Die Open-Source-Grid-Umgebung Alien

Alien im Wunderland

Alien (Alice Environment) ist eine Grid-Implementation aus Open-Source-Komponenten. Dieses Cern-Projekt einer kleinen Arbeitsgruppe stellt die Infrastruktur für verteilte Simulationen, Rekonstruktionen und Analysen von sehr umfangreichen Physikdaten bereit. Kilian Schwarz für das Alien-Entwicklerteam

Alice (A Large Ion Collider Experiment, [1]) ist eines der vier Experimente des Large Hadron Colliders (LHC, [2]), die derzeit am Cern in Genf gebaut werden (siehe Abbildung 1). Wenn das Experiment in Betrieb ist, wird es bis zu 2 Petabyte Daten pro Jahr aufzeichnen – entsprechend rund 25000 handelsüblichen 80-GByte-Platten. Während seiner Laufzeit von 20 Jahren wird es jährlich mehr als eine Milliarde Datenfiles weltweit in über 50 Instituten ablegen.

Diese Datenmengen wollen zudem analysiert werden, um die relevanten Informationen zu extrahieren. Bevor der LHC fertig aufgebaut ist, werden die Physiker schon in großem Stil Teilchenkollisionen simulieren, um ein korrektes Design des Detektors zu gewährleisten.

Das Alien-Team [3] (Alien: ALIce ENvironment) entwickelte die Alice-Grid-Umgebung innerhalb des Alice-Projekts als verteilte Computing-Infrastruktur für die Simulation, Rekonstruktion und Analyse von Daten aus dem Experiment. (Der Projektname hat nichts mit dem gleichnamigen Linux-Paketmanager zu tun.)

Alien erlaubt es den Zentren der virtuellen Alice-Organisation (VO), sich als Einheit zu sehen: Jeder erreichbare Rechenknoten vermag einlaufende Programme auszuführen und die Benutzer können verteilte Datensätze als logische Dateien ohne Kenntnis des Speicherorts ansprechen.

Anders als zum Beispiel das European Data Grid (EDG, siehe Artikel in diesem Schwerpunkt) arbeitet an Alien eine vergleichsweise kleine, aber sehr aktive Gruppe von Entwicklern. Gleichwohl bezeichnet die ARDA-Arbeitsgruppe, die sich mit verteilter Analyse für LHC-Experimente beschäftigt, Alien als das Grid-Projekt mit der diesbezüglich besten Funktionalität.

Für die Alice-Benutzer begann das Alien-Grid Ende 2001 mit der Arbeit. Gegenwärtig produzieren sie damit vor allem verteilte Monte-Carlo-Daten sowie Detektorsimulationen und -rekonstruktionen an über 40 Instituten auf vier Kontinenten. Seither liefen unter Alien mehr als 26000 Alice-Jobs, was 40 CPU-Jahren entspricht und mehr als 30 Terabyte Daten produziert hat.

Viel Open Source

Besonderen Wert legten die Alien-Entwickler auf Modularität und Erweiterbarkeit. Darum haben sie die Module und Komponenten funktionell geordnet (siehe Abbildung 2) und auf deren vollkommene Eigenständigkeit geachtet. Anders als das Globus-2-Toolkit ([4], siehe Artikel) setzte Alien von Anfang an auf Webservices. Damit werden künftige Standards wie OGSA und WSRF reibungsarm implementierbar.

Alien enthält sehr viele Standard-Open-Source-Komponenten. Die Projektmitglieder waren bestrebt, deren vorhandene Funktionalität zu nutzen und sie nur wenig zu modifizieren. Das und „Extreme Programming"-Techniken haben die Software-Entwicklung merklich erleichtert und beschleunigt.

Alien benutzt exzessiv Perl, hauptsächlich wegen der vielen Open-Source-Module. Verfügbar sind zum Beispiel Kryptographie- oder SOAP-Komponenten [5]. Alien zählt einschließlich externer Module etwa drei Millionen Zeilen Code. Nur etwa ein Prozent davon musste das Alien-Team zusätzlich entwickeln.

