8

8 Petabyte Computing

8.1 Einleitung

Ein Petabyte an Daten entspricht dem Textinhalt einer Milliarde Bücher oder dem monatlichen Output eines einzigen wissenschaftlichen Experimentes. Petabyte Computing bedeutet dem Umgang mit Datenmengen im Bereich ein oberer mehrere Petabytes. Ein Petabyte sind 1.000.000.000.000.000 Bytes.

Der Begriff Petabyte Computing taucht zum ersten Mal in Zusammenhang mit den am CERN geplanten Teilchenbeschleuniger Experimenten 1999 auf [Düllmann 1999]. In der März Ausgabe von Nature, einer renommierten Amerikanischen Zeitschrift, beschreiben Alexander Szalay und Jim Gray die modernen Analysemethoden der Wissenschaft, die auf sehr grosse Datenmengen zugreifen, um Ihre Experimente durchzuführen und auszuwerten [Szalay, Gray 2006]. Heute arbeiten die meisten Wissenschaftler und Wissenschafterinnen mit Notebooks und Desktops, die Datenanalysen mit Tools wie Matlab, Mathematica oder Excel erlauben. Keines dieser Programme kann jedoch Millionen von Datensätzen verarbeiten. Heute ist die Analyse von Daten in der Grössenordnung eines Terabytes eine Herausforderung, die Analyse von Daten in Petabyte Bereich steht in der Teilchenphysik, der Astronomie, der Meteorologie und der Biologie vor der Tür und kann mit der gängigen Analyse-Software nicht durchgeführt werden.

Petabyte Computing wird heute vor allem von der Wissenschaft bereits benötigt. Der Zugriff auf sehr Mengen von Rohdaten ist notwendig, um explorative Analysen durchzuführen und so die Verifikation oder Falsifikation von wissenschaftlichen Thesen zu erlauben.

Im kommerziellen Bereich wird die explorative Analyse von Daten im Rahmen von Business Intelligence und Data Warehouse Technologie eingesetzt. Dabei wird aufgrund derjenigen Daten, die das laufende Geschäft dokumentieren, versucht, das Unternehmen so zu steuern, dass eine Ressourcen-Maximierung erreicht werden kann. Die Datenmengen, die dabei anfallen, können bereits heute die Petabyte Grenze erreichen.

8.2 Petabyte Anwendungen

8.2.1 Die vier CERN Large Hadron Collider Experimente

Das CERN (Organisation Européenne pour la Recherche Nucléaire) hat im November des Jahres 2007 den Large Hadron Collider (LHC), den leistungsfähigen Teilchenbeschleuniger der Welt, in Betrieb genommen.

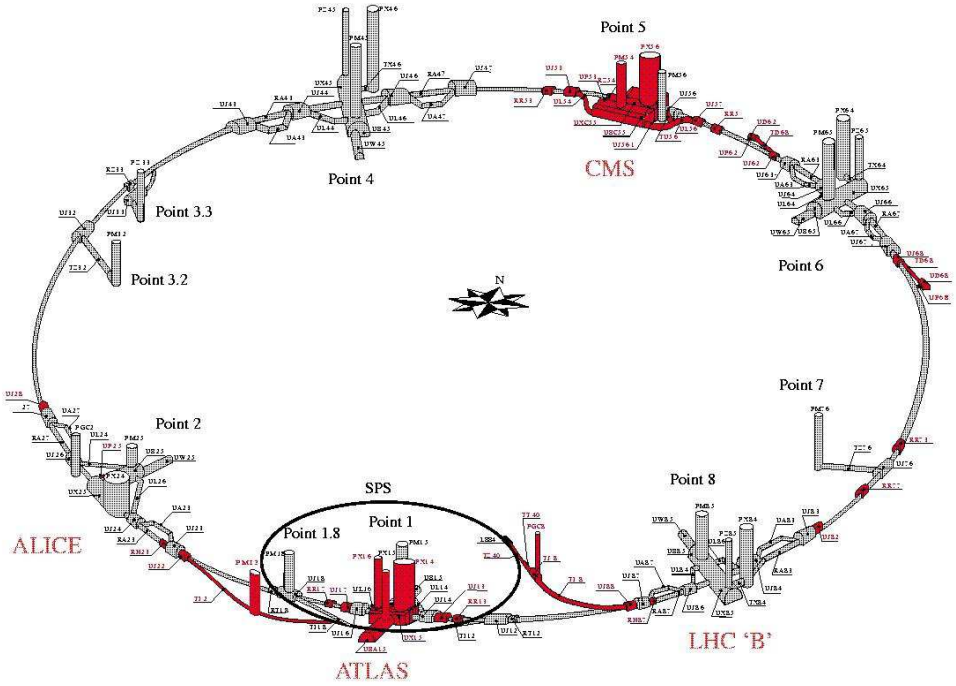

Abbildung 8.1 Der Teilchenbeschleuniger und seine vier Detektoren

Der Teilchenbeschleuniger (**Abbildung 8.1**) ist in einer Tiefe von 50 – 175 Metern gebaut worden und hat einen Umfang von 27 km. Die Idee zum Bau einer solchen Anlage stammt aus dem Jahr 1980. Der erste Meilenstein des Projektes wurde 1994 erreicht, Baubeginn war 1998, die ersten Experimente sollten ab dem Jahr 2008 durchgeführt worden sein.

"Der nach einer langjährigen Planungs- und Vorbereitungsphase derzeit im Bau befindliche Large Hadron Collider (LHC) am CERN in Genf wird allgemein als das Flaggschiff der Hochenergiephysik in den nächsten zehn bis zwanzig Jahren angesehen. Hier werden in Proton-Proton-Kollisionen die Zusammenstösse zwischen ihren elementaren Bausteinen - den Quarks und Gluonen - bei bisher unerreichten Energien im TeV-Bereich (1 Teraelektronenvolt = 10^{12} Elektronenvolt) untersucht." (www.weltderphysik.de).

Tabelle 8.1 Technische Daten des Teilchenbeschleunigers

Umfang	**26,659 km**
Magnete	supraleitend bei 1,9 K (Dipole und Quadrupole) bzw. 4,5 K
Magnetfeld	max. 9 T
Kollidierende Teil	Protonen und schwere Ionen
Schwerpunktsenergie	14 TeV für Protonen, 1150 TeV für Schwerionen
Kollisionsrate	max. 40 Millionen pro Sekunde

Der 2.2 Milliarden Franken teure Teilchenbeschleuniger soll Experimente erlauben, die Aufschluss über den Aufbau des Universums und das Verhalten von Elementarteilchen wie beispielsweise Bosonen erlauben. Im speziellen wird das so genannte Higgs-Boson experimental untersucht. Seine Existenz ist mit dem zentralen Problem der Teilchenphysik verbunden, mit der Frage wodurch Teilchen ihre Masse erhalten. Andere Grundlegende Fragestellungen, wie beispielsweise die mögliche Existenz höherer Raumdimensionen sind damit verbunden.

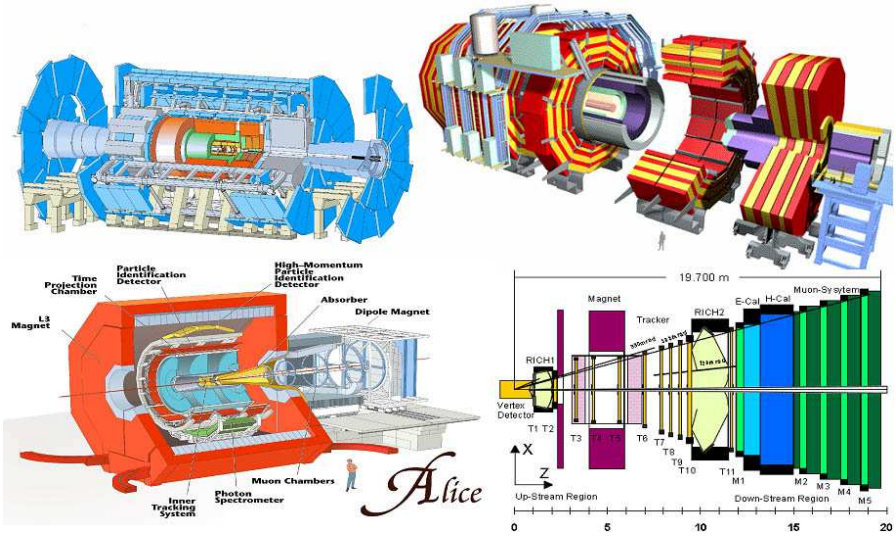

Abbildung 8.2 Die Detektoren für die Experimente ATLAS, CMS, Alice und LHCb

Heute sind vier Experimente geplant, die von einer Forschergemeinde bestehend aus über tausenden von Physikern aus über 30 Ländern konzipiert worden sind. Für jedes dieser Experimente sind spezielle so genannte Detektoren gebaut worden (**Abbildung 8.2**).

Tabelle 8.2 Die vier CERN Experimente

Experiment	**Detektor**	**Zweck**
ATLAS	Vielzweckdetektor für Proton-Proton-Kollisionen	Nachweis des Higgs-Bosons

CMS	Vielzweckdetektor für Proton-Proton-Kollisionen	Nachweis des Higgs-Bosons
Alice	Vielzweckdetektor, optimiert für Kollisionen von Schwerionen	Simulation des Urknalls
LHCb	Proton-Proton-Kollisionen, spezialisiert auf die Messung der Eigenschaften von Hadronen mit Bottom-Quarks	Symmetrieverletzung der so genannten schwachen Wechselwirkung

Der experimentale Nachweis der Formen $E = mc^2$ Ist nach heutiger Erkenntnis nur durch den Nachweis des Higgs-Bosons möglich. Der schottische Physiker Peter W. Higgs stellte 1964 die Theorie auf, dass das Weltall nicht leer ist, sonder überall ein Teilchenfeld existiert. Elementarteilchen, die zunächst ohne Masse sind, reagieren mit diesem Feld. Die Wechselwirkungsenergie der Teilchen mit dem Feld wird als Masse gedeutet. Hinter diesen Experimenten stehen zwei zentrale Fragen:

- Welcher physikalische Prozess wandelt Energie in Masse um?
- Wie wandelt man Masse in Energie um?

Und damit die Frage, ob eine Umwandlung mit einem Wirkungsgrad von 100 % erreicht werden kann [Herten 2005].

8.2.2 Das Datenaufkommen der CERN Experimente

Die Experimente des CERN werden Rohdaten in der Grössenordnung von 15 Petabytes pro Jahr liefern [Duellmann 1999].

