



HDFS als schneller und günstiger Storage?

Das Hadoop Distributed File System (HDFS) verwaltet spielend riesige Datenmengen, lässt sich im laufenden Betrieb bequem skalieren und ist komfortabel zu administrieren. Das besondere Konzept des HDFS macht es einerseits robust gegen Ausfälle, andererseits ist es enorm schnell in der Auslieferung an die entsprechende Applikation. Eine Entdeckungsreise in ein spannendes Projekt.

Rasanten Wachstum von Datenmengen und Benutzerzahlen fordern stetig neue Strategien und flexible Lösungsansätze der Informationstechnologien. Seit Langem schon in Einsatz sind Rechnerverbünde, so genannte Cluster bzw. verteilte Systeme. Bestanden früher die Cluster noch aus einer kleinen Anzahl an Rechnern, werden heute bis zu mehreren tausend Server in einem Verbund zusammengeschlossen.

Diese enorme Menge an Hardware bringt ein höheres Ausfallrisiko von Komponenten mit sich. Dies beziehen die Softwarehersteller mehr und mehr in ihre Konzeption mit ein. Ein Beispiel dafür ist das Apache Hadoop Projekt. Anwendung findet diese Software beispielsweise in Applikationen, die sehr große Datenmengen verwalten und ausliefern müssen. Datenvolumina bis in den Petabyte-Bereich können von Hadoop verarbeitet werden.

Apache Hadoop Projekt

Das Hadoop Distributed File System ist Teil des Apache Hadoop Projektes. Die ursprüngliche Aufgabe dieses Projektes bestand in der Verwaltung von riesigen Datenmengen, die bei Suchmaschinen anfallen, während sie das Internet durchsuchen. Ziel dieses Open-Source-Projektes ist eine sichere, gut skalierbare Datenverarbeitung auf verteilten Systemen. Die zentralen Bestandteile sind das Hadoop Common, HDFS und MapReduce.

Für die Verarbeitung dieser zahlreichen wie auch sehr großen Dateien wurde der sogenannte MapReduce-Algorithmus entwickelt, der im Apache Hadoop Projekt seine Implementierung fand. Hadoop Common enthält die grundlegenden Funktionen für sämtliche Hadoop Subprojekte. HDFS hält die Daten vor und MapReduce ermöglicht die Verarbeitung von sehr großen Dateien auf verteilten Systemen.

Aufgrund der enormen Datenmengen wird Hadoop im Cluster betrieben. Eine der aktuell größten Implementierungen von Hadoop umfasst rund 4.000 Server und ist bei Yahoo zu finden. Eine Besonderheit von Hadoop ist der Einsatz von Standardhardware, was für den professionellen Einsatz – je nach Projektgröße – ein ganz erheblicher Kosten- und Zeitvorteil sein kann.



HDFS ist nicht als Online Storage konzipiert, was bereits auf den Projektseiten des Apache Hadoop Projektes sehr klar postuliert wird. Bei einer genaueren Betrachtung des HDFS Konzepts ist dies auch recht schnell zu erkennen. Genannt sei exemplarisch der Single Point of Failure, die NameNode. Sie verwaltet die Indizes sämtlicher gespeicherter Datenblöcke.

Dennoch bietet HDFS so viele Vorteile als Dateisystem, dass es wie Verschwendung anmutet, diese zu verschenken. Der Reiz, die wenigen Hürden mittels durchdachter Sicherheitskonzepte [wie beispielsweise mehrfach redundante Datensicherung] zu überwinden und damit die Einsatzmöglichkeiten von Hadoop zu erweitern, trieb die Techniker der ADACOR Hosting GmbH zu wahren Forscherdrang an.

Herausforderung

Sie kam in Form einer Anfrage von der Red Bull Media House GmbH. Der Kunde plante für sein Projekt „Red Bull Content Pool“, die Infrastruktur für das gesamte Media Asset Management umzustellen. Der „Red Bull Content Pool“ dient dem Kunden als zentrales Repository, in dem der gesamte produzierte Content von Red Bull abgelegt wird und in weiterer Folge schnell und einfach weltweit zur Verfügung gestellt werden kann. Dazu zählen neben Moving Images (von kurzen Clips bis zu Kinoproduktionen) auch sämtliche Still Images und Audiodateien, die z.T. in verschiedenen Formaten und Qualitätsstufen vorgehalten werden bzw. on-demand entsprechend generiert werden können.