VO-Konfiguration per LDAP

Die statische Konfiguration einer virtuellen Organisation wird zur Laufzeit von einem LDAP-Konfigurationsserver ausgelesen, was die Beschreibung der Benutzer und ihrer Rollen sowie von vorhandenen Softwarepaketen, Instituten und Grid-Diensten beinhaltet. Der erste

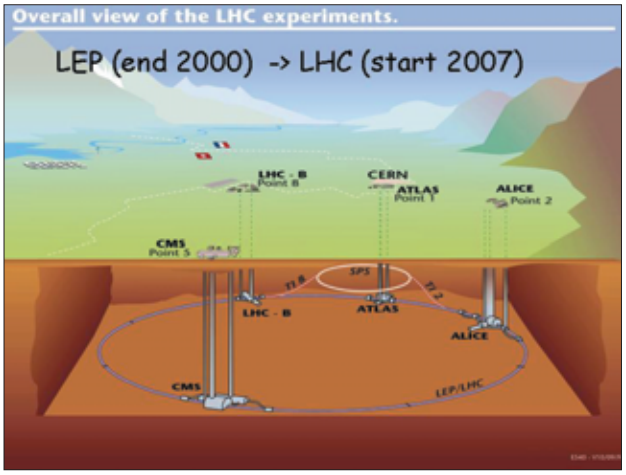

Abbildung 1: Der Large Hadron Collider an der Grenze zwischen der Schweiz und Frankreich bei Genf. Der Alice-Detektor befindet sich bei Point 2.

Alien-Eckstein war der Filekatalog (siehe Abbildung 3). Er bildet physikalische Filenamen (PFN) auf ihre logischen Pendants (LFN) ab – im Betrieb sieht der Benutzer nur den logischen.

Wie der Filekatalog LFNs in PFNs übersetzt, variiert mit der geografischen Lage und den Fähigkeiten des betreffenden Clients. Der Filekatalog hilft auch Daten zu registrieren, abzurufen und zu replizieren.

Im Interface ähnelt der Katalog einem Unix-Filesystem. So kann jeder Benutzer exklusive Lese- und Schreibrechte für seine LFNs vergeben. Dateibeschreibungen sind als Metadaten speicherbar.

Der für Alien vorliegende Filekatalog basiert zwar auf der relationalen Datenbank MySQL, sollte sich aber auch mit anderen SQL-DBs verstehen. Auch darf der Admin Teile des Verzeichnisbaums mit extra Datenbanken bedienen. Zudem können die Datenbanken auf mehreren Maschinen laufen.

Eine andere Zentralkomponente, der Package Manager, verwaltet alle Pakete automatisch, die die virtuellen Organisationen beisteuern. Damit wird es einfach, das System um Software zu erweitern, nach der eine VO verlangt. Die Pakete wissen um ihre Anforderungen an das System und kennen Abhängigkeiten von Paketen und Versionsnummern.

Alien-Services mit klaren Schnittstellen

Nach Installation, Setup (per Konfigurationsmanager) und Start beginnen die Alien-Services miteinander zu kommunizieren. Die zwischen den Komponenten (Dienste) ausgetauschten Nachrichten benutzen XML und SOAP. Vorteil des Dienste-Konzepts: Wegen der allgemein gehaltene Kommunikation braucht der Client keine Kenntnis über die Technik des Servers. In der Praxis sitzt zwischen Client und Serverapplikation meist ein vermittelnder Proxy-Service.

Das ist auch bei dem eben beschriebenen Filekatalog mit seiner SQL-Datenbank der Fall: Der Client muss nicht gegen spezielle Datenbanktreiber linken. Die Anwendung verbindet sich über einen Alien-Proxy-Treiber mit dem Proxy-Service, sodass der echte Datenbanktreiber nur dort installiert sein muss, wo deren Instanz läuft.

