Project report: Entry Points for DXRAM in Hadoop & HBase

To get or to become a part of the Hadoop tart

Jochen Peters

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

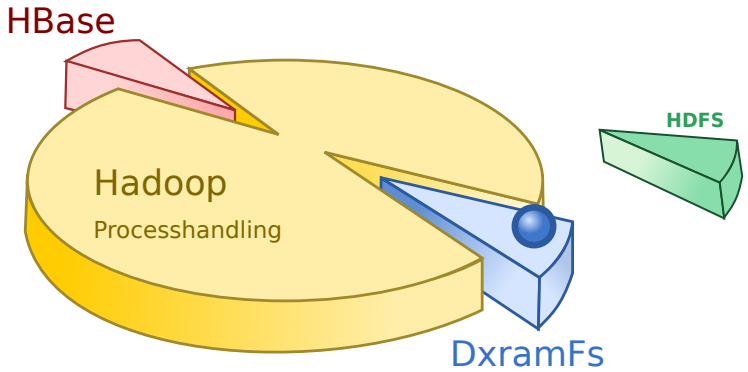
11. Dezember 2018

DXRAM

Contens

- Motivation
- Wie machen es Andere?
- Lösungswege
- Umsetzung
- Fazit

Motivation



DXRAM usage

- join to other popular Distrubuted Systems
- DXRAM as alternative storage system
- enlarge DXRAM popularity

DXRAM usage

Idea: become a part of a popular project

- Hadoop
- HBase

Vision: remove HDFS access in HBase by DXRAM

Excursion Hadoop

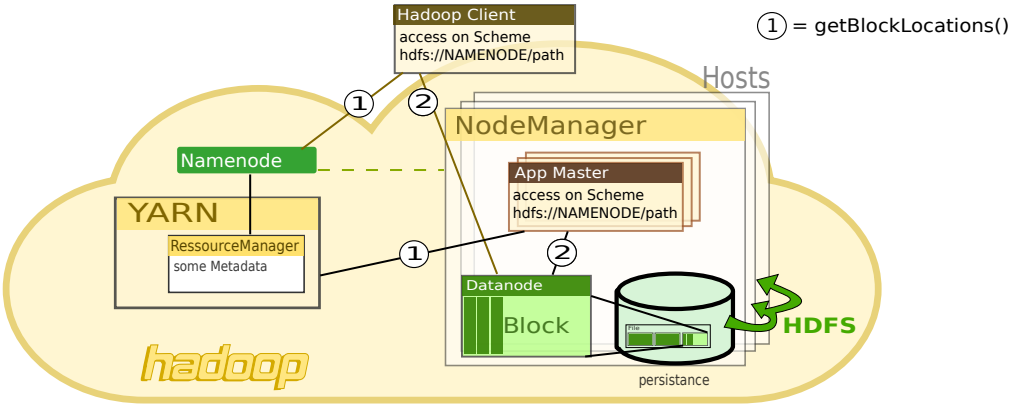
Hadoop

Excursion Hadoop

- starts with HDFS: split big files into big blocks
- block maybe replicated
- Namenode: stores Blocklocations and infrastructure info
- MapReduce: split Job into Tasks on blocks
- becomes more and more a Processhandling "ecosystem"

Run Task where data block is stored.

