Workflows für das frühe Universum

Die ProC-Workflowengine

Wolfgang Hovest · Jörg Knoche
Torsten Enßlin

Kosmologie und Informatik

Was verbindet Kosmologie und Informatik? Die Datenauswertung moderner kosmologischer Experimente ist ohne die Informatik nicht mehr zu bewältigen. Der Planck-Satellit zur Vermessung der Mikrowellen-Reliktstrahlung des Urknalls ist ein solches Experiment. Wie die Informatik in diesem physikalischen Grundlagenexperiment eingesetzt wird, soll dieser Artikel erläutern. Aber dafür ist ein wenig Kosmologie notwendig.

Das Universum, in dem wir heute leben, mit seinen Galaxien, Sternen und Planeten ist ursprünglich, vor 14 Milliarden Jahren, wesentlich kleiner, dichter, heißer und gleichförmiger gewesen und hatte damals keine der heute augenfälligen Strukturen. Dies zeigt uns das Licht, welches den Raum seit dem Urknall durchquert. Dieses Licht ist heute recht kalt, es hat eine Temperatur von nur drei Grad Kelvin über dem absoluten Nullpunkt und ist daher hauptsächlich als Mikrowellenstrahlung messbar (Abb. 1). Doch kurz nach dem Urknall, bis zum Jahre 380.000, war dieses Licht mehr als tausendmal heißer und wurde ständig an einem „Nebel“ von freien Elektronen gestreut. Zu diesem Zeitpunkt verbanden sich die freien Elektronen mit Protonen zu neutralem Wasserstoff und der das Licht streuende „Nebel“ lichtete sich schlagartig.

Seit diesem Zeitpunkt fliegt dieses Licht praktisch geradlinig durch das Universum, bis heute, wo es von einem Satelliten wie Planck vermessen werden kann. Die resultierende Himmelskarte zeigt die Temperatur des Lichtes an dem Ort seiner letzten Streuung in der „Nebelwand“. Wir bekommen ein Abbild der Temperaturverhältnisse des Universums 380.000 Jahre nach dem Urknall, wie sie sich auf einer Kugelschale um uns herum zeigen.

Was wir dort sehen, ist extrem gleichförmig. Wenn man die durch unsere eigene Bewegung hervorgerufene dipolare Temperaturstruktur korrigiert, hat der Mikrowellenhintergrund überall fast dieselbe Temperatur, 2.725 Kelvin. Aber dennoch gibt es winzigste Temperaturabweichungen von einigen hunderttausendstel Grad, die für Kosmologen überaus spannend sind. Sie wurden durch Inhomogenitäten in dem ansonsten extrem gleichförmigen frühen Universum erzeugt (siehe Abb. 3 unten). Zum Beispiel gab es Schallwellen im frühen Universum, ausgelöst durch das schwerkraftgetriebene Zusammenballen von Materie, die dem Mikrowellenhintergrund Temperaturabweichungen auf Winkelskalen von einem Grad aufgeprägt haben. Weitere Effekte haben auch ihre charakteristischen Muster hinterlassen. Kann man diese Muster erkennen und damit die Stärke dieser einzelnen Effekte bestimmen, so kann man viel über unser Universum lernen: wie alt es ist, wie viel normale Materie es darin gibt, und wie viel der mysteriösen Dunklen Materie und Energie darin enthalten sind.

Dafür braucht man eine hochpräzise Karte dieser winzigen Temperaturschwankungen, die von jeglichen Kontaminationen durch andere astro-

DOI 10.1007/s00287-009-0390-1

Wolfgang Hovest · Jörg Knoche · Torsten Enßlin
Max Planck Institut für Astrophysik,
Karl-Schwarzschild-Str. 1, 85748 Garching b. München
E-Mail: {woho, knoche, ensslin}@mpa-garching.mpg.de

Zusammenfassung
Die Planck-Satellitenmission zur Vermessung der Mikrowellenstrahlung des Urknalls stellt hohe Anforderungen an die Informatik. Wie kann die Expertise von Hunderten von Wissenschaftlern, die ihre Codes und Arbeitszeit zur Verfügung stellen, zu einer kohärenten, nachvollziehbaren Datenanalyse zusammengeführt werden? Am Max-Planck-Institut für Astrophysik wurde dafür eine datenbankgestützte wissenschaftliche Workflowengine entwickelt, die sich den Anforderungen aus Kosmologie und wissenschaftlicher Soziologie stellt.

