

ORIFS

Verteiltes Dateisystem mit ORIFS



March 19, 2015

TGM 5AHITT

Mair, Vogt

Table of Contents

[Aufgabenstellung 2](#_Toc414567720)

[Installation 3](#_Toc414567721)

[Downloads 3](#_Toc414567722)

[SSH 3](#_Toc414567723)

[ORI 3](#_Toc414567724)

[API 6](#_Toc414567725)

[replicate 6](#_Toc414567726)

[snapshot 6](#_Toc414567727)

[checkout 6](#_Toc414567728)

[graft 6](#_Toc414567729)

[filelog 7](#_Toc414567730)

[list 7](#_Toc414567731)

[log 7](#_Toc414567732)

[merge 8](#_Toc414567733)

[newfs 8](#_Toc414567734)

[pull 8](#_Toc414567735)

[remote 8](#_Toc414567736)

[removefs 8](#_Toc414567737)

[show 9](#_Toc414567738)

[status 9](#_Toc414567739)

[tip 9](#_Toc414567740)

[varlink 9](#_Toc414567741)

[Fehlerberichte 10](#_Toc414567742)

[Quellen 12](#_Toc414567743)

# 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](http://sigops.org/sosp/sosp13/), November 2013. [Paper](http://dl.acm.org/ft_gateway.cfm?id=2522721&ftid=1403940&dwn=1).  
[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 scons 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 Passworteingabe verbinden. **[5]**

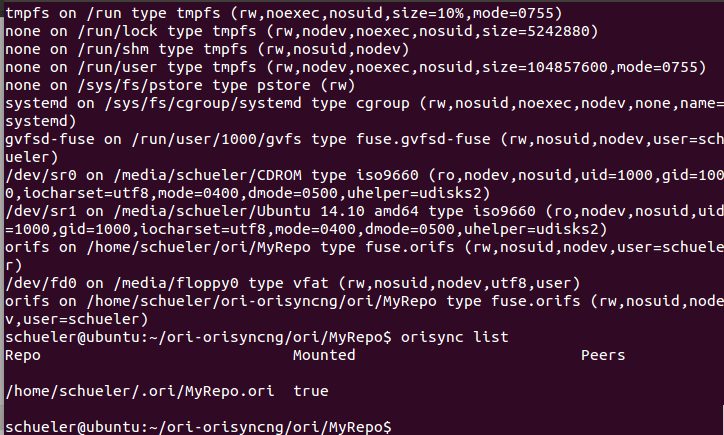
### ORI

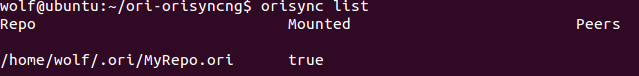
Hier werden die Voreinstellungen von ORI angewendet.

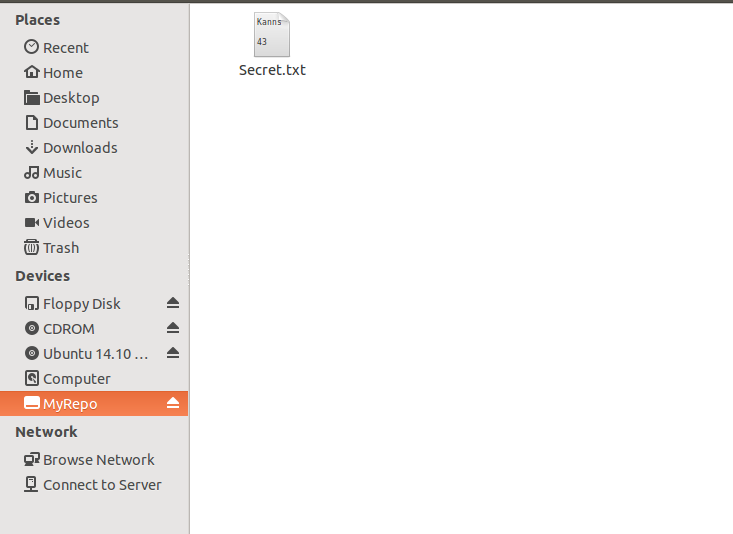
cd ori-orisyncng  
scons  
vim SConstruct  
--> change PREFIX to /usr/local/ instead of /usr/local/bin  
sudo scons install  
  