Die Liste der Anforderungen enthielt neben dem Hosting der gesamten Projekt-Infrastruktur die Punkte

- Storage im Petabyte-Bereich
- ohne Wartungsfenster skalierbar
- kurze Downtime im Fehlerfall
- Redundanz der Daten
- sehr schnelle Auslieferung der Daten
- einfache Nutzbarkeit für Applikationen
- deutlich günstiger als Standardlösungen, beispielsweise mit EMC-Storage

Verschiedene Ansätze, alle Anforderungen zu erfüllen, wurden bei der ADACOR Hosting GmbH diskutiert und getestet. Darunter waren NFS, GlusterFS, Lustre, Openfiler, CloudStore und schließlich HDFS. In den folgenden Abschnitten sind die jeweiligen Varianten mit ihren Möglichkeiten und Grenzen kurz skizziert; jeweils vor dem Hintergrund der konkreten Anforderungen des Kunden.



NFS

Dieser Lösungsansatz war der erste, der in Erwägung gezogen wurde, da er mit Unix-Bordmitteln zu realisieren ist. NFS ist ein Standard und auf sämtlichen Unix-Derivaten vorhanden. Die Vorteile sind unter anderem finanzieller [Einsatz von Standard-Hardware, keine Lizenzkosten] und administrativer Art [durchsichtiges Konzept, gut zu administrieren]. Außerdem ist NFS stabil und vielfach in verschiedensten Rechnerlandschaften getestet, sodass kaum Gefahr von unbekanntem Bugs ausgeht.

Einige Nachteile des NFS vor dem Kontext der konkreten Aufgabe führten dazu, dass dieser Ansatz verworfen wurde. Dazu zählten die Performance, die nicht vorhandene Replikation der Daten und die Verwaltung des Storage, welche die Applikation vollständig selbst hätte übernehmen müssen.

GlusterFS

Als Nächstes wurde GlusterFS einer genaueren Betrachtung unterzogen. Das dateibasierte GlusterFS wurde für verteilte Systeme [Cluster] entwickelt und ist bei der ADACOR Hosting GmbH bereits in anderem Kontext im Einsatz. Mit steigender Anzahl an GlusterFS-Servern steigt der maximal mögliche Datendurchsatz. GlusterFS verhält sich robust gegen Fehler und kann beliebig skaliert werden.

Aufgrund des sehr hohen administrativen Aufwands [aufwändig zu konfigurierende Replikationsmechanismen] und der Baumstruktur, die bei jeder Skalierung eine Downtime erfordert, wurde auch dieser Ansatz verworfen.

Lustre

Das blockbasierte Dateisystem Lustre überzeugte durch hohe Performance, gute Skalierbarkeit und komfortable Administration. Es verwaltet bis zu mehreren Tausend Clusternodes sowie Daten im Petabyte-Bereich. Nutzdaten und Metadaten werden getrennt auf unterschiedlichen, redundanten Servern vorgehalten.

Zum Zeitpunkt der Tests fehlten diesem Dateisystem noch die Replikationsmechanismen. So entfiel Lustre als Lösung für die Kundenanforderung.

Openfiler

Problemlos verwaltet Openfiler interne Platten als Software-RAID. Über iSCSI können externe Platten ebenfalls eingebunden werden, was in den Tests auch probiert wurde. Das Einbinden und Verwalten ging mühelos. Ein weiterer Vorteil ist die Verwaltung mit einem kleinen Standard-Hardware-Server.



Einzigste Hürde für die konkrete Anforderung ist die große Menge der Daten. Ein Rebuild von 3 Terabyte brauchte mehrere Tage; für Datenmengen im Petabyte-Bereich würde Openfiler zu lang brauchen; daher wurde auch diese Möglichkeit ausgeschlossen.

CloudStore

CloudStore, ehemals KFS bzw. Kosmosfs, konnte nicht ausgiebig getestet werden, da sich zum Zeitpunkt der Tests diese Variante noch im Entwicklungsstadium befand. Schon beim Kompilieren gab es so erhebliche Schwierigkeiten, dass diese Möglichkeit schon im Ansatz ausgeschlossen wurde.

HDFS

Das blockbasierte Hadoop Distributed File System (HDFS) überzeugte die ADACOR Hosting GmbH durch sehr schnelle Auslieferung der Daten, komfortable Verwaltung von vielen wie auch sehr großen Dateien, erhebliche Fehlertoleranz, zentrale Administration, Einsatz auf Standardware und Redundanz der Daten. Die HDFS-Umgebung kann ohne Wartungsfenster skaliert werden.

Es klingt wie die perfekte Lösung. Doch auch HDFS bringt eine Hürde mit. Und diese ließ die Köpfe der ADACOR-Techniker rauchen.