physikalische Strahlungsquellen befreit wurde (Abb. 2). Die Statistik der Temperatur- und Polarisationsfluktuationen in dieser Karte muss mit theoretischen Vorhersagen der kosmologischen Modelle verglichen werden. Daraus können dann Rückschlüsse über die Beschaffenheit des heutigen und frühen Universums gemacht werden. Dies ist die Mission des Planck-Satelliten und die Aufgabe seines Bodensegments zur Datenanalyse.

Komplexität der Datenanalyse

Der Planck-Satellit ist 1,5 Millionen Kilometer von der Erde entfernt stationiert am zweiten Lagrangepunkt des Sonne-Erde-Systems, wo er synchron mit der Erde die Sonne umrundet und beiden immer nur dieselbe Seite zuwenden kann. Er rotiert um seine auf Erde und Sonne ausgerichtete Achse und scannt dabei den Himmel in kreisförmigen Streifen ab. Die 74 Detektoren nehmen bei neun verschiedenen Frequenzen zwischen 30 und 900 GHz Zeitreihen des Temperatursignals des Mikrowellenhintergrunds auf, während dieser an den Detektoren vorbeizieht. Täglich werden diese Daten zur Erde gesendet, wo sie über Perth (Australien) und Darm-

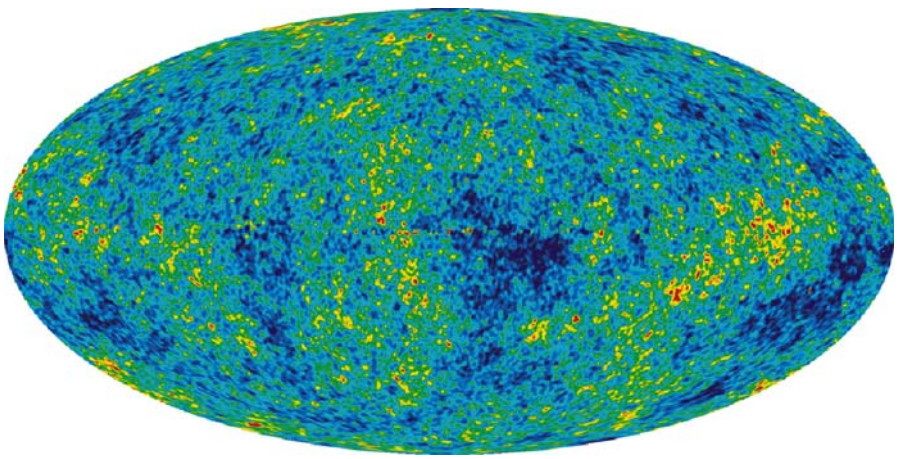

Abb. 1 Himmelskarte der komischen Hintergrundstrahlung. Dargestellt sind winzige Temperaturunterschiede. Image copyright: NASA/WMAP Science Team