exchange id\_rsa.pub to authorized\_keys (look into the SSH headline)  
  
orisync init

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]**

  
orisync list #Überprüft ob du bereits gemounted bist oder nicht

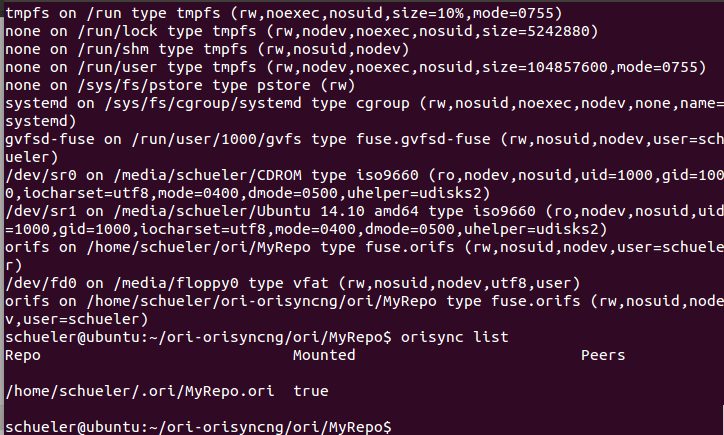
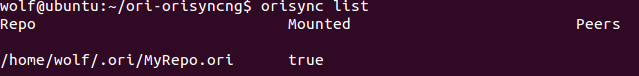


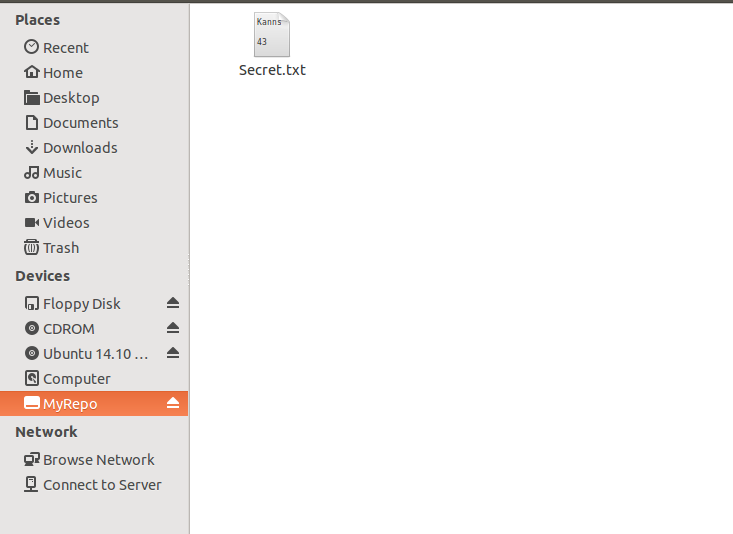


Nach dem Mounten sollte im Arbeitsplatz das File System erscheinen und auswählbar sein. **[6]**

-------- Wenn bereits ein externes repository existiert nimm diesen Teil ---------  
ori replicate mike@vmxubuntu:MyRepo #Herunterladen der .ori Datei eines externen Repos

C:\Users\Wolf\Dropbox\5AHITT\GitProjects\Ori\Img\replicate.PNG  
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]**

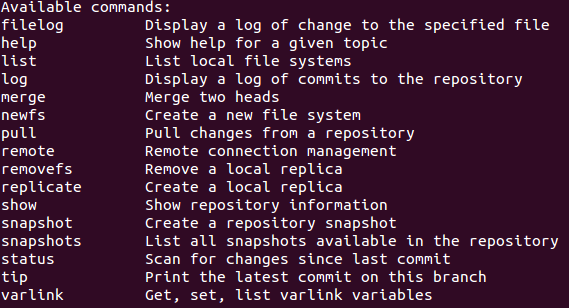
  
orisync list #Überprüft ob du bereits gemounted bist oder nicht  




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.