Tabelle 8.3 Datenaufkommen der CERN Experimente

Experiment	Data Rate	Data Volume
ATLAS	100 Mb / sec	1 Petabyte / year
CMS	100 Mb / sec	1 Petabyte / year
Alice	1.5 Gb / sec	1 Petabyte / month
LHCb		400 TB / year

Zur Verarbeitung dieser Datenmengen hat das CERN im Jahr 2001 das LHC Computing Project lanciert [Robertson 2001], welches auf dem MONARC Multi-Tier Modell basiert. Für die Speicherung und die Verwaltung der Daten wurde zunächst ein Objekt Orientiertes Datenbank Konzept vorgesehen, welches jedoch später zugunsten der Grid Computing Architecture aufgegeben worden ist.

8.2.3 The World-Wide Telescope

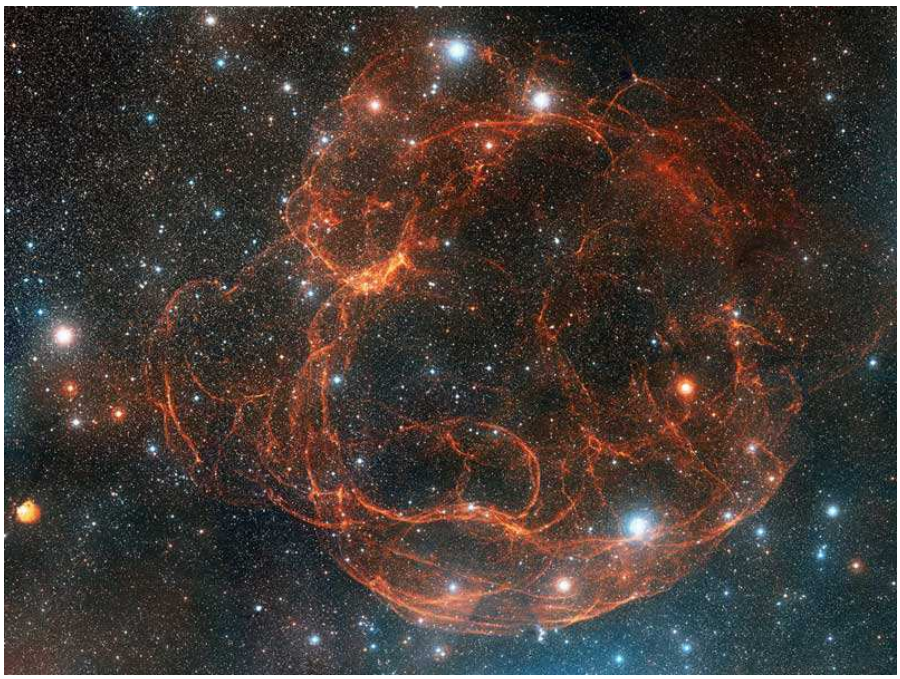

Abbildung 8.3 Supernova Simeis 147

Die Supernova Simeis 147 (**Abbildung 8.3**) wurde am 29.11.2005 vom Palomar Observatorium aufgenommen. Sie hat 100'000 Lichtjahre entfernt stattgefunden und umfasst eine Breite von 150'000 Lichtjahren.

The World Wide Telescope ist eine Initiative von Astronomen und Informatikern, die eine Kombination aller Astronomie Archive zu einem Virtuellen weltweiten Observatorium zu Ziel hat [Gray, Szalay 2006]. Bereits heute produzieren moderne Teleskope Terabytes von Daten jedes Jahr. In naher Zukunft werden Daten, die mehrere Terabytes pro Nacht umfassen, erzeugt. Astronomiedaten werden durch eine Reihe von Instrumenten rund um die Erde, wie auch durch Satelliten oder Raumstationen gesammelt. Jedes dieser Instrumente misst die Lichtintensität (Flux) in bestimmten Spektralbereichen. Basierend auf diesen Informationen extrahieren Astronomen hunderte von Objektattributen wie beispielsweise die Grösse, die Ausbreitung, die wahrscheinliche Struktur und die Morphologie (geordnete Art der Betrachtung und der Bewertung der Dinge) eines Objektes.

Der Palomar Overservatory Sky Survey hat bereits 1958 ein umfassendes Bildarchiv des Nachthimmels publiziert. Dieses Archiv umfasst 1872 Photographien, die damals für 25'000 $ pro Stück zu erwerben waren. Diese Daten sind heute Online verfügbar.

http://www.worldwidetelescope.org/ (Interact – HTML 5 Web Client)

Die Analyse eines astronomischen Objektes bedeutet in vielen Fällen die Kombination verschiedenster Daten. So wird die zeitliche Entwicklung des Objektes durch die Kombination derselben Messung über grosse Zeiträume hinweg betrachtet oder es werden verschiedene Spektralanalysen kombiniert. Diese Kombination der Daten bedeutet in vielen Fällen, dass der Forscher / die Forscherin verschiedene Datensätze über FTP von verschiedenen Institutionen anfordert und als FITS (Flexible Image Transport System) Datei herunterlädt. Diese Daten werden dann konvertiert, um die Analyse durchzuführen. Der Datentransfer eines Gigabytes an Daten über FTP dauert eine Minute, ein Terabyte kann einen Tag oder mehr dauern, für ein Petabyte Download braucht es Jahre.

Das World-Wide Telescope umfasst sämtliche heute verfügbaren astronomischen Archive, die jedes einzelne für sich in grossen (ca. 100 Terabyte) relationalen Datenbank gespeichert ist. Die US-Astronomen Alexander Szalay von der Johns Hopkins University in Baltimore und Jim Gray vom Microsoft Bay Area Research Center in San Francisco sehen in ihrem Konzept drei zusätzliche Komponenten vor, um das virtuelle Teleskop technisch zu realisieren:

- *Plumbing:* Die Speicherung, Organisation, der Zugriff und die Suche der Daten in grossen Archiven und in verteilten Internet Datenbanken. Dies erfordert ein umfassendes Meta-Data Management, um die verschiedenen Archive zu integrieren.
- *Data Mining Algorithms and Tools:* Data Mining Algorithmen und Instrumente helfen Anomalien und Trends zu entdecken. Dies erfordert eine Kombination statistischer Methoden mit adaptiven Systemen.
- *Data Visualization:* Die gute Darstellung der Daten ist erforderlich, um Anfragen zu erzeugen und Resultate visuell darzustellen.

Bereits im Jahre 2002 wurde ein Prototyp des World-Wide Telescope, der Virtual Sky Server, erstellt [Gray, Szalay 2002].

8.2.4 Das Datenaufkommen des World-Wide Telescope

Die Initianten des World-Wide Telescope haben ein Szenario durchgerechnet, um die technische Realisierung des Virtuellen Observatoriums zu illustrieren [Vandenberg et al. 2002].

Die Astronomie verfügt heute über Instrumente, die Hochauflösende Bilder erzeugen. So produziert eine 5 Gigapixel Kamera ein 10 Gigabyte Bild. Die übliche astronomische Beobachtungsserie deckt den gesamten Himmel in 4 Nächten ab, wobei pro Nacht ca. 5 Terabyte Daten erzeugt werden. Ein gut positioniertes Observatorium erlaubt eine ungestörte Beobachtung während 200 Nächten pro Jahr, was ein Datenaufkommen von 1 Petabyte bedeuten würde.

Bereits heute ist die Speicherung von 5 Terabyte in 8 Stunden, also von 170 Megabyte pro Sekunde auf Tapes eine Herausforderung, die Speicherung auf Disks ist einfach. Der Zugriff auf die Daten ist jedoch eine andere Sache. Als Faustregel kann gemäss Vandenberg

gelten, dass der lesende Zugriff in Sekunden auf eine Disk im Maximum mit einer Geschwindigkeit der Hälfte der Quadratwurzel der Diskgrösse erfolgen kann.

Die Analyse von astronomischen Daten erfordert zusätzlich zu den reinen Rohdaten (Bildinformationen in verschiedenen Spektren) Objektkataloge, Kalibrierungsparameter, Observationsaufzeichnungen und andere Informationen. Es wird heute von ungefähr 2 Milliarden Himmelsobjekten ausgegangen, die astrologisch erfasst werden. Diese Zusatzdaten umfassen circa 10% der Rohdaten, also 100 Terabyte pro Jahr und Observatorium.

8.2.5 Large Scale Commercial Data Warehouse

Ein DWH ist das Instrument zur Konsolidierung verschiedenster Daten eines Unternehmens aus unterschiedlichen Datenquellen. Zweck ist die möglichst vollständige Darstellung des Unternehmenszustandes zu einem bestimmten Zeitpunkt oder über einen gewissen Zeitraum hinweg. So ist beispielsweise die Firma AT&T daran interessiert, sämtliche Daten über die Nutzung ihrer verschiedenen Netzwerke zu sammeln, um Aussagen über die Auslastung oder die Gefährdung machen zu können [Singhal 2004]. Diese Datensammlungen erreichen bereits heute mehr als 100 Terabyte. Die konsolidierten Daten einer grossen Einzelwarenkette wie beispielsweise Walmart kann bereits heute 1 Petabyte erreichen. Die analytischen Instrumente, die Erforderlich sind, um Trends zu erkennen, sind auf eine gewisse Historie dieser Daten angewiesen. Dies bedeutet, dass sich die Daten Jahr für Jahr kumulieren.

Die meisten Unternehmen setzten heute Themen bezogene DWH ein, um die Datenmenge in Grenzen zu halten. So wird eine Firma wie beispielsweise die Swisscom über ein DWH für den Bereich Festnetz (Fixnet), eines den Bereich Mobile und möglicherweise eines für den Bereich Netzbetrieb verfügen. Der Nachteil dieser Lösung ist, dass ein Unternehmen immer nur partiell betrachtet werden kann, das heisst die Messung und die Steuerung von Geschäftsmodellen, die über verschiedene Geschäftsbereiche hinweg funktionieren sollen, ist kaum möglich.

Auf dem ersten Platz in Bezug auf Datenvolumen der Liste der grössten kommerziellen Datenbanken sind die DWH von eBay mit 9.2 Petabytes Rohdaten und von Apple mit 8 Petabytes Rohdaten – beide durch das Produkt von Teradata betrieben. Im Guinessbook wird ein Labor DWH der SAP mit 12.1 Petabytes Rohdaten als grösste DB überhaupt aufgeführt.