Excursion Hadoop - Sketch



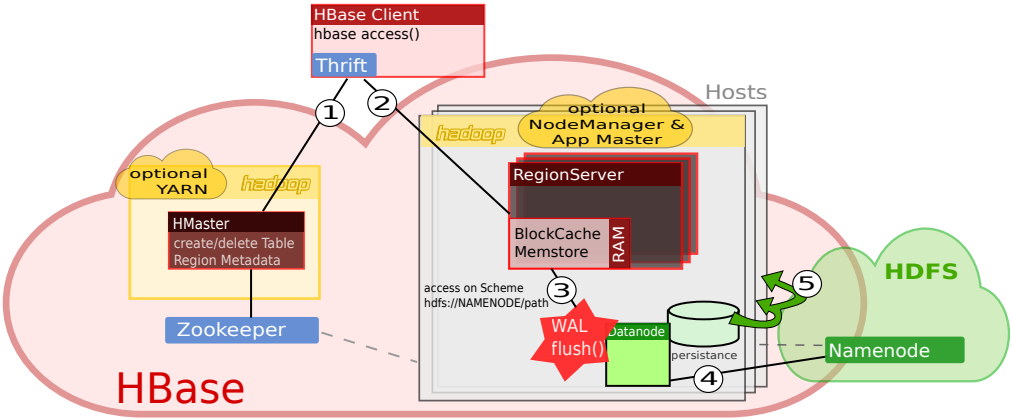
Excursion HBase

Hbase

Excursion HBase

- noSQL mit BASE statt ACID (SQL)
- RAM und WAL je Node
- HDFS zur Persistenz
- Balance und Config wichtig (read, write, RAM, flush, Kompression)
- RegionServer: App in Hadoop

Exkurs HBase - Sketch



HBase und DXRAM

HBase und DXRAM ?

HBase und DXRAM

- HBase nutzt MemStore & BlockCache (RAM)
- WAL: ACK erst, nachdem in HDFS geschrieben
- viel Aufwand für Persistenz und Compaction
- NoSQL: warten auf Festplatte bedeutet Tod für Anwendung

Warum nicht gleich DXRAM als verteilten Speicher nutzen?

Wie machen es Andere?

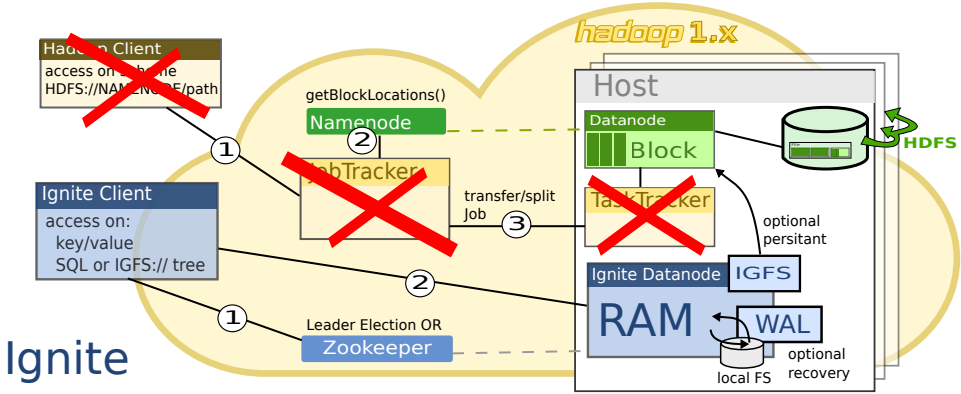
Verteilter Speicher und Hadoop + HBase:
Wie machen es andere Projekte?

Wie machen es Andere?

Ignite:

- verteilter Speicher (key-value)
- hat SQL Erweiterung
- eher Konkurrenz zu HBase
- Hadoop FS Connector
- WAL: HDFS zur Persistenz (SQL) optional

Ignite - Grafik

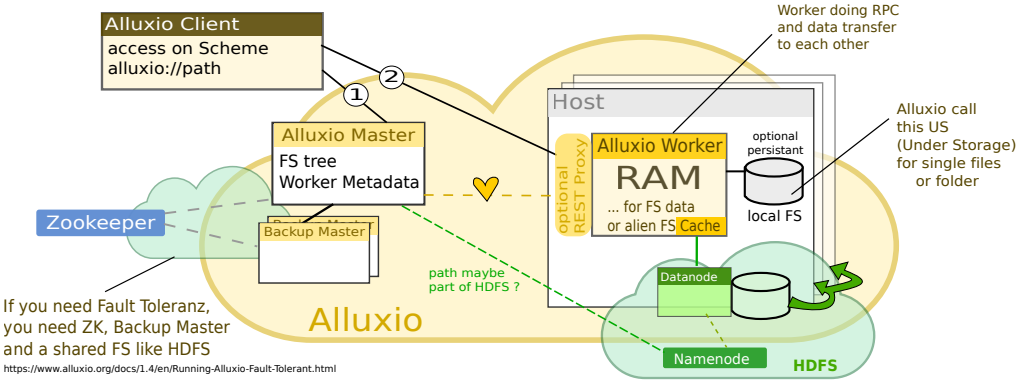


Wie machen es Andere?

