließung der Virtuellen Maschine erzeugt, beim erneuten Versuch sich an die bereits vorbereiteten Umgebung zu verbinden ,diesen Fehler. Es ist sehr schön zu wissen das sie daran arbeiten, allerdings ist es nicht schön Anweisungen zu schreiben die nicht funktionieren.

```
An error has occurred!

Problem: orifs may have previously exited uncleanly

Solution:
Check the .ori/tmp/fuse directory for any files that may not have been
saved to the file systems store.  You can copy or move these files to
another location. Then delete the .ori/tmp/fuse directory and all
remaining files.

Notes: This is a known bug and will be fixed in the future.
```

Der Lösungsweg den ich mir ausgewählt habe ist folgender:

Schritt 1: Löschen aller ORI befangenen Archive (inklusive des unsichtbaren .ori/ Archives).

Schritt 2: Zurück zu der Installation von ORI, SSH muss dabei nicht neu konfiguriert werden.

Ein weiterer schöner Fehler ist das wiederholte initialisieren von orisync in einem falschen Ordner, welcher mittels des Befehles orisync init aktiv wird. Wie man bereits sehen kann spammt die Console eine Anfrage die nicht aufzuhören scheint. Sie kann nicht mittels einer Eingabe oder CTRL + C gestoppt werden.

[illegible]

Der Lösungsweg den ich mir ausgewählt habe ist folgender:

Schritt 1: Öffnen eines neuen Konsolen-tabs.

## Schritt 2: Ausführen des Befehles killall orisync

### Schritt 3: Weiterführen der Installation/Vorbereitung

Desweiteren sollte man bei der Automatischen Textvollendung aufpassen. Oft wird bei der automatischen Textvollendung ein / am Ende beigefügt, auch dies kann zu einer Fehlerausgabe führen.

## Quellen

- [1] Ori File System, Stanford Website, online: <http://ori.scs.stanford.edu/>, visited: 2015-03-02
- [2] Ori File System, Bitbucket Wiki, online: <https://bitbucket.org/orifs/ori/wiki/Home>, visited: 2015-03-02
- [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: 2015-03-02
- [5] GlusterFS, <http://www.gluster.org/documentation/howto/HowTo/>, visited: 2015-03-02
  
- [6]Titel:** SSH Login without password; **Link:** [http://www.linuxproblem.org/art\\_9.html](http://www.linuxproblem.org/art_9.html); **Author:** Mathias Ketner; **Zuletzt geändert:** 21. April 2014; **Zuletzt gesehen:** 19. März 2015
  
- [7]Titel:** Re: DezSys 7 - Synchronisation & Replikation mit Ori; **Link:** <https://elearning.tgm.ac.at/mod/forum/discuss.php?d=1928>; **Author:** Michael Borko; **Zuletzt geändert:** 17. März 2015; **Zuletzt gesehen:** 19. März 2015