8.2.6 Personal Memex

Die Vision von Bush alle Informationen eines Forschers im Memex zu speichern, ist heute im privaten Bereich eine Realität. Die privaten Speichermedien der Menschen werden gefüllt mit persönlichen Aufnahmen. Ob das nur Familienbilder, die Aufnahme der ersten Gehversuche des Kindes oder andere Dinge sind, mehr und mehr wird sämtliche Information, also alles was ein Individuum hört und sieht gespeichert. Wenn jemand alles, was er oder sie liest, speichert, so fallen circa 25 GB Daten im Jahr an. Würde alles, was wir im

laufe eines Jahres hören, abgespeichert, so fallen 100 GB an. Dies zu speichern ist heute möglich. Würden wir jedoch alles, was wir sehen als Video in TV Qualität speichern, so fallen 10 Terabyte pro Jahr an, die bedeutet ca. 1 Petabyte im Verlaufe unseres Lebens [Gray 2003].

8.3 Die Basis: Database Engines

Datenbankmaschinen (Database Engines) sind bei weitem das wichtigste Instrument zur Speicherung und Verwaltung von Informationen. Bereits die Definition von Dr. Andrea Rodríguez beschreibt diese Tatsache: *Eine Datenbank ist das Model eines Systems der realen Welt. Der Inhalt (manchmal die „Extention" genannt) repräsentiert den Status des zu modellierenden Gegenstandes. Änderungen in der Datenbank repräsentieren Events in der Umgebung, die den Status des modellierten Gegenstandes ändern. Es ist angemessen, eine Datenbank so zu strukturieren, dass sie dass, was modelliert wird, spiegelt.* [Rodríguez 2004].

8.3.1 Architektur

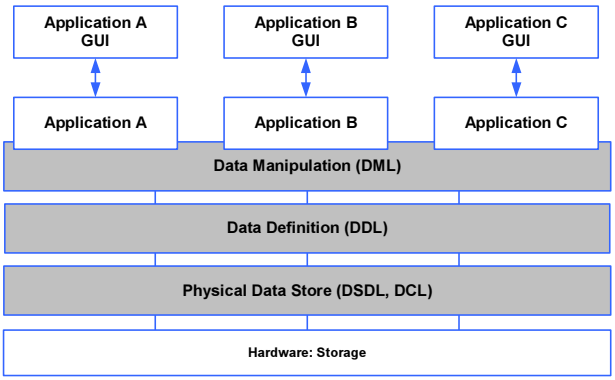

Abbildung 8.4 Layering eines Database Management System

Die logische Architektur jedes DBMS (Database Management System) umfasst drei Schichten, den Data Definition Layer, den Data Manipulation Layer und den Physical Data Store Layer, wie in **Abbildung 8.4** dargestellt. Diese wiederum setzten einem Storage Medium auf. Die meisten DBMS verwenden Disks als Storage. Wenige Ausnahmen, wie beispielsweise die TimesTen In-Memory Database, werden für Real-Time OLTP Processing Aufgaben eingesetzt. Um eine optimale Speicherung der Daten zu erreichen, organisiert ein DBMS die Speicherung selbst. Die Strukturierung der Speicherung erfolgt im Physical Data Store Layer, der untersten der drei Schichten. Die Definition der einzelnen Datenty-

pen erfolgt in einer weiteren Schicht, dem Data Definition Layer. Die oberste Schicht (Data Manipulation Layer) ist für die Manipulation der Daten zuständig.

- *Data Manipulation:* Auf dieser Ebene wird mittels einer Data Manipulation Language (DML) das Abfragen, Einfügen, Ändern und Löschen von Daten ausgeführt. Eine DML ist eine Datenbanksprache.
- *Data Definition:* Die Data Definition Language (DDL) wird zur Definition von Tabellen, Objekten und Indizes eingesetzt. Eine DDL ist eine Datenbankbeschreibungssprache.
- *Physical Data Store:* Die physische Speicherung von Daten erfolgt durch eine Data Storage Description Language (DSDL), der Zugriff auf die gespeicherten Daten erfolgt über die Data Control Language (DCL).

Die logischen Architektur wird durch jedes DBMS in eine Reihe von Building Blocks abgebildet, die zusammen die Grundfunktion des Systems ausmachen. Die Building Blocks sind die Database selbst, die Shared Database Services, der Database Cache und der Database Server Pool.

Die Datenbank stellt ein logisches Datenschema physisch dar und enthält die eigentlichen Daten. Die grundlegenden Dienste sind im Building Block Shared DB Services zusammengefasst. Der Database Cache ist für die Optimierung der Zugriffe zuständig. Der Database Server Pool fasst alle Prozesse, die extern aufgerufen werden können.

8.3.2 Building Blocks einer Datenbank

Die zentralen logischen Bestandteile einer Datenbank können als Building Blocks bezeichnet werden, die in jeder Datenbank-Implementierung zu finden sind (**Abbildung 8.5**). Eine typische Datenbank besteht aus einer Reihe von DB Applications, dem DB Server Pool, einem DB Cache, den Shared DB Services, der Database und dem logical Schema.

- Applikationen, die eine Datenbank als zentralen Datenspeicher verwenden, bauen eine Verbindung zum Datenbank Server Pool auf. Die Verbindung starten einen DB Server Process. Dieser Prozess parst die Anfrage und führt sie aus.
- Der DB Cache enthält bereits geparste SQL Statements.
- Die angeforderten Daten werden vom Database File, welches die Tabellen und die Indizes enthält, gelesen und in den Database Cache geschrieben (in den Table Buffer).
- Der DB Writer ist ein Beispiel eines Shared DB Service. Dies bedeutet, dass alle DB Server Prozesse dieselben Dienste verwenden. Ein Database Management System (DBMS) umfasst normalerweise viele Shared Services.
- DBMS verwalten ein Transaction Log, welches sämtliche Änderungen der Daten protokolliert. Solange eine Transaktion nicht vollständig durchgeführt worden ist, werden die Daten in den Log Buffer geschrieben. Der Transaction Log Writer schreibt die Daten nach einem Commit in das Transaction Logfile.

- Ein anderer typischer DB Service ist der Backup Writer, der ein Backup File (Archive File) schreibt.

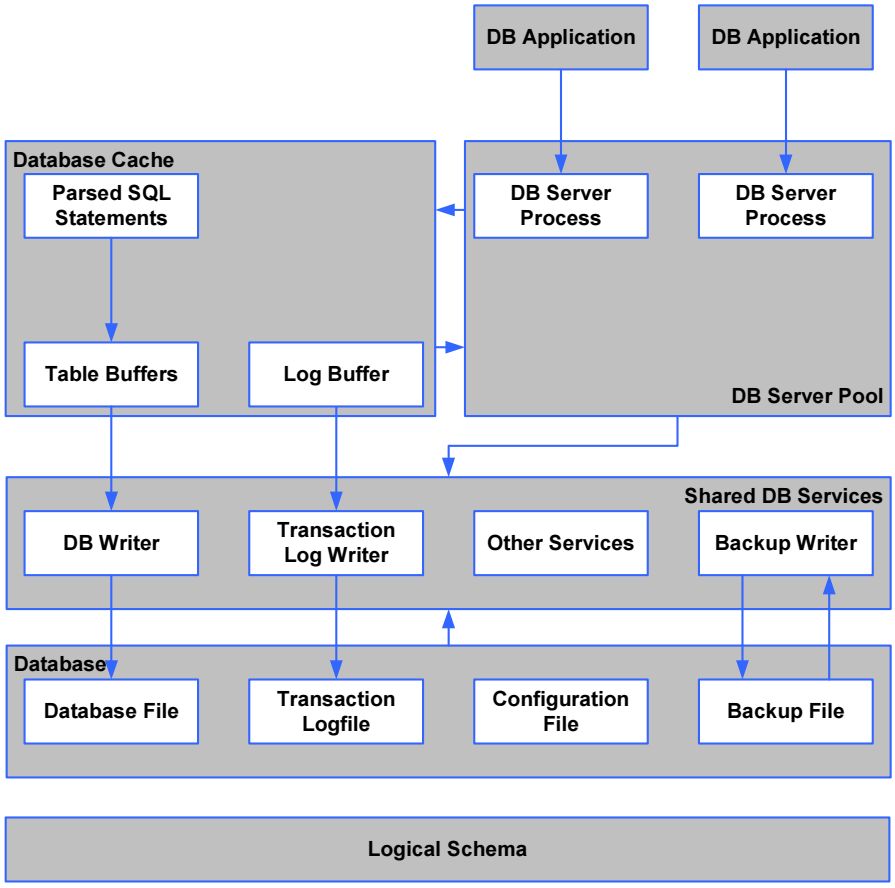

Abbildung 8.5 Building Blocks eines DBMS

8.3.3 Die Regeln von Codd

Dr E.F. Codd, ein IBM Forscher, entwickelte das Relationale Datenmodell im Jahr 1970. Er formulierte 1985 12 Regeln, die als die Basis für die Realisierung von RDBMS Systemen gelten.

Rule 1: The Information Rule
All data should be presented to the user in table form.

Alle Informationen in relationalen Datenbanken sind als Werte in Tabellen darzustellen.

Rule 2: Guaranteed Access Rule

All data should be accessible without ambiguity. This can be accomplished through a combination of the table name, primary key, and column name.

Jeder Wert einer relationalen Datenbank muss durch eine Kombination von Tabellenname, Primärschlüssel und Spaltenname (Attributname) auffindbar sein. Es darf also an jedem Schnittpunkt einer Spalte mit einer Zeile in einer Tabelle nur ein Wert stehen.

Rule 3: Systematic Treatment of Null Values

A field should be allowed to remain empty. This involves the support of a null value, which is distinct from an empty string or a number with a value of zero. Of course, this can't apply to primary keys. In addition, most database implementations support the concept of a non-null field constraint that prevents null values in a specific table column.

Das RDBMS behandelt Nullwerte durchgängig gleich als unbekannte oder fehlende Informationen, insbesondere unabhängig vom Datentyp der Spalte.

Rule 4: Dynamic On-Line Catalog

Based on the Relational Model a relational database must provide access to its structure through the same tools that are used to access the data. This is usually accomplished by storing the structure definition within special system tables.

Die Datenbankstruktur, d.h. die Datenbank und ihre Inhalte, wird in derselben Struktur wie die Daten selbst gespeichert - also in Tabellen. Diese Tabelle ist der so genannte Systemkatalog.

Rule 5: Comprehensive Data Sublanguage Rule

The database must support at least one clearly defined language that includes functionality for data definition, data manipulation, data integrity, and database transaction control. All commercial relational databases use forms of the standard SQL (Structured Query Language) as their supported comprehensive language.