Alluxio:

- Hadoop „Branch“
- statt Scheme: mounten anderer FS in Alluxio
- wie ein verteilter FS Cache
- Hadoop FS Connector
- etwas Schräg: HBase nutzen bedeutet quasi 2 Hadoops
- Under Storage: lokale Hard Disk
- Persistenz optional im FS

Alluxio - Grafik

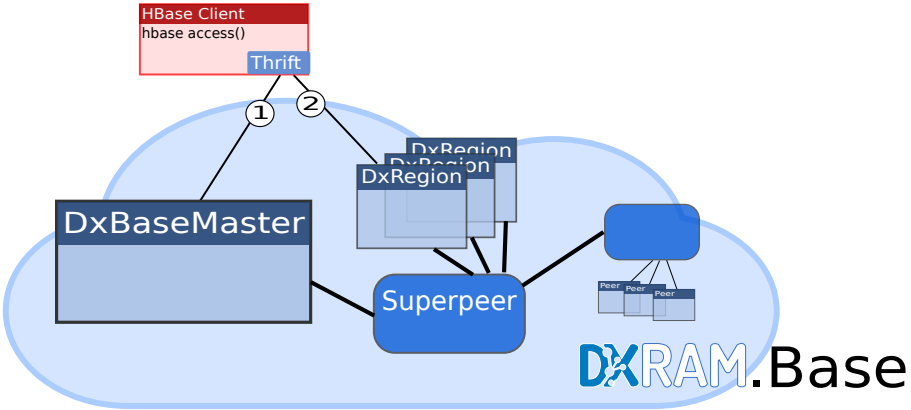


Lösungswege

Lösungswege DXRAM in Hadoop und HBase zu nutzen

Idee 1

HBase Replacement auf der Basis der Thrift Schnittstelle für einen Client.



Idee 1: DXRAM.Base

Pro

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

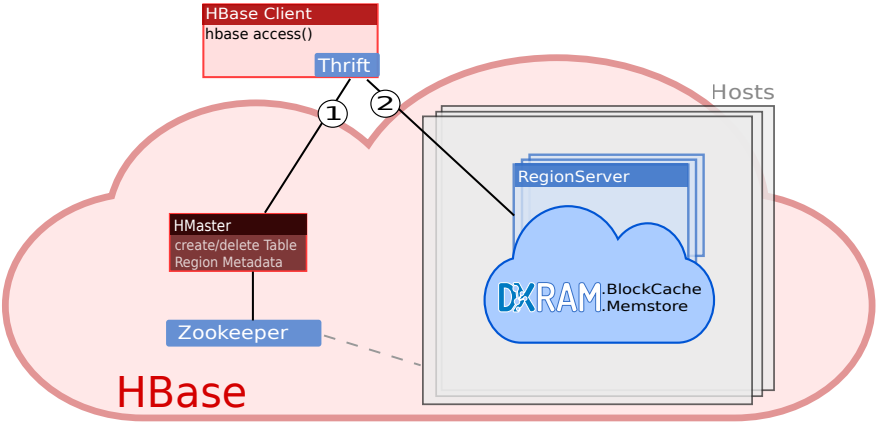
Idee 1: DXRAM.Base

Contra

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

Idee 2

RegionServer RAM zugriff durch DXRAM ersetzen.



Idee 2: DXRAM RegionServer

Pro

- Lösung auf HBase zugeschnitten
- kein Dateisystem, was zu implementieren wäre
- HBase Anwendungen brauchen nicht umprogrammiert werden