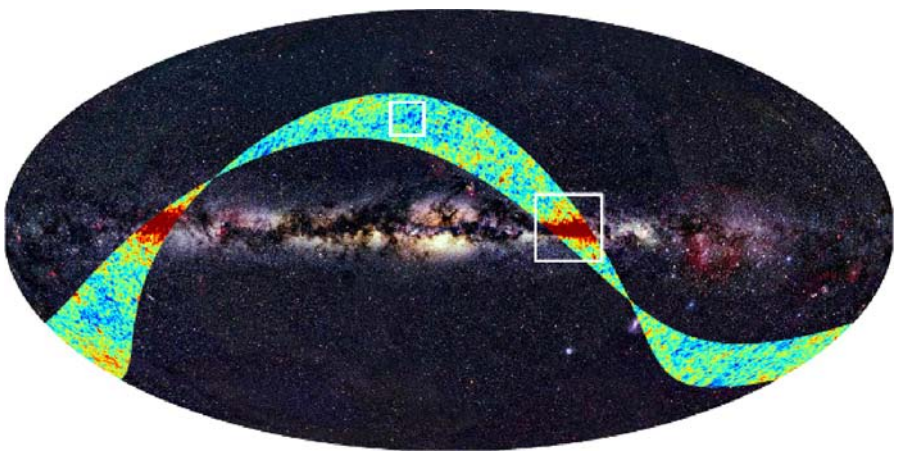

Abb. 2 Level-2-Planck-Himmelskarte aus den Daten des 15-tägigen „First Light Surveys". Zur Verdeutlichung der roten Vordergrundstruktur ist ein Bild der Milchstraße unterlegt. Image copyright: ESA, LFI and HFI Consortia (Planck); Background image: Axel Mellinger

Abb. 3 „Riding Early Waves" – eine lockere Einführung in das frühe Universum: http://www.mpa-garching.mpg.de/mpa/institute/news_archives/news_cosmic_01/news_cosmic_01-de.html. Image copyright: Jojo Ensslin

stadt an die Datenverarbeitungszentren in Paris und Triest geleitet werden. Dort werden die Daten in dem ersten Level der Verarbeitung auf Vollständigkeit kontrolliert und kalibriert. In Level 2 werden daraus Himmelskarten bei den neun Frequenzen konstruiert.

Wenn nur der kosmische Mikrowellenhintergrund gemessen würde, wären diese Karten praktisch identisch. Doch leider gibt es astrophysikalische Strahlungsquellen wie den Staub unserer Milchstraße, die sich in denselben Frequenzen bemerkbar machen – glücklicherweise aber mit einem jeweils anderen Frequenzspektrum, sodass sich diese Komponenten in Level 3 von der Mikrowellenstrahlung des Urknalls separieren lassen. Die resultierende Temperaturkarte von Letzterem muss nur noch mit allen denkbaren kosmologischen Szenarien verglichen werden. Dies kann nur in einer statistischen Art und Weise geschehen, da die genaue Ausprägung der Temperaturstrukturen nicht vorhersagbar ist, wohl aber deren statistische Eigenschaften. Deren sogenannte Leistungsspektren müssen für alle infrage kommenden Szenarien berechnet und das Maß der Verträglichkeit der Daten mit diesen bestimmt werden. Dies erfordert das rechenintensive Durchmustern hochdimensionaler Parameterräume. Für jeden untersuchten Punkt in diesen Räumen müssen umfangreiche Simulationen ausgeführt und analysiert werden. Das Datenvolumen dieser Simulationen wird das der eigentlichen Satellitenmessungen um Größenordnungen übersteigen.

Die jeweils neuen Daten des Tages werden, zusammen mit den bereits akkumulierten, mehrere komplexe Workflows durchlaufen. Astronomen sprechen daher bildlich von Datenanalyse-Pipelines. Da man bei einem Experiment wie Planck nie im Vorhinein genau weiß, was einen erwartet, sind diese Pipelines keine statischen Gebilde, sondern werden während der Missionsphase ständig modifiziert und verbessert. Und zwar von Wissenschaftlern, die an verschiedenen Orten arbeiten und meist individualistische Arbeitsstile pflegen. Damit die Datenverarbeitungszentren dabei nicht die Übersicht verlieren, wurde zu Beginn der Missionsentwicklung beschlossen IDIS, ein integriertes Daten- und Informationssystem, zu entwickeln. Herzstück von IDIS ist die ProC-Workflowengine, der Prozesskoordinator, und die daran angeschlossene Datenmanagementkomponente, kurz DMC.