Derselbe Proxy-Service, der die Datenbankzugriffe übersetzt, arbeitet auch für den Authentication Service. Der Authentifizierungsdienst prüft per SASL-Proto-

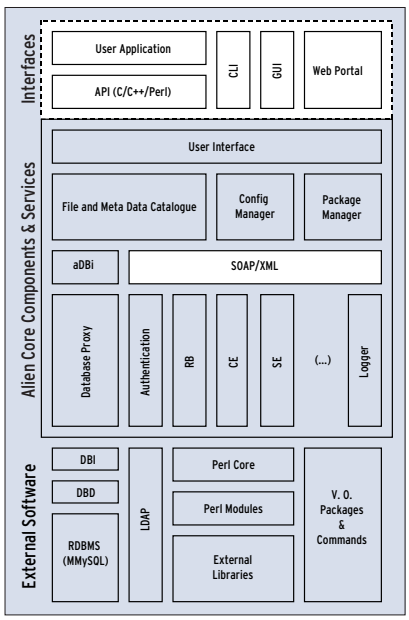

Abbildung 2: Alle Alien-Komponenten auf einen Blick.

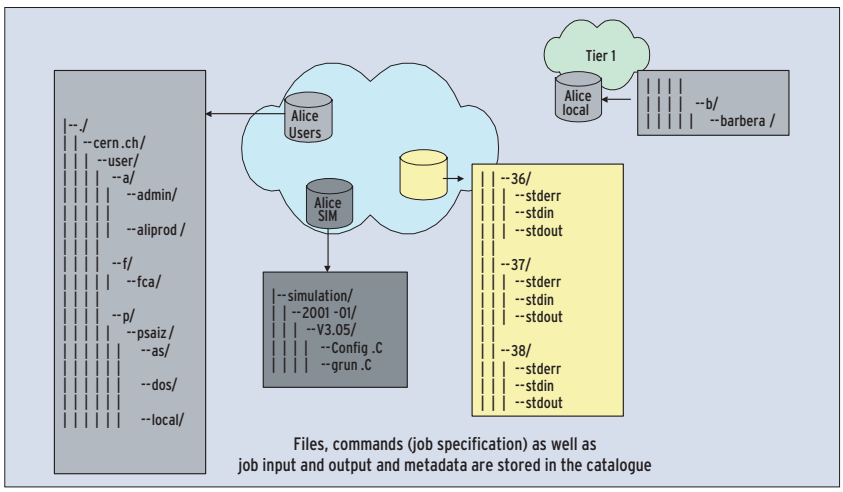

Abbildung 3: Der Alien-Filekatalog bildet physikalische Dateien – egal wo sie liegen – als logische Namen ab.

koll [6] die Credentials (englisch etwa für Ausweispapiere) der Benutzer. Alien kennt die SASL-Mechanismen GSSAPI mit Globus/GSI, AFS-Passwort, SSH-Key, X509-Zertifikate und Alien-Token (siehe Artikel „Safer Grid“).

Der Alien Resource Broker (RB) benutzt eine Pull-Architektur, im Gegensatz zur Push-Architektur der Globus-Grid-Systeme. Beim Pull fordert ein eingebundenes Computing-Element (siehe unten) neue Aufgaben vom Server an, sobald es wieder freie Kapazitäten hat. In dieser Konstellation muss der RB nicht den Status aller Systemressourcen kennen.

Wie das European Data Grid erwartet Alien von jedem übergebenen Programm eine nähere Beschreibung seiner Eigenschaften, um es ausführen zu können. Beispielsweise die Forderung nach einer bestimmten Laufzeitumgebung. Für die Beschreibung verwendet Alien die Condor Class Ads [7], gespeichert wird in Datenbanktabellen. Das Beschreibungsformat heißt Job Description Language (JDL). Das System wartet einfach darauf, dass sich ein Computing-Element mit ihm verbindet, und preist ihm als Antwort seine Fähigkeiten an.