Zu einem relationalen System gehört mindestens eine Abfragesprache mit einem vollständigen Befehlssatz für Datendefinition, Datenmanipulation, Datenintegrität und für die Transaktionskontrolle.

Rule 6: View Updating Rule

Data can be presented to the user in different logical combinations, called views. Each view should support the same full range of data manipulation that direct-access to a table has available. In practice, providing update and delete access to logical views is difficult and is not fully supported by any current database.

Daten können dem User in verschiedenen logischen Kombinationen dargestellt werden. Diese Kombination werden Views genannt. Jede View sollte sämtliche Datenmanipulationen erlauben, die auch auf einer Tabelle zulässig sind. In der Praxis ist es schwierig Operationen wie update oder delete auf logische Views zuzulassen.

Rule 7: High-level Insert, Update, and Delete

Data can be retrieved from a relational database in sets constructed of data from multiple rows and/or multiple tables. This rule states that insert, update, and delete operations should be supported for any retrievable set rather than just for a single row in a single table.

Daten einer relationalen Datenbank können als Menge bestehend aus Daten aus mehreren Rows oder / und mehreren Tabellen dargestellt werden. Diese Regel definiert, dass insert, update und delete Operationen auf sämtliche dargestellten Mengen als Ganzes appliziert werden müssen.

Rule 8: Physical Data Independence

The user is isolated from the physical method of storing and retrieving information from the database. Changes can be made to the underlying architecture (hardware, disk storage methods) without affecting how the user accesses it.

Der Zugriff auf die Daten durch den Benutzer oder die Anwendung muss unabhängig von den physikalischen Zugriffsmethoden oder den Speicherstrukturen der Daten sein. Es darf also nur auf die logische Struktur des Systems zugegriffen werden.

Rule 9: Logical Data Independence

How a user views data should not change when the logical structure (tables structure) of the database changes. This rule is particularly difficult to satisfy. Most databases rely on strong ties between the user view of the data and the actual structure of the underlying tables.

Information erhaltende Änderungen der Tabellenstrukturen dürfen nicht die Änderung der Logik von Anwendungen und Zugriffen bedingen.

Rule 10: Integrity Independence

The database language (like SQL) should support constraints on user input that maintain database integrity. This rule is not fully implemented by most major vendors. At a minimum, all databases do pre-serve two constraints through SQL.

Integrätsbedingungen müssen in der Abfragesprache definierbar sein. Die Regeln müssen im Systemkatalog gespeichert werden und dürfen nicht umgangen werden können.

Rule 11: Distribution Independence

A user should be totally unaware of whether or not the database is distributed (whether parts of the database exist in multiple locations). A variety of reasons make this rule difficult to implement; I will spend time addressing these reasons when we discuss distributed databases.

Der logische Zugriff auf die Daten von Anwendungen oder Ad-hoc-Programmen darf sich beim Wechsel von einer nicht-verteilten zu einer verteilten Datenbank nicht ändern.

Rule 12: Nonsubversion Rule

There should be no way to modify the database structure other than through the multiple row data-base language (like SQL). Most databases today support administrative tools that allow some direct manipulation of the data structure.

Integritätsbedingungen, die über die Datenbanksprache definiert wurden, dürfen sicht nicht mit Hilfe von Low-Level-Sprachen umgehen lassen.

8.3.4 Oracle Database Engine

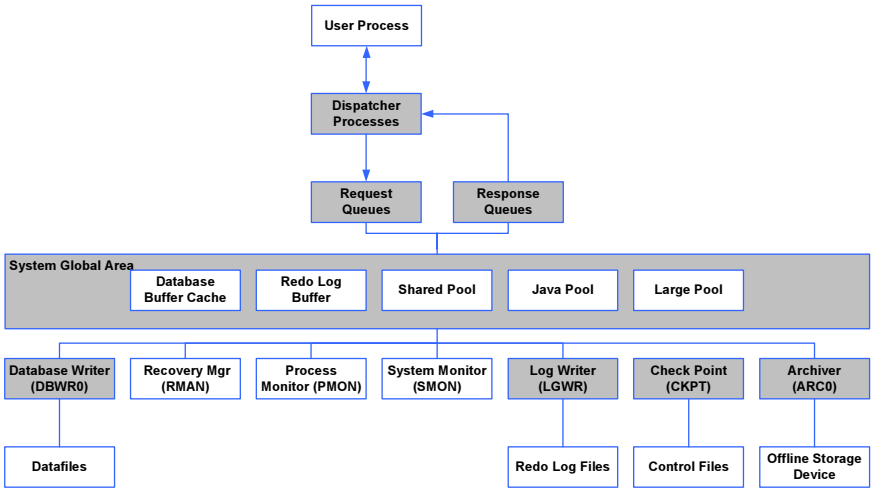

Abbildung 8.6 Aufbau und Prozesse der Oracle Database Engine

Die Oracle Database Engine ist die relationale DB Engine mit der längsten Vorgeschichte (**Abbildung 8.6**). Sie ist bereits über 20 Jahre alt. Oracle ist heute im Bereich RDBMS der Marktführer.

- *Dispatcher Process:* Daemon zur Verteilung der Anfragen auf verschiedene Prozesse (Listener).
- *Request Queues:* Warteschlange für Anfragen.
- *Response Queues:* Warteschlange für Antworten.
- *System Global Area:* Oracle Datenbankinstanz.
- *Database Buffer Cache:* Der Database Buffer Cache speichert die zuletzt verwendeten Kopien von Datenblöcken mittels LRU-Algorithmus. Der Database Buffer Cache besteht aus verschiedenen Sub-Caches. Dem Cache für Standard-Blocks, dem Keep Buffer Cache, welcher Blöcke zum Wieder verwenden speichert, dem Recycle Cache, welcher nicht verwendete Blöcke aus dem Speicher entfernt und der Primary Block Size Store.

- *Redo Log Buffer:* Der Redo Log Buffer ist ein Circular Buffer und speichert Änderungen von Datenblöcken.
- *Shared Pool:* Der Shared Pool speichert die zuletzt verwendeten SQL Statements und Datendefinitionen. Er besteht aus einem Library Cache und einem Data Dictionary Cache. Der Share Pool speichert die zuletzt verwendeten SQL Statements und Datendefinitionen. Der Library Cache speichert die zuletzt verwendeten PL/SQL-Befehle mittels LRU-Algorithmus (Last Recentliy Used).
- *Java Pool:* Der Java Pool ist optional und wird für Java verwendet.
- *Large Pool:* Der Large Pool ist ein optionaler Speicherbereich im SGA. Er wird für Session Memory verwendet bei Shared Server. Weitere Anwendungen sind I/O und Backup/Restore (RMAN).

Background Prozesse

- *Database Writer (DBWR0):* Der Database Writer schreibt Dirty Buffers vom Database Buffer Cache in die Database Files.
- *Recovery Manager (RMAN):* Der Recovery Manager ist ein Utility zur Sicherung und Wiederherstellung von Datenbanken
- *Process Monitor (PMON):* Fehlerhafte Prozesse im PGA werden durch den Process Monitor aufgeräumt mittels Zurückrollen der Transaktion, Freigeben von Locks. Freigeben anderer Resourcen, Neustart von toten Dispatchern.
- *System Monitor (SMON):* Bei einem Crash können keine Informationen des System Global Area auf die Festplatten geschrieben werden. In diesem Fall führt der System Monitor ein automatisches Instanz-Recovery durch.
- *Log Writer (LGWR):* Der Log Writer schreibt sequentiell vom Redo Log Buffer in die Redo Log Files
- *Check Point (CKPT):* Alle drei Sekunden schreibt der CKPT-Prozess Daten in die Control-Files, um die Stellen in den Redo Logs zu kennzeichnen, bei denen ein Recovery beginnen muss. Diese Daten nennt man Checkpoints.
- *Archiver (ARC0):* Der Archiver dient zum Erstellen der Archive Log-Files.

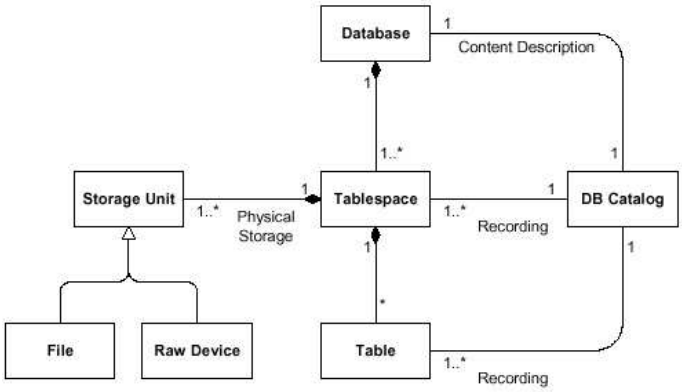

Abbildung 8.7 Die Struktur der Oracle Datenbank

Eine Oracle Datenbank besteht aus einer Anzahl von Tablespaces. Tablespaces sind logisch zusammengesetzt aus Tabellen oder anderen Datenbankobjekten wie z.B. Indizes, die physische entweder in einem "File" oder einem "Raw Device" gespeichert werden. Eine Beschreibung aller Tablespaces und Tabellen ist im DB Catalog enthalten (**Abbildung 8.7**).