Die Befehle die laut dem help Befehl zur Verfügung stehen.

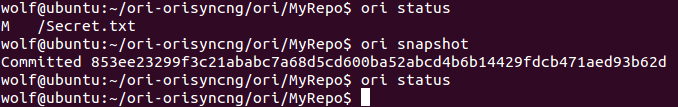
### replicate

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.

C:\Users\Wolf\Dropbox\5AHITT\GitProjects\Ori\Img\replicate.PNG

### 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]**



### checkout

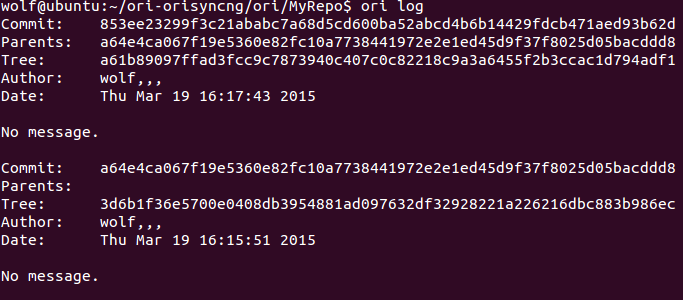
Nicht in der Befehlsliste von ORI vorhanden.

### 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



### 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.

C:\Users\Wolf\Dropbox\5AHITT\GitProjects\Ori\Img\list.PNG

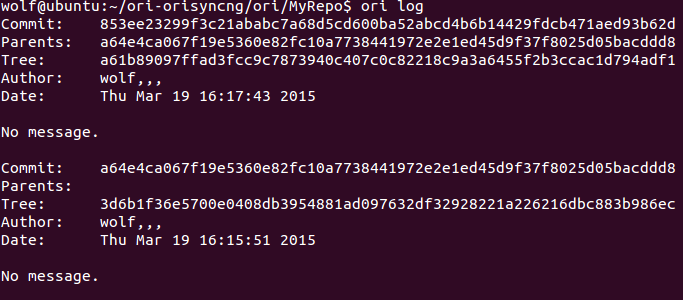
Ein lokales Repository.

C:\Users\Wolf\Dropbox\5AHITT\GitProjects\Ori\Img\orilist.PNG

Ein externes Repository.

### log

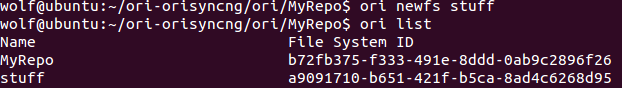
Log gibt den User eine Übersicht über alle Commits die in dem Repository ausgeführt worden sind.



### merge

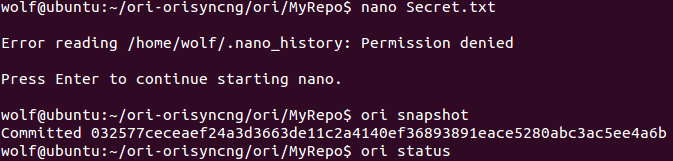
Ich mag keine merges und werde sie wahrscheinlich auch nie mögen. Aufgrund von eigener Faulheit und des Aufwandes welcher meist für einen erfolgreichen merge benötigt wird hier kein example Bild (Ich möchte wenigstens noch das Gefühl haben einen eigenen Willen zu haben).

### newfs



### 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.



Das hochladen einer Änderung auf dem remote FS.

C:\Users\Wolf\Dropbox\5AHITT\GitProjects\Ori\Img\Oripull.PNG

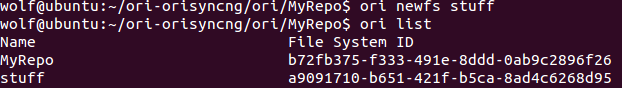
Das "pullen" oder auch herunterladen der neuen Daten.