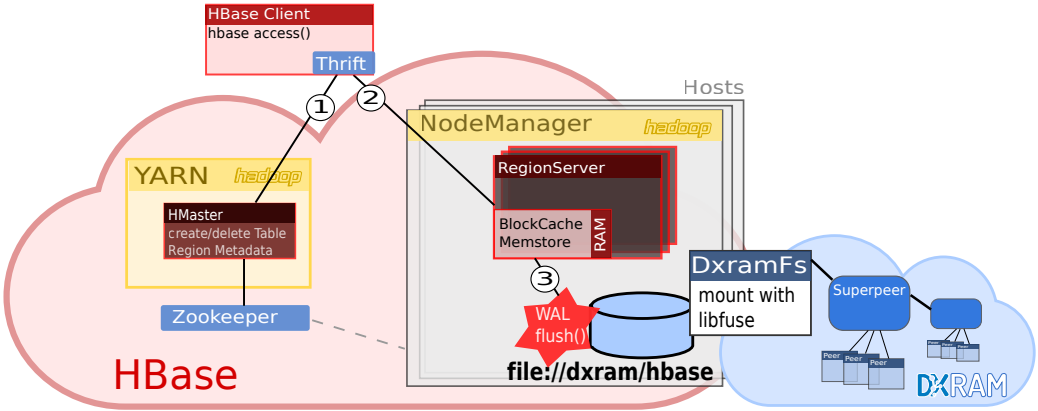
Idee 2: DXRAM RegionServer

Contra

- tiefes Verständnis von HBase Quellcode nötig
- HBase Updates muss man evtl. aufwändig einpflegen
- kein Vorteil für allgemeine Hadoop Projekte

Idee 3

DXRAM zu einem mountfähigen Medium machen mit `libfuse` (vergleichbar mit HDFS auf RAM-Drive)



Idee 3: mount DxramFs

Pro

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

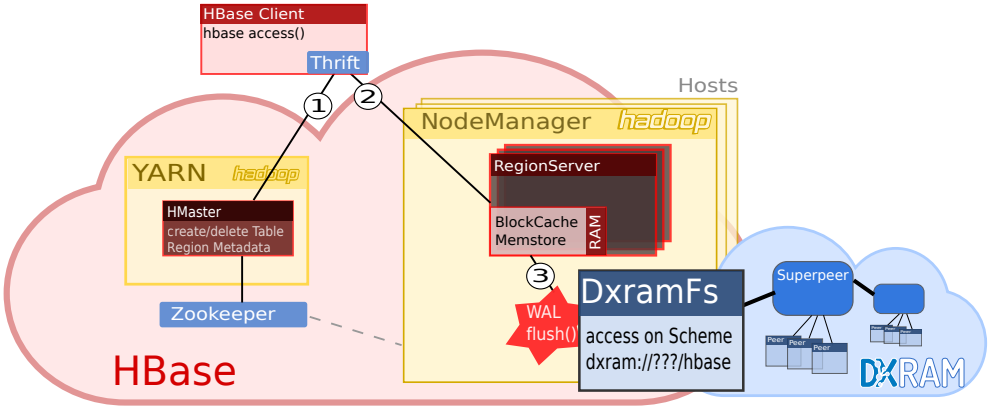
Idee 3: mount DxramFs

Contra

- Prozessverteilung: YARN weiss echten Speicherort nicht mehr
- Performance Probleme bei libfuse
- hier muss ein komplettes verteiltes Dateisystem programmiert werden

Idee 4

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



Idee 4: 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 4: DxramFs Connector

Contra
Mal eben HDFS nach programmieren :o/

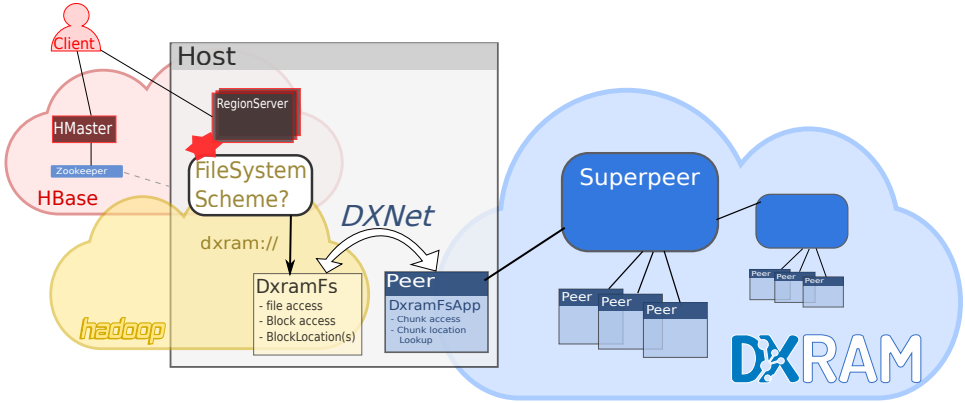
Wahl

Die Wahl fiel auf die Lösung, wo HBase und Hadoop unberührt bleiben, und NUR eine HDFS kompatibler Connector beigefügt wird (Idee 4).

