



HBase auf DXRAM

Einstiegspunkte für DXRAM in Hadoop & HBase

in Bearbeitung!

Department of Computer Science
Heinrich-Heine-University Düsseldorf, Germany

22. November 2018



- HBase auf DXRAM
- Lösungswege
- Umsetzung
- Fazit



DXRAM benutzen

- Einbindung in andere Software ausprobieren
- zeigen, dass es echte Alternative sein kann
- Popularität erhöhen





Beispiel: HBase

- noSQL mit BASE statt ACID
- RegionServer als Memory Cache
- HDFS als langsame Persistenz-Schicht
- Balance und Config wichtig (read, write, RAM, flush, compression)

Warum nicht gleich DXRAM als verteilten Speicher nutzen?



- Ignite wie DXRAM verteilter Speicher
- Ignite SQL: ACID und nicht BASE
- Ignite nutzt auch HDFS als Persistenz-Schicht
- Ignite und Hbase: Ignite FS Connector zu Hadoop

Idee 1

Idee 1: DXRAM auch als verteiltes Dateisystem anbieten und Connector für Hadoop machen.





Idee 1: DxramFs Connector

Pro

- Anwender muss auf HBase und Hadoop Seite nichts umprogrammieren
- alle Hadoop Anwendungen können es nutzen
- Host basierte Prozesssplittung durch Hadoop ist möglich



Idee 1: DxramFs Connector

Contra

Mal eben HDFS nach programmieren :o/



Idee 2

Idee 2: DXRAM zu einem mountfähigen Medium machen mit `libfuse`.



Idee 2: mount DxramFs

Pro

- Anwender muss nicht umprogrammieren
- nicht nur Hadoop könnte das nutzen





Idee 2: mount DxramFs

Contra

- Verteilung der Daten unklar
- Hadoop weiss echten Speicherort nicht mehr
- Performance Probleme bei libfuse
- auch hier muss ein Verteiltes Dateisystem Programmiert werden



Idee 3

Idee 3: HBase Replacement auf der Basis der Thrift Schnittstelle für einen Client.





Idee 3: DXRAM.Base

Pro

- kein Umweg über Implementierung eines Dateisystem oder Hadoop
- vermutlich die effizienteste Art
- Prozesssplittung von Hadoop losgelöst



Contra

- unklar, wie HBase und Hadoop Community darauf reagiert
- vermutlich wird man auf Hadoop nicht verzichten wollen

Ist es einfacher HDFS oder HBase nachzuprogrammieren?

Idee 4: Wie Ignite oder Alluxio eine Prozessverarbeitung vorbei an Hadoop konstruieren. Konkret: RegionServer ist eine DXRAM App.



Idee 4: DXRAM RegionServer

Pro

- Lösung auf HBase zugeschnitten
- weniger Konflikte als bei einem HBase Replacement zu erwarten
- kein Dateisystem, was zu implementieren wäre
- evtl. nur eine minimale Anpassung nötig





Idee 4: DXRAM RegionServer

Contra

- tiefes Verständnis von HBase Quellcode nötig
- HBase Updates muss man evtl. aufwändig einpflegen
- kein Vorteil für andere Hadoop Projekte
- unklar, ob RegionServer ganz von Hadoop trennbar ist



Die Wahl fiel auf die Lösung, wo HBase und Hadoop unberührt bleiben, und NUR eine HDFS kompatibler Connector beigefügt wird (Idee 1).



- Connector in Hadoop nutzt DXNET um FS Operationen durchzuführen (CRUD)
- DXRAM ist nicht in Hadoop
- DxramFs App bietet Connector FS API an

Projekt scheiterte primär an Debugging der Serialisierung reiner Attribut-Klassen.



Umsetzung: Fail

Grafik





Umsetzung: Serialisierung

- Initialisierung, ändernde Größen bei Updates
- gut wäre IDL wie bei Apache Thrift





Umsetzung: Schlauer sein

Hinterher ist man schlauer: Anstatt multi-Peer und DXRAM Entwicklung auf zu schieben, wäre z.B. als erster Ansatz ein Multi-FTP Connector (aus dem bestehenden) gut gewesen. So hätte man Fragen des Prozesshandlings von HBase auf Basis von Hostnamen bereits ausprobieren können.





Umsetzung: Schlauer sein

Unelegant: DXNET eigentlich nur zum Transfer auf dem selben Host genutzt, um zwischen Hadoop und DXRAM Infos austauschen zu lassen.



Fertig: FS Aufbau, Ordner Operationen





Offen

- Fehler bei Chunk-Speicherung klären
- Begonnen: create, open, flush, In- und OutStream
- kleiner Bugs (siehe Webseite)
- Handling von Mehrfachanfragen
- Chunk sperren, Hadoop Unittests
- Tests mit MapReduce, Hadoop Multinode, HBase
- Performance Tests



Hadoops Prozess- bzw. Ressourcen-Management ist zu stark an HDFS und dessen Blockverteilung gekoppelt! Ignite und Alluxio konstruierten daher auch ein Replacement! Außerdem: Ist es nicht leichter HBase mit DXRAM nachzubauen, anstatt DXRAM zu einem verteilten Dateisystem zu machen? Vermutlich Ja. -> **Apache Thrift**

Aber: Alle von mir gefundenen Projekte werben mit einer EINBINDUNG in Hadoop, nicht aber mit einem ERSATZ. Ein art **DXRAM.Base** wäre aber sogar ein HBase Ersatz! Konkrete Anwendungsfälle, wo auf Hadoop bei Verwendung von HBase verzichtet werden kann, sollten in Zukunft gesucht werden.