Softwareanforderungen

Die einzelnen Programme der Workflows (oder Pipelines) müssen von Experten für eine Vielzahl von Fragestellungen entwickelt und gewartet werden. Jeder dieser Wissenschaftler hat seine Lieblingsprogrammiersprache, sei es C, C++, IDL, Mathematica, oder oft noch immer FORTRAN, in

der, und nur in der, sein Code für ihn pflegbar ist. Wie kann man diese Codes zusammenbinden, den Wissenschaftlern in ihrer Programmiergestaltung dabei möglichst wenig Einschränkungen auferlegen, aber dennoch ein System haben, in dem jegliche Berechnung auch in Zukunft nachvollziehbar ist, und welches jede der Aktionen der vielen Akteure dokumentiert und offensichtliche Fehler möglichst verbietet? Die Entwicklung des ProCs, einer projekteigenen, datenbankgestützten, multilingualen, wissenschaftlichen Workflowengine, die in möglichst vielen Umgebungen eingesetzt werden kann, war die Antwort des Planck-Projektes auf diese Herausforderung. ProC und DMC wurden allerdings so generisch konzipiert, dass sie in Zukunft auch in anderen Projekten ihre Dienste tun können. Maximale Flexibilität war die oberste Priorität im Design. So sollten die verwendeten Datenstrukturen, die benutzten Datenbanksysteme und auch die Prozessscheduler möglichst frei konfigurierbar bzw. wählbar sein. Der ProC sollte auf dem Multi-Core-Laptop des Wissenschaftlers sowie in Computational-Grids die zur Verfügung stehenden Ressourcen möglichst vollständig ausnutzen können, denn die Wissenschaftler sind rechenhungrig und ungeduldig.

ProC-Workflow-Engine

Der ProC besteht aus mehreren Komponenten, welche die verschiedenen Stadien eines Workflows unterstützen. Da wäre zum einen der Editor, der zur Definition des Workflows (Bauen einer Pipeline) genutzt wird (Abb. 4). Der Editor ist ein grafisches Interface, welches dem User die zur Verfügung stehenden Module und Kontrollelemente auflistet und es dem User erlaubt, diese frei auf einem Canvas zu platzieren und zu verbinden. Module sind im Kontext des ProCs eigenständige Programme, die z. B. Karten aus den Rohdaten berechnen oder eine Karte in ein Leistungsspektrum umwandelt. Der gesamte Workflow ist datenflussgesteuert, d. h. dass Beziehungen zwischen Modulen durch die Datenprodukte hergestellt werden. Ein Modul nimmt also das Datenprodukt eines anderen Moduls entgegen und verarbeitet es weiter. Der Editor überprüft mithilfe der Datenbank, ob der Output und der Input zweier zu verbindender Module kompatibel ist. Eine XML-Datei beschreibt die In- und Outputs eines Moduls sowie alle Parameter, die es erwartet. Um die Workflows möglichst modular zu halten, wurde eine Unterscheidung zwischen logischem Aufbau und Parametrisierung der Module eingeführt. Dies erleichtert die Wiederverwertung