Umsetzung

- DxramFs App: stellt Chunks als Blöcke in einem FS dar
- DXNET: für RPC und Datentransport
- DxramFs Connector in Hadoop: nutzt DXNET
- DXRAM bleibt losgelößt von Hadoop

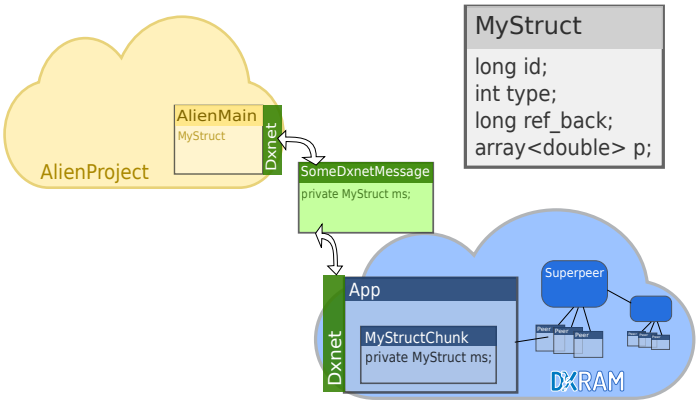
Umsetzung - Grafik



Umsetzung :-(

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

Umsetzung: Fail 1



MyStruct

```
long id;
int type;
long ref_back;
array<double> p;
```

Open questions:

- initial & maximal size
- initial values
- fill with new array length
- when to do NEW ? (read/write payload)
- Java Heap slowdown with to many copies?

=> get/set each single attribute is ugly
=> Bytearray and JSON?

Abbildung: wihout wrapper or preprocessing

Umsetzung: Serialisierung

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

Umsetzung: Wunsch

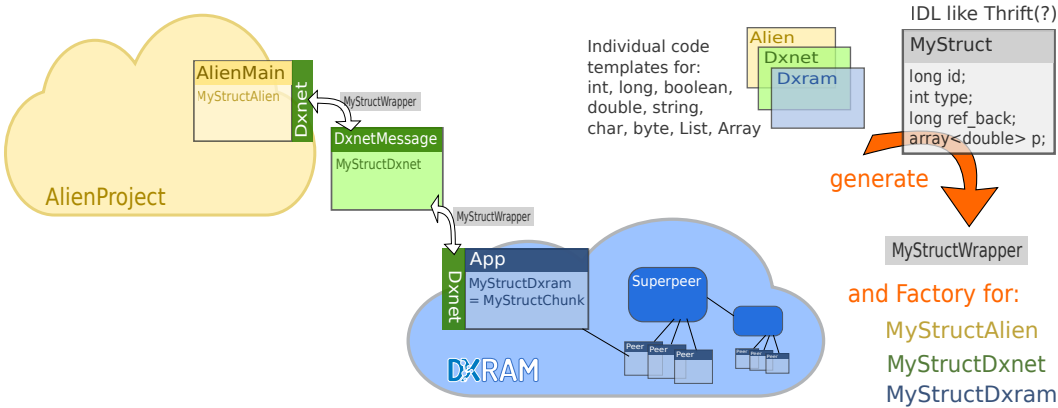


Abbildung: with preprocessing

Umsetzung: Fail 2

Fehler Nr. 2: Aufschieben von Multipeer-Umgebung

Umsetzung: Multipeer

Anstatt Multipeer 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: DXNET Transport

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

Umsetzung: Aktuell

Fertig: FS Aufbau, Ordner Operationen

Umsetzung: Aktuell

Offen

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

Fazit

- YARN zu stark an HDFS und Blockverteilung gekoppelt!
- Ignite & Alluxio: YARN Replacement
- Key-Value Store: HDFS nachbauen schwerer, als Datenbank nachbauen?

Fazit

- YARN zu stark an HDFS und Blockverteilung gekoppelt!
- Ignite & Alluxio: YARN Replacement
- Key-Value Store: HDFS nachbauen schwerer, als Datenbank nachbauen?

Vermutlich Ja. **-> Apache Thrift**

Fazit

Aber: Jeder wirbt auch mit *EINBINDUNG* in Hadoop, nicht mit *ERSATZ*.
To Do: Anwendungsfälle finden, wo auf Hadoop & HBase Replacement sinnvoll ist.

Fragen

Fragen?

Fragen

Danke :o)

Quellen

Im Grunde ist alles auf no-go.github.io/HadoopDxramFS.