8.3.5 DB2 Database Engine

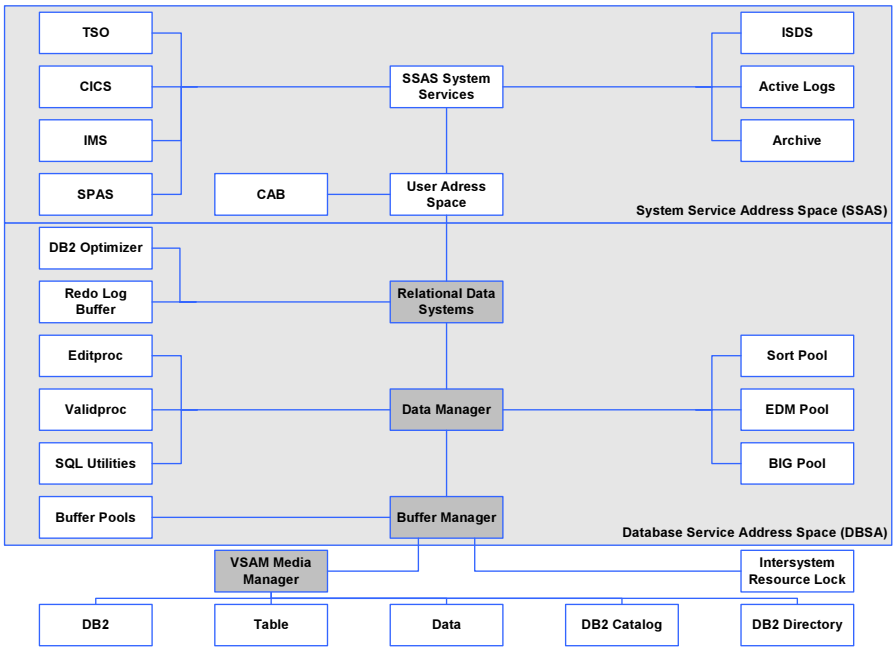

Abbildung 8.8 Aufbau und Prozesse der IBM DB2 Datenbank

Die DB2 Database Engine besteht aus 5 Tasks, die separat gestartet werden. Physikalisch besteht DB2 aus einer Sammlung von Address Spaces und Intersystem Communication Mechanismen (**Abbildung 8.8**). Die fünf Tasks sind:

- *Database Service Address Space (DBSA):* Die Komponenten zur Manipulation von DB2 Datenstrukturen.
- *System Services Address Space (SSAS):* Die Mechanismen zur Verbindung einer DB2 Datenbank zu anderen Subsystemen.
- *Distributed Data Facility (DDF):* Die Verbindung zwischen verschiedenen Datenbankinstanzen, falls eine verteilte Datenbank vorliegt.
- *Intersystem Resource Lock Manager (IRLM):* Diese Komponente ist verantwortlich für die Verwaltung von DB2 Locks (inclusive Deadlock Detection).
- *Stored Procedure Address Space (SPAS):* Die Mechanismen zur Unterstützung von Stored Procedures und Remote Procedure Calls (RPC).

Definition VSAM: *VSAM (Virtual Storage Access Method) is a file management system for IBM's larger operating systems, including its primary mainframe operating system, MVS (Multiple Virtual Storage), now called OS/390. Using VSAM, an enterprise can create and access records in a file in the sequential order that they were entered. It can also save and access each record with a key (for example, the name of an employee).*

VSAM (Virtual Storage Access Method) beschreibt eine Zugriffsmethode auf Dateien. Die Namensgebung basiert auf der Idee, Dateiinhalte wie Zellen im virtuellen Memory adressieren zu können, was mit Hilfe einer RBA (Relative Byte Adresse) möglich ist. VSAM abstrahiert die physikalischen Eigenschaften des Storage Mediums wie beispielsweise die Anzahl Cylinder. VSAM löst ISAM (Indexed Sequential Access Method), die eine Datei anhand von drei Informationen beschreibt (PRIME, INDEX und OVERFLOW) ab.

VSAM-Dateien werden auch als Cluster bezeichnet. Es fünf Varianten dieser VSAM-Cluster, KSDS (Key sequential DataSet), ESDS (Entry sequential DataSet), RRDS (Relative Record DataSet), LDS (Linear DataSet) und zFS (zSeries Files System). Ein DataSet ist nichts anderes als eine Datei, die auf einem IBM Grossrechnersystem existiert.

- *KSDS (Key Sequential DataSet):* Bei dieser Dateiform erfolgt der Datenzugriff über einen Index, der auf Betriebssystemebene in einem eigenen DataSet gespeichert wird. Gleichermaßen wird der sequentielle Zugriff unterstützt
- *ESDS (Entry Sequential DataSet):* Bei dieser Dateiform wird sequentiell auf den Datenbestand der Datei zugegriffen. Mit Hilfe von Alternativindizes kann nachträglich ein Direktzugriff mit Hilfe von Indizes (Schlüsselfelder) ermöglicht werden.
- *RRDS (Relative Record DataSet):* Der Zugriff erfolgt mit Hilfe von logischen Satznummern.
- *LDS (Linear DataSet):* Ist ein völlig unstrukturierter Bytestrom, der aber beliebig von der Anwendungssoftware interpretierbar ist.
- *zFS (zSeries Files System):* Ist eine Weiterentwicklung der HFS (Hierarchical File System) DataSets.

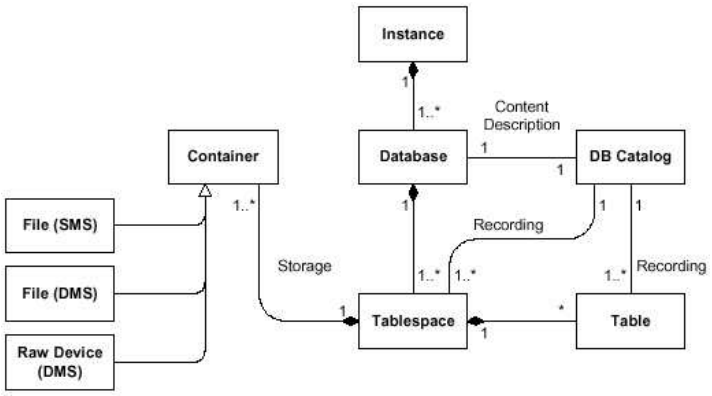

Abbildung 8.9 Die Struktur der DB2 Datenbank

Die Daten eines Tablespace sind in einer DB2 Datenbank physisch in DB2 Container gespeichert. Diese Container sind entweder "Files" oder "Raw Devices". Die Verwaltung des Speicherplatzes der Container wird entweder durch die Datenbank (Database Managed Space – DMS) oder durch das Betriebssystems (System Managed Space - SMS) durchgeführt. Eine DB2 Datenbank hat genau einen einzigen Database System Catalog, der Informationen über Datenbankobjekte (Tabellen, Indizes) speichert (**Abbildung 8.9**).

8.3.6 Microsoft SQL Database Engine

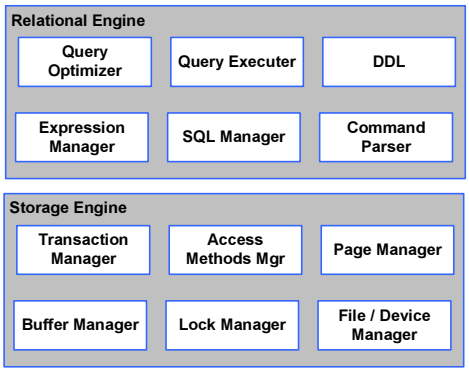

Abbildung 8.10 Die Struktur der Microsoft SQL Datenbank

Der MS-SQL Server besteht aus zwei Grundkomponenten der Relational Engine und der Storage Engine, wie in **Abbildung 8.10** skizziert.

- *Relational Engine:* Komponenten zur Ausführung der relationalen Anfragen.
- *Storage Engine:* Sämtliche Komponenten des MS-SQL Servers zur Verwaltung von gespeicherten Daten.

Das Produkt existiert in sechs Varianten [Lowe 2006], der Mobile Edition, der Developer Edistion, der Express Edition, er Workgroup Edition, der Standard Edition und der Enterprise Edition:

- *Mobile Edition (nur 32-Bit):* Für SQL Server 2000 bot Microsoft eine Windows CE-Edition des Produkts. Für die SQL Server 2005-Edition wurde diese in Mobile Edition umbenannt und Microsoft hat die Mobilfunktionen erweitert.
- *Developer Edition (32- und 64-Bit):* Die Developer Edition bietet denselben Leistungsumfang wie die Enterprise Edition (siehe unten) - mit einem wichtigen Unterschied: Aufgrund der Lizenzvereinbarung für diese Edition sind Deployments in Produktionsumgebungen untersagt.
- *Express Edition (nur 32-Bit):* Die SQL Server 2005 Express Edition dient als Ersatz für die SQL Server 2000 Desktop Edition und bietet Software-Entwicklern eine kostenlose Datenbank, welche diese ohne Lizenzgebühren zusammen mit ihrem Produkt vertreiben können. Die Express Edition kann kostenlos von der Microsoft-Website heruntergeladen werden, hat allerdings einige Einschränkungen: Einmal funktioniert sie nur mit einem einzelnen Prozessor, was die Verwendung für umfangreiche Installationen einschränkt, zweitens sind nur 1 GByte an RAM damit nutzbar und schliesslich kann die Datenbank einer Express Edition nicht grösser als 4 GByte werden.
- *Workgroup Edition (nur 32-Bit):* Die Workgroup Edition hat kein entsprechendes SQL-Server-2000-Gegenstück und ist für kleinere Unternehmen gedacht, die keine extrem leistungsfähige Datenbank benötigen. Die wichtigste Einschränkung der Workgroup Edition ist die Begrenzung auf 3 GByte verfügbaren RAM im System, was sie für umfangreiche Datenbank-Deployments ungeeignet macht. Zu den weiteren Beschränkungen gehört das Fehlen von Partitionierungsfunktionen, Parallelverarbeitung sowie indizierte Ansichten. Die Workgroup Edition bringt auch keine signifikanten Funktionen für Hochverfügbarkeit, Management und Analyse mit wie andere Editionen dieses Produkts. Trotz dieser Einschränkungen ist die Größe einer Workgroup-Edition-Datenbank nicht begrenzt.
- *Standard Edition (32- und 64-Bit):* Dies ist das direkte Gegenstück zu SQL Server 2000. Microsoft hat die Standard Edition für kleine und mittlere Unternehmen ausgerichtet, die in einem gewissen Maße E-Commerce-Services und andere geschäftsbezogene Datenbankanwendungen einsetzen. Die Standard Edition enthält bereits Funktionen, die für eine Unternehmensdatenbank erforderlich sind, wie zum Beispiel 2-Node-Clustering, unbegrenzte RAM-Unterstützung, bis zu 4 Prozessoren, unbegrenzte Datenbankgröße sowie eine Auswahl an Business Intelligence-Funktionen.
- *Enterprise Edition (32- und 64-Bit):* Die Enterprise Edition ist das Flaggschiff der Reihe. Sie bietet eine skalierbare Datenbankserverumgebung für Unternehmen jeglicher Größe. Die Enterprise Edition hat keine Beschränkungen in Hinsicht auf CPU, RAM oder Datenbankgröße, ermöglicht Multimode-Clustering, Onlineindizierung, Oracle-Replikation und viele weitere Funktionen.

8.3.7 MySQL Database Engine

MySQL ist die populärste Open-Source Datenbank weltweit. Sie hat sich technisch von Nischenprodukt in der aktuellen Version 5.0 zu einer Software entwickelt, die den Anspruch hat, die grossen drei Datenbankhersteller Oracle, IBM und Microsoft herauszufordern. Konkrete Erfahrungen mit der Version 5.0 sind noch sehr beschränkt. Verfügbar ist MySQL als MySQL 5.0 Pro Server und als MySQL 5.0 Community edition.