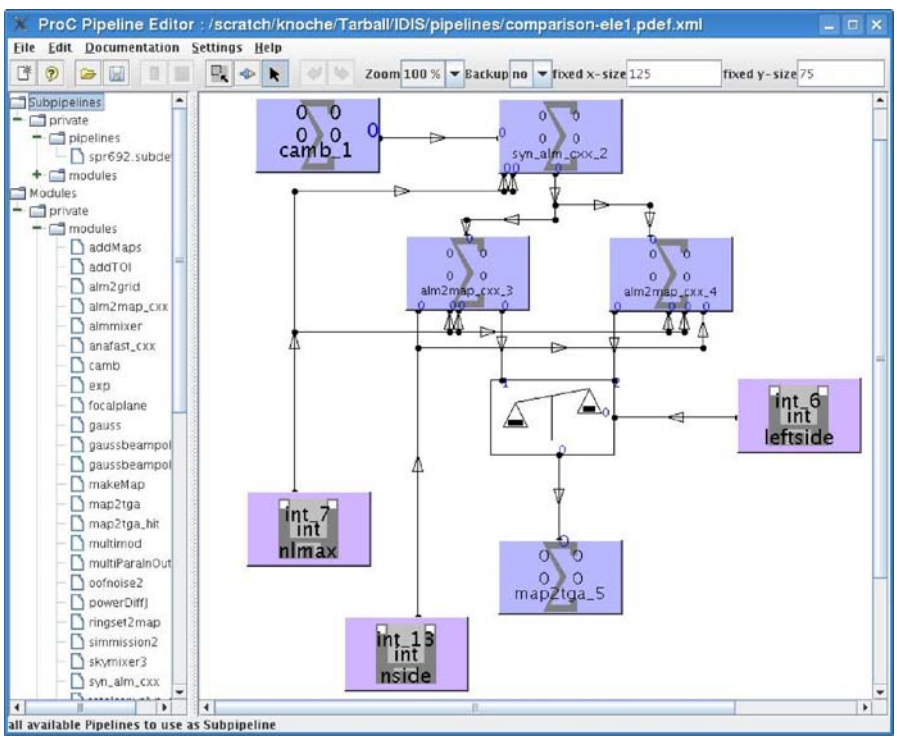

Abb. 4 ProC-Editor (links: Liste der verfügbaren Subpipelines/Module; rechts: Canvas zum Zeichnen der Pipeline)

von bereits erstellten Workflows mit geänderten Parametern.

Die zentrale Komponente des ProCs, der Pipeline Koordinator (PiCo), ist als Forward-Chaining-Engine für Ausführung und Ablaufsteuerung der Workflows zuständig, Backward-Chaining als Alternative ist in Vorbereitung. Der PiCo stellt fest, für welche Module alle notwendigen Inputs vorhanden sind und startet dann deren Ausführung, wann immer möglich parallel. Alle Module werden somit so früh wie möglich gestartet.

Eine Resume-Funktion ermöglicht die Wiederverwertung von bereits erstellten Datenprodukten. Mithilfe des DMCs wird erkannt, ob ein Modul mit exakt gleichen Inputs und Parametern schon ausgeführt wurde und die Ergebnisse in der Datenbank vorhanden sind. In diesem Fall wird das Datenobjekt direkt ohne Neuberechnung übernommen. Problematisch ist hierbei, dass verschiedene Versionen der Module existieren und kontinuierlich entstehen, ohne dass eine verlässliche Versionierung garantiert ist. Daher bietet der ProC vier verschieden abgestufte Modi, welche das „Vertrauen" in die bereits existierenden Daten widerspiegeln. Abgesehen davon, dass Resume gänzlich abgeschaltet werden kann, können entweder nur die durch denselben User produzierten Daten für Resume benutzt werden oder aber alle in der Datenbank verfügbaren Daten. In einem weiteren Modus überprüft der ProC anhand der MD5-Summe des Modul-Binaries, ob ein potenzielles Datenobjekt mit dem exakt identischen Modul erzeugt wurde.

Der ProC protokolliert die Ausführung der Workflows auf verschiedenen Ebenen. Es werden für jeden Pipelinelauf sowie für jedes Modul Statusobjekte erzeugt, die die Details der Datenprozessierung in einer History zusammenfassen: Zeitmarken, Benutzer-IDs, Prozessierungsstatus, Referenzen zu den erzeugten Datenobjekten, Checksummen der Pipelines, Binaries, Modulbeschreibungen, u. v. m. Des Weiteren wird das Logging der einzelnen IDIS-Komponenten (DMC, Module und weitere Komponenten) gesammelt und im ProC zentral ausgegeben. Der PiCo kann von der Kommandozeile aus gestartet werden, es existiert jedoch noch eine weitere Komponente im ProC, der SessionManager (SeMa, Abb. 5), welcher als grafisches Frontend eine bequeme Bedienung des ProCs erlaubt. Mit dieser GUI kann der User Pipelines laden, die Ausführung kontrollieren, den Editor aufrufen und das Logging anschauen. Zusätzlich kann die Konfiguration des ProC selbst (Schedulingsystem, Datenbank, Resume, Logging, etc.) hier vorgenommen werden.