Die Pull-Architektur erlaubt ein robustes System, das nicht auf die ständige Gegenwart aller Ressourcen angewiesen ist. Die lose Kopplung zwischen Ressourcen und den Resource Brokern, die die auszuführenden Programme an die Rechenressourcen verteilen, erlaubt es, fremde Grid-Systeme als Alien-Rechen- und Speicher-Elemente im Alien-Grid abzubilden. Nach diesem Prinzip entstand eine Alien-Schnittstelle zum European Data Grid. Seit März 2004 benutzen Forscher die EDG-Schnittstelle als Zugang zum LHC Computing Grid [8].

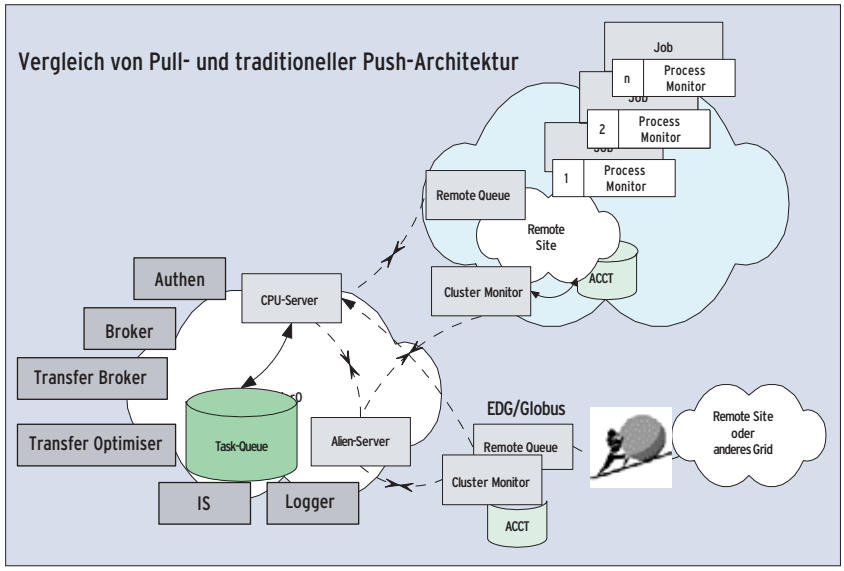

Abbildung 4: An einen Alien-Server kann man ein Alien-Rechenzentrum, aber auch ein anderes Grid anbinden.

Lokale Dienste

Üblicherweise läuft in einem Alien-Client-Zentrum eine Reihe von Diensten lokal. Ein typisches Beispiel ist der Cluster Monitor. Er fungiert als Gatekeeper, bietet also eine einheitliche Schnittstelle für alle einlaufenden Anfragen. Außerdem ist er der Proxy für die Dienste hinter dem Firewall. Ein anderer lokaler Dienst sind die Computing-Elemente (CE, Rechen-Elemente). Sie bilden die Schnittstelle zu lokalen Batchsystemen. Die Storage-Elemente (SE, Speicher-Elemente) übernehmen das Abrufen und Speichern von Daten auf und von den lokalen Massenspeichern.

Über den Process Monitor Service, der als Interface zu jedem Alien-Job fungiert, steuert der Benutzer seine laufenden Jobs. Der File Transfer Service – sein Name ist selbsterklärend – läuft auf demselben Rechner wie das Storage-Element. Die Filetransfer-Dienste haben die Entwickler als Daemons implementiert. Sie authentifizieren sich gegenseitig mit den von der Alien-CA ausgestellten Zertifikaten und führen Datenübertragungen im Namen des Benutzers durch.

Als Teil des Alien-Monitoring-Moduls installiert sich das Mona-Lisa-System. Es sammelt Statusinformationen aus dem Grid ein und veröffentlicht sie über Webdienste. Nutznießer sind Alien-Optimierprogramme und Visualisierungen, Abbildung 5 zeigt ein Beispiel.

Benutzerschnittstellen

Die Benutzer bedienen Alien über einen User Interface Layer. Die Kommandozeilen-Version hat die wichtigsten Unix-Filesystemkommandos eingebaut. Daneben gibt es ein grafisches Benutzerinterface und ein generisches Webportal. Mit ihnen kann der Benutzer große Mengen zu verteilender Jobs abschicken, inspizieren und manipulieren.