### remote

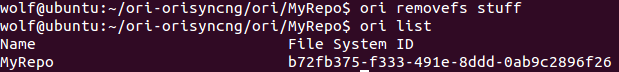
Nicht getestet, ehrlicherweise wirklich vergessen ... Nachholen der Tests erweist sich als schwierig da sich mein Kollege bereits zuhause auf ein Referat vorbereitet.

### removefs

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



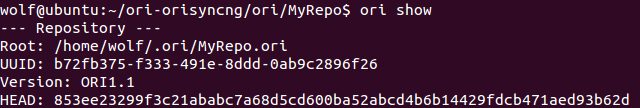
Erstellen eines neuen File Systems namens Stuff.



Löschen des neuen File Systems.

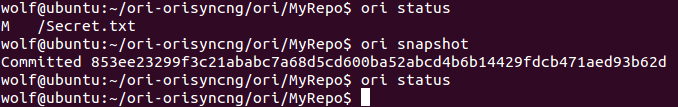
### show

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.



### status

Zeigt dem User welche Dateien noch commited werden müssen.



### 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.

C:\Users\Wolf\Dropbox\5AHITT\GitProjects\Ori\Img\tip.PNG

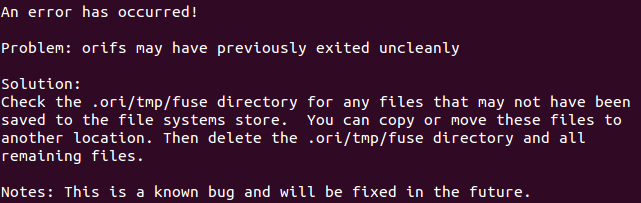
### varlink

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

# Fehlerberichte

Kommen wir zu einen 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.

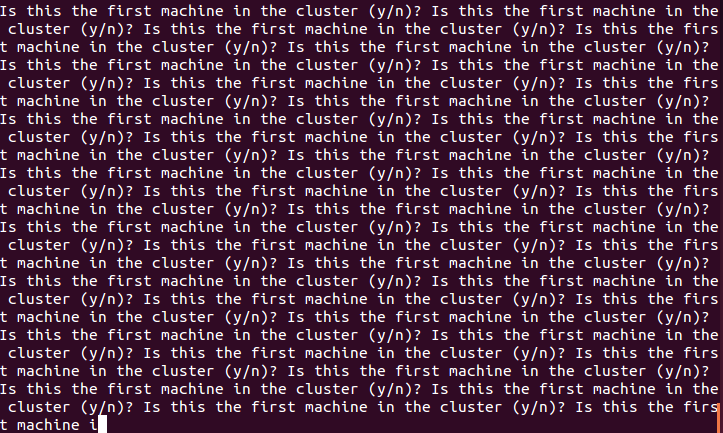


Der Lösungsweg den ich mir ausgewählt habe ist folgender:

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.