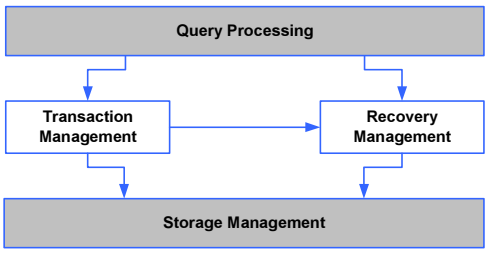

Abbildung 8.11 Konzeptionelle Architektur der MySQL Database Engine

Die Konzeptionelle Architektur der MySQL Database Engine ist sehr einfach aufgebaut, sie besteht aus den vier Teilen Query Processing, Transaction Management, Recovery Management und Storage Management, wie in **Abbildung 8.11** dargestellt.

- *Query Processing:* Die Ausführung der einzelnen Queries ist Aufgabe des Query Processing.
- *Transaction Management:* Die Verwaltung von Transaktionen mit den entsprechenden Rollback Mechanismen wird durch das Transaction Management ausgeführt.
- *Recovery Management:* Diese Komponente ist für die Wiederherstellung korrupter Datenbestände zuständig.
- *Storage Management:* Die physikalische Speicherung der Daten erfolgt durch das Storage Management.

Die Prozesse der MySQL Database Engine

- *Embedded DML Precompiler:* Der Precompiler extrahiert die einzelnen SQL Statements aus den vom Client gesendeten API Aufrufen.
- *DLL Compiler:* Die SQL Commands, die durch einen Administrator gestartet werden, werden direkt vom DLL Compiler verarbeitet.
- *Query Parser:* Der Query Parser prüft die SQL Commands und sendet diese nur an den Query Preprocessor weiter, falls sie dem geforderten Eingangs-Alphabet entsprechen.
- *Query Preprocessor:* Syntaktische Prüfung der SQL Commands.
- *Security / Integration Manager:* Der Security / Integration Manager gleicht das SQL Statement mit der ACL (Access Control List) ab.
- *Query Optimizer:* Die Optimierung der SQL Anfrage erfolgt durch den Query Optimizer.

- *Execution Engine:* Diese Komponente führt den Query aus.
- *Concurrency-Control Manager:* Der Concurrency-Control Manager garantiert die separate und unabhängige Ausführung eines Query.
- *Transaction Manager:* Der Transaction Manager ist verantwortlich für die Erfüllung der ACID (Atomic, Consistent, Isolated, Durable) Kriterien.
- *Log Manager:* Sämtliche Operationen werden durch den Log Manager aufgezeichnet.
- *Recovery Manager:* Diese Komponente ist verantwortlich für das Restore einer korrupten Datenbank.
- *Resource Manager:* Der Resource Manager akzeptiert Requests der Execution Engine und setzt diese als Data Control Language Requests um.

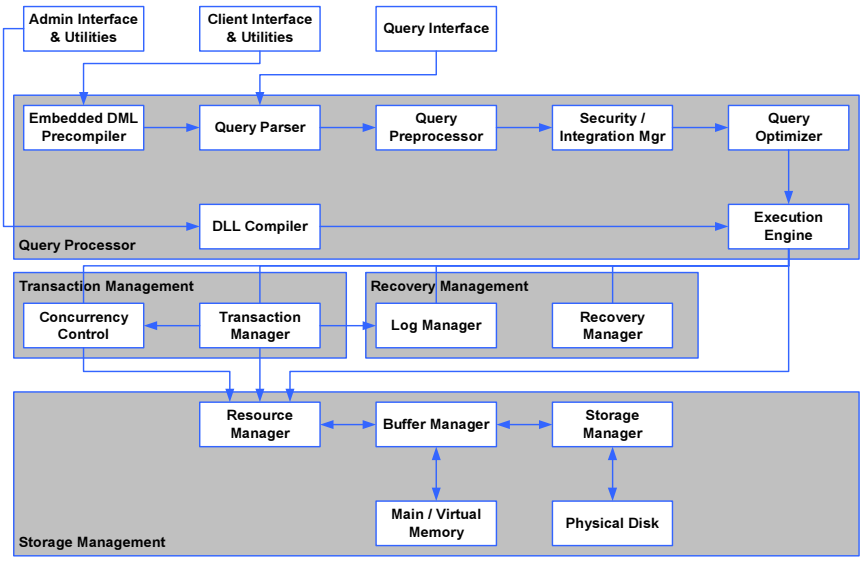

Abbildung 8.12 Aufbau und Prozesse der MySQL 4.0 Database Engine

Die Architektur von MySQL 5.0 unterscheidet sich strukturell nicht mehr vom Aufbau der logischen Architektur und der Building Blocks jeder anderen DB Engine (**Abbildung 8.12**). Neue Eigenschaften sind [Rüttger 2006]:

- ACID-Transaktionen zum Erstellen zuverlässiger und sicherer geschäftskritischer Anwendungen
- Gespeicherte Prozeduren (Stored Procedures) für höhere Produktivität von Entwicklern
- Trigger zum Erzwingen komplexer Geschäftsregeln auf Datenbankebene
- Sichten (Views), um sicherzustellen, dass sensible Informationen nicht gefährdet werden
- Informationsschema (Information Schema) für einfachen Zugriff auf Metadaten

- Verteilte Transaktionen (XA) zur Unterstützung komplexer Transaktionen über mehrere Datenbanken
- Modulare Speicher-Engine-Architektur für maximale Flexibilität
- Speicher-Engine Archive für historische Daten
- Speicher-Engine Federated um eine logische Datenbank von verschiedenen physikalischen Servern zu erstellen

8.3.8 Teradata Database Engine

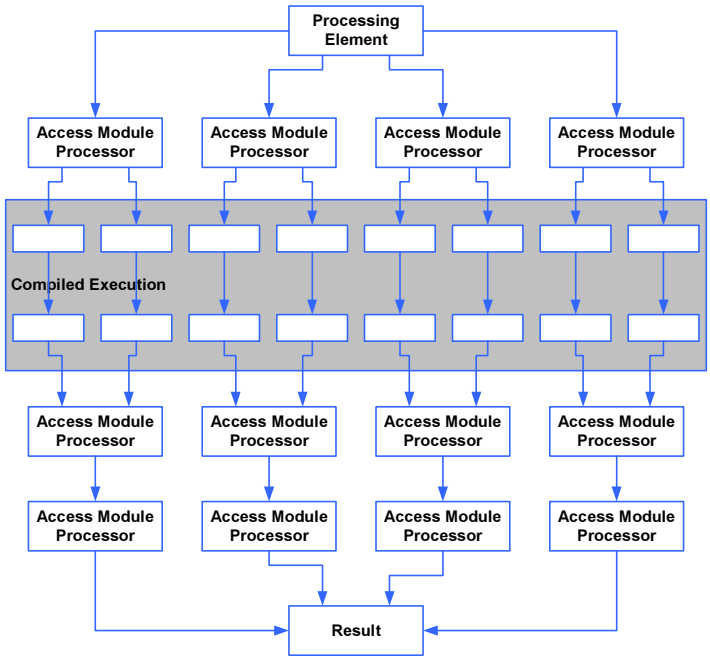

Abbildung 8.13 Paralelle Query Verarbeitung der Teradata Database Engine

Die Teradata DB Engine ist spezialisiert auf DWH Anwendungen und realisiert eine parallele Architektur, die als SMP (Symmetric Multi-Processing) oder MPP (Massively Parallel Processing) Cluster umgesetzt werden kann (**Abbildung 8.13**). Die maximale Kapazität der Datenbank ist 1 Petabyte. Sie kann auf bis zu 512 Nodes verteilt werden. Das Architekturmodell entspricht demjenigen der Connection Engine, nur dass statt der Nexus eine spezielle Hardwarekomponente namens BYNET eingesetzt wird. Die konkrete Verteilung ist statisch und wird konfiguriert. Jeder einzelne Node kann aus einem oder zwei Processing Elements bestehen. Er kann maximal 8 Access Module Processors (AMP) verwalten und fügt über einen LAN Gateway und einen Channel Gateway für das BYNET. Die einzelnen Bereiche der Data Storage werden verschiedenen Nodes zugeordnet. Dabei wird

immer von einer einzigen Tabelle ausgegangen die ohne Partitionierung abgelegt wird. Die Partitionierung erfolgt automatisch. Das Laden der operativen Daten kann parallel erfolgen.

8.4 Petabyte Computing Infrastructures

Petabyte Computing Infrastructures unterstützen die Verarbeitung von Datenmengen im Petabyte Bereich. Dazu müssen die entsprechenden Infrastrukturen für Netzwerke, CPU Power und Storage Capacity bereitgestellt werden. Auf Netzwerkebene sind zum Transport von Daten im Petabyte Bereich Bandbreiten von 100 Gbps und höher notwendig [Bunn et Al. 2002]. Auf Ebene CPU Power stehen heute die entsprechenden Supercomputer bereits zur Verfügung. Die Storage Capacity wird heute mit Architekturen realisiert, die auf den Prinzipien von Grid Computing basieren. Es wird mit vollständig virtualisiertem Storage gearbeitet.

8.4.1 Explicit Control Protocol

Auf Netzwerkebene werden vor allem die Schwächen des TCP/IP Protokolls durch Erweiterungen kompensiert. TCP hat leider eine ungünstige Eigenschaft, was die Mechanismen für das Congestion Controlling (Management im Falle der Überlastung des Netzes) bei sehr hoher Bandbreitenauslastung betrifft. Als CSMA/DA (Carrier Sense Multiple Access/Collision Detection) Protokoll, welches dasselbe Medium für verschiedene Verbindungen auf ein und demselben Kanal verwendet, neigt es dazu, ein hoher Last zunehmend Pakete zu verlieren. Eine weitere Eigenschaft ist die Tatsache, das TCP keine faire Lastverteilung vorsieht. Es kann also sein, dass bestimmte logische Verbindungen mehr Bandbreite benützen können als andere. Eine Reihe von Protokollerweiterungen versuchen, diesen Nachteil zu kompensieren. Eine davon ist XCP (Explicit Control Protocol), ein Protokoll, welches die Reservierung von gewünschter Bandbreite erlaubt. Zu diesem Zweck werden die TCP Pakete mit einem so genannten Congestion Header erweitert. Dieser Congestion Header erlaubt es dem Sender, eine gewünschte Erhöhung der Bandbreite zu definieren. Falls die Netzverbindung die entsprechende Bandbreite nicht zur Verfügung stellen kann, also ein Router nicht den notwendigen Durchsatz liefert, so kann der Empfänger dies über das ACK Paket an den Sender zurückmelden. Der Sender kann dann alternative Wege zum Empfänger initiieren oder andere Massnahmen ergreifen [Falk et al. 2003].