Um auch Anwendungsprogrammen Zugang zu Alien zu verschaffen, bedarf es passender Schnittstellen (APIs). Neben einem dafür geeigneten Perl-Modul stellt Alien Programmierern ein C- und ein C++ -API bereit. So wurde mit dem C++ -API – es ist Thread-safe – eine Variante des Open-Source-Projekts LUFS [9] implementiert, die den Alien-Filekatalog einklinkt. Benutzer des Konstrukts führen ein entsprechendes »mount«-Kommando aus, melden sich beim Grid an – und haben Zugriff auf das Alien-Filesystem.

Analyse mit Alien und ROOT

Die aktuelle Alien-Version eignet sich gut zum Lösen typischer Anwendungsprobleme der Hochenergiephysik (HEP), beispielsweise verteilter Simulationen, Rekonstruktionen und darauf folgender zentral koordinierter Verarbeitungsschritte. Sobald die Ergebnisse der Berechnungen vorliegen, kann der Benutzer seine Analyseprozesse starten. Viele User wollen und müssen dabei selbst geschriebene Algorithmen auf große verteilte Datensätze anwenden.

Für diese Aufgabe verwenden Alice und die meisten HEP-Experimente das Analyse-Framework ROOT [10]. Es fußt auf einem C++ -Interpreter und ist um eigene Klassen und vorkompilierte Bibliotheken gut erweiterbar. Um ROOT Alientauglich zu machen, haben die Entwickler entsprechende C++ -Klassen hinzugefügt: Die »TAliEn«-Klasse – sie basiert auf der abstrakten »TGrid«-ROOT-Klasse und benutzt das Alien-C++ -API – im-

Abbildung 5: Das Mona-Lisa-Monitoring-System zeigt angeschlossene Rechenzentren (gelbe Kreise). Zudem sind rechnende Alien-Jobs in Karlsruhe zu sehen.

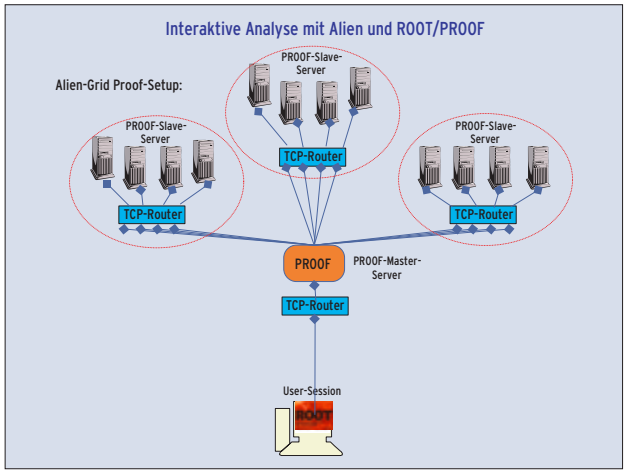

Abbildung 6: SuperPROOF und Alien schalten die PROOF-Umgebungen mehrerer Institute zusammen. Der Master-Server läuft auf der Alien-Master-Maschine.

plementiert die Basismethoden. Über sie verbinden sich die selbst geschriebenen ROOT-Routinen mit dem Alien-Grid und melden sich wieder ab. »TAliEn« regelt zudem die Zugriffe auf Aliens virtuellen Filekatalog.

Die Parallel ROOT Facility

Ein Wissenschaftler, der bei einer Analyse mehr Ressourcen benötigte, als der eigene Rechner besitzt, band in der Vor-Grid-Zeit per PROOF (Parallel ROOT Facility), die lokale Rechnerfarm ein. Dank der Parallelisierung konnte er selbst dann noch interaktiv weiterarbeiten, wenn riesige Datenmengen im Spiel waren. Üblich war, dass ein PROOF-Master die PROOF-Slave-Server verwaltet, indem er Arbeiten verteilt und die Ergebnisse einsammelt.