Im HDFS gibt es zwei Typen von Servern (Clusternodes): NameNodes und DataNodes. Die Datenblöcke werden auf den DataNodes vorgehalten; sämtliche Meta-Informationen sind auf den NameNodes gespeichert. Sämtliche Daten werden mehrfach redundant gesichert.

Die NameNode stellt einen Single Point of Failure dar. NameNodes sind grundsätzlich redundant ausgelegt, jedoch als Aktiv/Passiv-Cluster organisiert. Fällt die aktive NameNode wegen eines Hardwarefehlers aus, muss die passive manuell gestartet und einige Konfigurationsdateien angepasst werden, was einige Minuten dauert. Ist beispielsweise eine Kernelpanic, verursacht durch einen Softwarefehler, die Ursache, kann die aktive Node neu gestartet werden und die Konfigurationsarbeiten entfallen. Während dieser Zeit ist kein Zugriff auf das HDFS möglich. Aktuell laufende Schreibprozesse auf das Filesystem werden mit einer Fehlermeldung abgebrochen.

Die verschiedenen Lösungen wurden mit dem Kunden abgestimmt und die Wahl fiel auf HDFS. Die Problematik bezüglich der NameNode ist auf Anbieter- wie auf Kundenseite bekannt und bewusst. Um Fehler frühzeitig zu erkennen, werden alle Server rund um die Uhr mit der Monitoringsoftware Nagios überwacht. Dort ist verankert, dass jederzeit bei einem Alarm sofort ein Techniker informiert wird, der umgehend eingreifen kann.



Fazit

Alle getesteten Dateisysteme weisen viele Vorteile, aber auch einige Nachteile auf. Hier bleibt dem professionellen Anwender die wichtige Aufgabe überlassen, die für ihn günstigste Variante zu wählen und gegebenenfalls zusätzliche Sicherheitsmaßnahmen zu ergreifen.

Für das aktuell betrachtete Projekt stellte sich HDFS als günstigste Lösung heraus, die nicht nur in der Theorie, sondern auch in der Praxis sehr gut funktioniert. Beide Seiten – Anbieter wie Kunde – sind mit dem Ergebnis sehr zufrieden.

Jedes Projekt hat seine ganz spezifischen Anforderungen und findet individuelle Lösungen. Zu jedem Topf ein Deckel.

Anhang

Bestandteile des Apache Hadoop Projektes

- Hadoop Common,
- Chukwa,
- HBase,
- HDFS,
- Hive,
- MapReduce,
- Pig und
- Zookeeper.

Den Kern des Konglomerats bildet das Hadoop Common [ehemals Hadoop Core] in den die übrigen Komponenten eingebettet sind. Hier sind die grundlegenden Funktionen aller Komponenten enthalten.

Chukwa ist die Grundlage für das Arbeiten auf verteilten Systemen, die Skalierbarkeit und die Robustheit von Hadoop. Mittels Chukwa lässt sich Hadoop komfortabel monitoren und es lassen sich Analysen zur Optimierung der Hadoop-Installation durchführen.

HBase ist die Hadoop-Datenbank und spezialisiert auf Lese-Schreibzugriffe von sehr großen Dateien über viele Systeme [Nodes]. HBase orientiert sich an der Bigtable von Google und hat sich als Ziel gesetzt, eine gigantische Tabelle zu verwalten, die Billionen von Zeilen und Millionen von Spalten hat. Das Besondere an HBase – und am gesamten Hadoop-Konzept – ist die Verwendung von Standard-Hardware.



HDFS verwaltet die in Blöcke aufgespaltenen Dateien sowie ihre Repliken. Es regelt außerdem die Verteilung der jeweiligen Datenblöcke auf den verschiedenen Nodes.

Hive enthält neben verschiedenen Analysewerkzeugen auch die Sprache Hive QL, die auf SQL basiert und Datenbankabfragen auf SQL-vertraute Weise erlaubt. Ganz dem Open-Source-Gedanken entsprechend können Entwickler mit Hive QL eigene, spezifischere Analysemethoden implementieren.

Das Framework MapReduce ermöglicht einen sehr schnellen und parallelen Zugriff auf eine hohe Anzahl an Dateien, die auf zahlreichen Clusternodes verteilt sind.

Pig ist die Grundlage für die parallele Verarbeitung. Wie Hive ist auch Pig so konzipiert, dass Entwickler eigene Funktionen programmieren und implementieren können.

Mit dem Zookeeper werden Konfigurationen auf verteilten Systemen zentral verwaltet. Über eine Weboberfläche lässt sich der Zookeeper bequem administrieren.