8.4.2 Grid Architecture Basics

Die beiden Forscher Ian Foster und Carl Kesselmann haben 1999 in ihrem Buch "The Grid – Blueprint of a New Computing Infrastructure" den Begriff Grid Architecture geprägt [Foster, Kesselmann 1999]. *The word "grid" is chosen by analogy with the electric power*

grid, which provides pervasive access to power and, like the computer and a small number of other advances, has had a dramatic impact on human capabili-ties an society.

Grid Computing (englisch grid computing = Gitterberechnung) bezeichnet alle Methoden, die Rechenleistung vieler Computer innerhalb eines Netzwerks so zusammenzufassen, dass über den reinen Datenaustausch hinaus die (parallele) Lösung von rechenintensiven Problemen ermöglicht wird (verteiltes Rechnen). Jeder Computer in dem "Gitter" ist eine, den anderen Computern gleichgestellte Einheit. Damit kann, zu deutlich geringeren Kosten, sowohl die Kapazität als auch die Rechenleistung heutiger Supercomputer übertroffen werden. Grid-Systeme skalieren sehr gut: durch Hinzufügen von Rechnern zum Netz oder Zusammenfassen von Grids zu Meta-Grids erhöht sich die Rechenleistung in entsprechendem Masse.

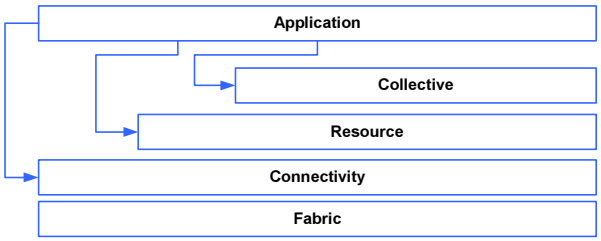

Abbildung 8.14 Grid Protocol Architecture

Die grundlegenden Bestandteile der Grid Protocol Architecture sind in **Abbildung 8.14** dargestellt.

- *Fabric:* Die Fabric ist die Schnittstelle zur lokalen Kontrolle, z.B. einem Speichersystem.
- *Connectivity:* Die Connectivity implementiert einfache und sichere Kommunikations- und Authentifizierungsprotokolle sowie ein Single-Sign-on-Verfahren, mit dem man sich nur einmal am Grid anmelden muss, auch wenn man über Institutionsgrenzen hinweg auf Ressourcen zugreift.
- *Resource:* Die Resource verwaltet eine einzelne Resource.
- *Collective:* Der Collective organisiert die Interaktion zwischen mehreren Ressourcen.
- *Application:* Applikation, die in der Umgebung einer virtuellen Organisation ablaufen.

Es bestehen bereits eine Reihe von verschiedenen Grid Architektur Ausprägungen, wie beispielsweise das Globus Toolkit [Sandholm, Gawor 2003] oder die Open Grid Services Architecture [Tuecke et al. 2003].

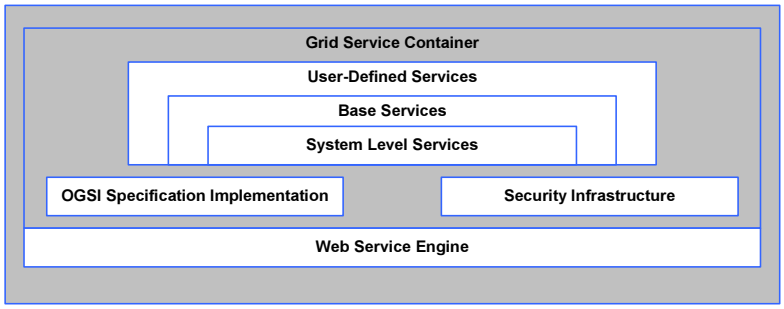

Abbildung 8.15 Core Components des Globus Toolkits Version 3

Das Globus Toolkit basierent auf dem so genannten Grid Service Container, dem zentralen Mechanismus zur Verarbeitung von Requests (**Abbildung 8.15**).

- *OGSI Specification Implementation:* Alle Open Grid Services Infrastructure Dienste sind in dieser Komponente realisiert worden.
- *Security Infrastructure:* SOAP und Transport Level Message Protection.
- *System Level Services:* Basis Run Time Services.
- *Base Services:* Program Execution, Data Management und Information Services.
- *User-Defined Services:* Dienste, die abhängig von der zu realisierenden Applikation implementiert werden.
- *Grid Service Container:* Kontrolliert den Service Lifecycle und die Verteilung respektive die Zuteilung von Service Aufrufen.
- *Web Service Engine:* Diese Komponente ist verantwortlich für die Realisierung von XML Messaging.
- *Hosting Environment:* Das Hosting Environment realisiert die Web Server Funktionalität eines Grid Services und ist beispielsweise über HTTP aufzurufen.

8.4.3 Das Information Power Grid

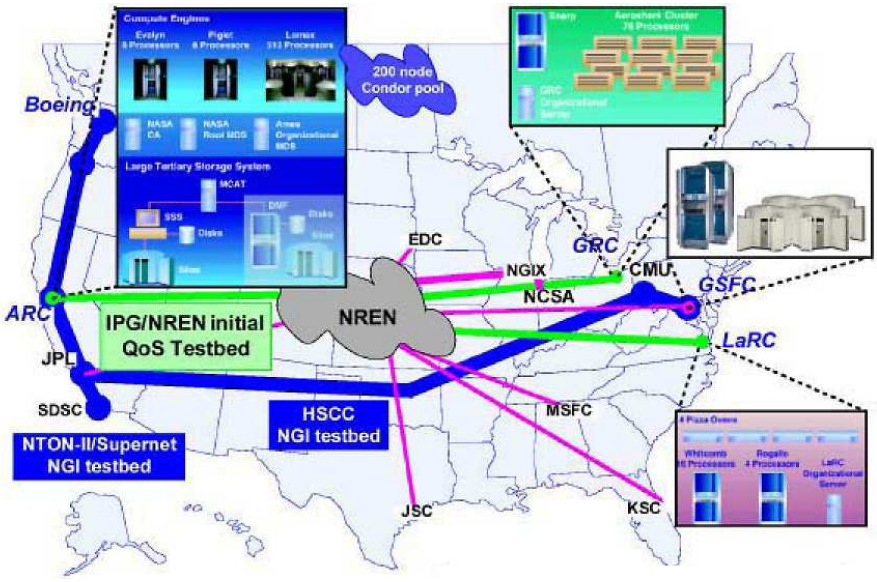

Abbildung 8.16 Das Information Power Grid der NASA

Das Information Power Grid der NASA ist eine Data Grid Versuchsumgebung, die auf der Globus Architektur basiert [Johnston et Al. 2000]. Sie besteht aus 1944 Prozessoren, 665 GB System Memory und 17.7 Terabyte Storage und wurde im April 2003 bereitgestellt (**Abbildung 8.16**). Als Hardware wurden SGI Origin Workstations eingesetzt.

Fazit dieser Versuchsanordnung war, dass die Grid Architektur geeignet ist für Anwendungen, die substantielle Ressourcen benötigen und grosse Datenmengen generieren respektive konsumieren. Die Integration verschiedener Datenarchive, die Bereitstellung geeigneter Data Caching Mechanismen und die notwendige Bandbreite wurden als die grossen Herausforderungen für die Zukunft angesehen.

8.4.4 Das Computing Model des LHC

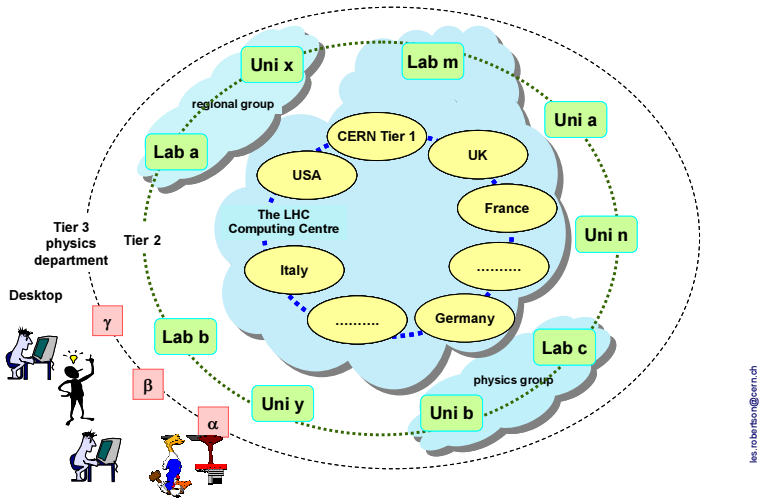

Abbildung 8.17 LHC Computing Grid (Robertson 2001)

Das Computing Grid des Cern für die LHC Experimente ist in mehrere Tiers zusammengefasst (**Abbildung 8.17**). Die Planung aus dem Jahr 2001 sah folgende Schichtung vor:

- *Tier 0:* Die Infrastruktur des Cern, die ausschliesslich für den Teilchenbeschleuniger bereitsteht. Sie umfasst 1727 Prozessoren und 1.2 Petabyte Disk Kapazität.
- *Tier 1:* Die Infrastruktur des Cern, die zu einem bestimmten Zeitpunkt, also während der Durchführung einzelner Versuche, eingesetzt wird. Sie stellt 832 Prozessoren und weitere 1.2 Petabyte Storage zur Verfügung.
- *Tier 2:* Die so genannten Regional Centers, die ihrererseits weitere 4974 Prozessoren und 8.7 Petabyte Storage bereitstellen.
- *Tier 3:* Über diese Schicht können die Wissenschaftler auf die Versuchsdaten zugreifen und diese Auswerten.

8.4.5 Pressemeldung 500 MByte pro Sekunde

Pressemeldung (Networld) vom 26.4.2005:

Das wissenschaftliche Grid-Computing hat im Rahmen des Large Hadron Collider Computing Grids einen wichtigen Meilenstein erreicht: Acht große Rechenzentren in Europa und den USA haben über zehn Tage hinweg einen kontinuierlichen Datenstrom von 500 MByte pro Sekunde ausgetauscht - insgesamt etwa ein halbes Petabyte an Daten. Wollte man diese Datenmenge über eine ISDN-Leitung verschicken, würde dies etwa 3.000 Jahre dauern.