Heute gibt es eine PROOF-Erweiterung, die das Paket mit Alien-Grids zusammenarbeiten lässt: SuperPROOF schaltet die PROOF-Umgebungen mehrerer Institute zusammen. Jedes Zentrum wird als SuperPROOF-Slave-Server betrachtet (siehe Abbildung 6). Der SuperPROOF-Master-Server läuft auf der Alien-Master-Maschine. In einer dynamischen Umgebung startet ein Alien-Grid-Service auf Nachfrage »proofd«-Prozesse. In den Batchsystemen der lokalen Zentren laufen dedizierte Queues.

Der SuperPROOF-Master verbindet sich über einen Alien-TCP-Routing-Service mit den PROOF-Daemons in den verteilten Zentren. Alien weiß, wo die zu analysierenden Daten liegen, und lässt die C++-Analyse-Makros an jenen Zentren laufen, die die Datensätze vorhalten.

Ausblick und Zusammenfassung

Die Open Grid Services Architecture (OGSA, [11]) und das Web Service Resource Framework (WSRF) sind als gemeinsame Basis künftiger Grids in reger Diskussion. Alien hat von Anfang an auf diese Basis gesetzt, die das Einbinden neuer Standards leicht macht. Mit viel Open Source und Extreme Programming entstand in kurzer Zeit eine einfach aufgebaute, funktionsfähige Grid-Umgebung, die sich in großen Produktionen bewährt. Das relativ kleine Team beweist, dass mit vertetbarem Aufwand und freien Komponenten effiziente Grid-Software programmierbar ist.

Die Funktionalität reicht zum Lösen typischer Simulations- und Rekonstruktionsaufgaben. Für verteilte Analysen – Start von der ROOT-Kommandozeile – zerlegt man eine komplexe Anfrage in mehrere kleine. Die führt das Grid aus und reicht dem Benutzer das Ergebnis zurück. Richtig Leistung entsteht, wenn Alien viele PROOF-Cluster zu einem SuperPROOF-Cluster vereinigt.

Nach dem Alice-Experiment zeigen nun auch weitere HEP-Experimente und medizinische Projekte Interesse an Alien oder seinen Komponenten. Schon im Einsatz ist Alien als Grid-Komponente für Mammo-Grid [12] und GPCALMA [13], noch im Entstehen ist zurzeit eine Testumgebung für das GSI-Zukunftsexperiment Panda [14]. *(jk)* ■

Der Autor

Dr. Kilian Schwarz hat an der Uni Heidelberg 1997 sein Physikdiplom abgelegt und 2001 promoviert. Seither ist er wissenschaftlicher Mitarbeiter im Bereich Datenverarbeitung bei der Gesellschaft für Schwerionenforschung (GSI), Darmstadt. Als Mitglied der Alice-Kollaboration und von Alice-Offline liegt der Schwerpunkt seiner Arbeit jetzt beim Grid Computing.

Infos

[1] Alice: [http://alice.web.cern.ch/Alice/AliceNew]
[2] LHC: [http://lhc-new-homepage.web.cern.ch/lhc-new-homepage/]
[3] Alien: [http://alien.cern.ch]
P. Saiz, L. Aphecetche, P. Buncic, R. Piskac, J. E. Revsbech, V. Sego, „AliEn – ALICE environment on the GRID“: Nuclear Instruments and Methods, 2003, S. 437 bis 440
[4] Globus Project: [http://www.globus.org]
[5] SOAP: [http://www.soaplite.com]
[6] SASL: [http://asg.web.cmu.edu/sasl]
[7] Condor Classads: [http://www.cs.wisc.edu/condor/classad]
[8] LCG: [http://lcg.web.cern.ch/LCG/]
[9] LUFS: [http://lufs.sourceforge.net/lufs]
[10] ROOT: [http://root.cern.ch]
[11] OGSA: [http://www.globus.org/ogsa]
[12] Mammo-Grid: [http://www.gridstart.org/MAMMOGRID.shtml]
[13] GPCALMA: [http://content.aip.org/APCPCS/v682/i1/67_1.html]
[14] Panda: [http://www.gsi.de/zukunftsprojekt/experimente/hesr-panda/index.html]