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

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

# Verschiedene DFS

| **Client** | **Written in** | **License** | **Access API** |
| --- | --- | --- | --- |
| [**Ceph**](http://en.wikipedia.org/wiki/Ceph_(file_system)) | C++ | LGPL | librados (C, C++, Python, Ruby), [S3](http://en.wikipedia.org/wiki/Amazon_S3), Swift, [FUSE](http://en.wikipedia.org/wiki/Filesystem_in_Userspace) |
| [**FhGFS**](http://en.wikipedia.org/wiki/FhGFS) | C / C++ | FRAUNHOFER FS (FhGFS) EULA,[[2]](http://en.wikipedia.org/wiki/Comparison_of_distributed_file_systems#cite_note-2)  GPLv2 client | [POSIX](http://en.wikipedia.org/wiki/Posix#POSIX.1) |
| [**GlusterFS**](http://en.wikipedia.org/wiki/GlusterFS) | C | GPLv3 | libglusterfs, [FUSE](http://en.wikipedia.org/wiki/Filesystem_in_Userspace), NFS, SMB, Swift, libgfapi |
| [**Quantcast File System**](http://en.wikipedia.org/wiki/Quantcast_File_System) | C | Apache License 2.0 | C++ client, [FUSE](http://en.wikipedia.org/wiki/Filesystem_in_Userspace) (C++ server: MetaServer and ChunkServer are both in C++) |
| [**Lustre**](http://en.wikipedia.org/wiki/Lustre_(file_system)) | C | GPLv2 | [POSIX](http://en.wikipedia.org/wiki/Posix#POSIX.1), liblustre, [FUSE](http://en.wikipedia.org/wiki/Filesystem_in_Userspace) |
| [**OpenAFS**](http://en.wikipedia.org/wiki/OpenAFS) | C | IBM Public License | [Virtual file system](http://en.wikipedia.org/wiki/Virtual_file_system), [Installable File System](http://en.wikipedia.org/wiki/Installable_File_System) |
| [**Tahoe-LAFS**](http://en.wikipedia.org/wiki/Tahoe-LAFS) | Python | [GNU GPL](http://en.wikipedia.org/wiki/GNU_GPL) 2+ and other[[3]](http://en.wikipedia.org/wiki/Comparison_of_distributed_file_systems#cite_note-3) | HTTP (browser or [CLI](http://en.wikipedia.org/wiki/Command-line_interface)), [SFTP](http://en.wikipedia.org/wiki/SSH_File_Transfer_Protocol), [FTP](http://en.wikipedia.org/wiki/File_Transfer_Protocol), [FUSE](http://en.wikipedia.org/wiki/Filesystem_in_Userspace) via [SSHFS](http://en.wikipedia.org/wiki/SSHFS), pyfilesystem |
| [**HDFS**](http://en.wikipedia.org/wiki/HDFS) | Java | Apache License 2.0 | Java and C client, HTTP |
| [**XtreemFS**](http://en.wikipedia.org/wiki/XtreemFS) | Java, C++ | BSD License | libxtreemfs (Java, C++), [FUSE](http://en.wikipedia.org/wiki/Filesystem_in_Userspace) |
| [**MooseFS**](http://en.wikipedia.org/wiki/Moose_File_System)[[4]](http://en.wikipedia.org/wiki/Comparison_of_distributed_file_systems#cite_note-4) | Python,C | GPL |  |
| **Ori**[[5]](http://en.wikipedia.org/wiki/Comparison_of_distributed_file_systems#cite_note-5) | C, C++ | MIT | libori, [FUSE](http://en.wikipedia.org/wiki/Filesystem_in_Userspace) |

[3]

# Gegenüberstellungstabelle

|  |  |  |  |
| --- | --- | --- | --- |
|  | ORIFS | HDFS | GLUSTERFS |
| Ausfallsicher | Niedrig | Sehr Hoch(Data auf mehrere Nodes Kopiert) | Sehr Hoch |
| Skalierbarkeit | ? | Möglich von 1 Server auf tausende Maschinen zu Skalieren | Sehr Hoch |
| Sicherheit | Arbeitet nur mit ssh(recht sicher) | Eingebaute Authentication | Nur IP/port based Authentifizierung |
| Dokumentiert | Kaum | Sehr Gut | Sehr Gut |
| Verfügbarkeit | ? | Sehr hoch (Manuel Failover) | Sehr hoch (Manuel Failover) |
| Kommunikation | Peer to Peer | TCP/IP sockets | TCP/IP, InfiniBand oder Sockets Direct Protocol |
| Open Source | Ja | Ja | Ja |
|  |  |  |  |

# 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](http://sigops.org/sosp/sosp13/), November 2013. [Paper](http://dl.acm.org/ft_gateway.cfm?id=2522721&ftid=1403940&dwn=1).  
[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; **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

[8] ComparisionsDistributed File Systems <http://en.wikipedia.org/wiki/Comparison_of_distributed_file_systems>