Die Daten wurden vom europäischen Forschungszentrum CERN in Genf verschickt, um einen Datenaustausch zu simulieren, der ab 2007 notwendig sein wird, um die Daten des Large Hadron Collider (LHC) zu verarbeiten. Der 27 km lange, kreisförmige Teilchenbeschleuniger soll über einen Zeitraum von zehn Jahren rund um die Uhr einen Datenstrom von etwa 1.500 MByte pro Sekunde liefern.

Im LHC werden Protonen oder Bleikerne mit nahezu Lichtgeschwindigkeit aufeinander geschossen, wobei Unmengen kleinerer Elementarteilchen entstehen, die Einblicke in die innerste Struktur der Materie geben sollen. Der LHC wird so zum datenintensivsten Experiment der Physik: In jeder Sekunde müssen über 100 Millionen Messdaten registriert, etwa drei Millionen GByte an Daten jährlich gespeichert und für weltweit verteilte Wissenschaftlergruppen aufbereitet werden.

Da ein einzelnes Rechenzentrum diesen Bedarf nicht mehr abdecken kann, sollen die Daten und die benötigte Rechenleistung in einer hierarchischen Schichtenstruktur über die ganze Welt verteilt und die einzelnen Standorte über eine besonders leistungsfähige Internetstruktur miteinander verknüpft werden. Der Aufbau dieser Internetstruktur ist mit einer Reihe von Meilensteinen verknüpft, von denen der zweite jetzt erreicht wurde: Für zehn Tage verteilte das CERN einen Datenstrom von 500 MByte pro Sekunde an sieben große Rechenzentren in den USA und in Europa. Der nächste Meilenstein wird für Sommer 2005 angepeilt: Dann sollen viele weitere Rechenzentren angebunden werden und ein stabiler Dauerbetrieb über drei Monate stattfinden.

Die Daten-Infrastruktur ist dabei in mehreren Schichten angelegt: Das CERN definiert als Schicht 0, wo die Experimente durchgeführt werden, wird die Daten ab 2007 weltweit auf elf Rechenzentren der so genannten Schicht 1 verteilen. Von dort laufen sie auf etwa 100 Zentren der Schicht 2, bis sie schließlich mit Schicht 3 in wissenschaftliche Institute und mit Schicht 4 auf mehrere tausend Arbeitsplätze der beteiligten Wissenschaftler verteilt sind. Die deutsche Schaltstelle für das schnellste Rechnernetz der Welt - ein Knoten der Schicht 1 - ist das Forschungszentrum Karlsruhe, das vom DFN mit einer Bandbreite von 1.250 MByte pro Sekunde vom Deutschen Forschungsnetz (DFN) angebunden wurde.

8.4.6 Data Grid Architecture

Die Data Grid Architecture ist eine Ausprägung der Grid Architecture, die die Fabric und die Collective Komponente dieser Architektur auf die Verarbeitung von grossen Datenmengen hin spezifiziert.

Die Grid Fabric enthält alle Resourcen, auf die über Grid Protokolle zugegriffen werden kann. Eine Resource ist eine logische Entität, wie beispielsweise ein verteiltes Dateisystem, ein Computer Cluster oder ein verteilter Rechnerverbund. In diesem Fall enthält die Realisierung einer Resource eine Reihe von internen Zugriffprotokollen, die jedoch aus Sicht Grid Protocol nicht wichtig sind. The Komponenten einer Farbric implementieren lokale Operationen, die auf bestimmte Resourcen als Resultat von "Sharing Operations" auf höherer Ebene zugreifen. Es existiert eine gegenseitige Abhängigkeit zwischen Funktionen auf lokaler Ebene (Fabric Level) und denjenigen auf einer höheren Ebene. Beispiels-

weise ist die Belegung im Voraus für lokale Resourcen eine Voraussetzung dafür, dass high-level Services Resourcen aggregieren können.

Der Collective Layer umfasst Protokolle zur globalen Verwaltung und zum globalen Zugriff unabhängig von der konkreten Realisierung einer bestimmten Fabric Layer Resource.

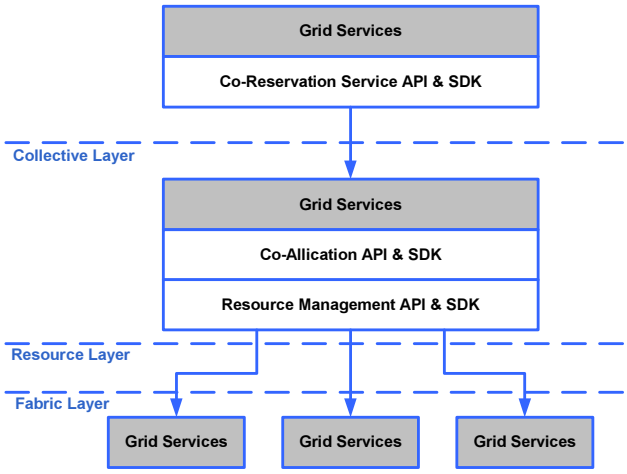

Abbildung 8.18 Collective und Resource Layer Protokolle, Dienste, API's und SDK's

Collective Funktionen können als persistente Dienst mit den zugehörigen Protokollen oder auch als System Development Kits (SDK's) mit den zugehörigen API's eingesetzt werden, um Grid Applikationen zu erstellen.

Collective Komponenten realisieren verschiedene Dienste:

- *Directory Services:* Sie erlauben VO-Teilnehmer das auffinden von VO Resourcen.
- *Co-allocation, Scheduling and Brokering Services:* Sie erlauben die Allozierung bestimmter Resoucen für einen bestimmten Zweck sowie das Scheduling von Tasks über verschiedene Resourcen hinweg.
- *Monitoring and Diagnostics Services:* Sie werden für die Überwachung von Resourcen sowie für Fehlerbehandlung und Schutzmechanismen verwendet.
- *Data Replication Services:* Diese Dienste verwalten die Storage eines VO, um den Datenzugriff hinsichtlich Performance und Kosten zu optimieren.
- *Grid-Enabled Programming System:* Diese System erlauben die Umsetzung konventioneller Programmiermodelle, um Grid Technology produktiv einzusetzen.
- *Workload Management Systems and Collaboration Frameworks:* Diese System werden auch PSE (Problem Solving Networks) genannt. Sie erlauben die Verwendung und die Verwaltung von asynchronen Workflows über verschiedenen Komponenten hinweg.
- *Software Discovery Services:* Diese Dienste finden die am besten geeignete Software und Ausführungs-Plattform für die Durchführung einer bestimmten Berechnung.

- *Community Authorization Services:* Dienste zur Realisierung von Community Policies als Grundlage für den Zugriff und die Leistungsfähigkeit von Community Resourcen.

Ein Data Grid konzentriert sich auf Services, die einem User helfen, grosse Datensets, die in verschiedenen und verteilten Repositories gespeichert sind, zu finden, zu transferieren und zu manipulieren. Im Minimum weisen Data Grid zwei Funktionen auf, ein hochleistungs- Datentransfer Mechanismus und einen skalierbaren Replikations-Mechanismus. Andere Dienste, wie Beispielsweise eine Konsistenzverwaltung, Metadata Management und Datenfilterung sind in vielen Realisierungen vorhanden [Venugopal et Al. 2006].

Einen Data Grid kann man sich als eine Sammlung von verteilten Data Centers denken, die über Grid Services und eine Grid Infrastruktur als virtueller grosser Datenspeicher dargestellt wird. Sämtliche Operationen, die auf einer lokalen Datenbank möglich sind, sind auch auf der globalen virtuellen "Meta Datenbank" möglich.

Data Grid haben folgende Eigenschaften:

- *Massive Datasets:* Datenintensive Applikationen sind Charakterisierung durch das Vorhandensein sehr grossen Datasets von einer minimalen Grösse von mehr mehreren Gigabytes. Das Ressourcen Management eines Data Grids muss minimale Latenzzeiten beim Datentransfer garantieren, Replikate über heterogene Umgebungen hinweg erzeugen können und eine Vielzahl von Storage Ressourcen verwalten können.
- *Shared Data Collections:* Das Ressource Sharing eines Data Grids muss unter anderem Mechanismen zur Verwaltung von Shared Data Collections bereitstellen. Teilnehmer einer *wissenschaftlichen* Auswertung müssen ihre Resultate im selben Repository wie die Rohdaten speichern können, damit sie auf dieselbe Art und Weise global zugänglich sind.
- *Unified Namespace:* Sämtliche Daten in einem Data Grid teilen denselben globalen Namensraum. Dies bedeutet, *dass* jedes Datenelement einen eineindeutigen logischen Namen haben muss. Dieser logische Name muss auf eine oder mehrere physiche Namen, die auf verschiedenen Storage Ressourcen gespeichert werden, abgebildet werden können.
- *Access Restrictions:* Die Authentisierung und die Autorisierung verschiedener Datenbereiche muss global *verwaltet* und nachgeführt werden können.

8.5 Fragen zum Kapitel

Nr	Frage
1	Welches sind die wichtigen Fragen, auf die die CERN Experimente eine Antwort suchen?
2	Was ist VSAM?
3	Wie funktioniert das World Wide Telescope?
4	Vergleichen Sie die Struktur der Oracle Datenbank Engine mit derjenigen der IBM DB2 Engine. Was sind die wichtigsten Unterschiede?
5	Woran errinnert Sie der Aufbau der MySQL 4.0 Database Engine, im speziellen der Query Processor?
6	Was bedeutet ACID und aus welchem Grund ist dieses Prinzip für RDBMS und Transaktionsmonitoren so wichtig?
7	Wozu werden Trigger in einer Datenbank eingesetzt?
8	Auf welcher Flynn Klasse basiert die Terradata DB Engine?
9	Was ist GRID Computing?
10	Was ist der Unterschied zwischen einem "Raw Device" und einem "File System" bezogen auf eine Datenbank?

8.6 Übung zum Kapitel

Suchen Sie die Präsentation „A Personal TP System for Everything“ von Gordon Bell, Jim Gemmell und Roger Lueder (Microsoft Research) und schauen Sie sich die Präsentation an.

Glauben Sie, dass in Zukunft ein Verschmelzen der PDA’s und Mobile Devices zu einer Infrastruktur möglich und nützlich sein wird? Kann das System MyLifeBits bereits heute als Hardware Device realisiert werden? Wenn ja, welche Technologie würden Sie einsetzen?