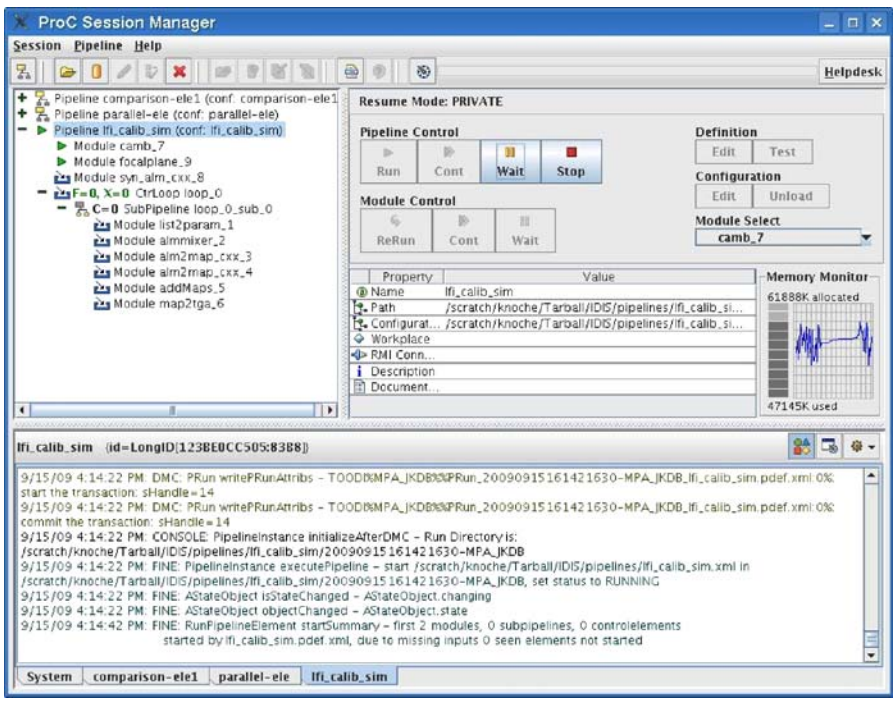

Abb. 5 ProC-SessionManager (links: Liste der geladenen Pipelines; rechts: Steuerelemente zum Starten, Stoppen, Konfigurieren der Pipelines; unten: Logging aller beteiligten Komponenten)

Modularität

Da beim Entwurf des ProC die späteren Anforderungen noch nicht absehbar waren und das Projekt außerdem auch für andere Anwendungen nutzbar sein sollte, galt bei der Entwicklung Flexibilität als oberstes Designprinzip. Wo immer möglich und sinnvoll, wurden generische Schnittstellen implementiert, was die Komponenten unabhängig bzw. austauschbar macht. Das fängt beim Editor an, der leicht durch einen anderen zu ersetzen ist, geht über generische Jobverteilung an Scheduling- bzw. Gridsysteme, bis hin zu frei wählbaren Datenbankbackends.

Auch der Wissenschaftler als Anwender des ProC wird dazu ermuntert modular zu arbeiten. Dazu können einerseits Pipelines zu Subpipelines kombiniert und in beliebiger Verschachtelungstiefe verwendet werden.

Andererseits werden dem Wissenschaftler Kontrollelemente angeboten, um Pipelines zu strukturieren, ohne diese Kontrollstrukturen selbst innerhalb der Module implementieren zu müssen.

Kontrollelemente

Die Kontrollelemente umfassen einfache Ablaufsteuerungen wie Schleifen und If-Then-Else-Strukturen, aber auch Elemente zur einfachen Parallelisierung diskreter Subworkflows bis hin zu komplexen Samplerelementen, die es erlauben beliebige Parameterräume mit diversen, auch selbst implementierbaren Optimierungs- bzw. Sampling-Algorithmen zu durchsuchen.

Das Comparison-Element (If-Then-Else) kann u. a. Parameter vergleichen und dementsprechend den Workflow in einen von zwei möglichen Ästen leiten. Das Loop-Element stellt eine einfache Möglichkeit dar, Subpipelines wiederholt und auch parallel auszuführen. Im ersten Fall ist es möglich Datenobjekte zirkulieren zu lassen (Output der Schleife als Input wiederverwenden), während bei der parallelen Abarbeitung die Zahl der gleichzeitig gestarteten Subpipelines konfiguriert wird. Dies ist insbesondere interessant, wenn der ProC die Jobs in ein Gridsystem submittiert.

Wie oben erwähnt, werden während der wissenschaftlichen Analyse der Daten viele Simulationen durchgeführt um hochdimensionale Parameterräume zu durchforsten. Hierbei unterstützt das Samplerelement die Wissenschaftler, die Parameter einer Subpipeline innerhalb vorzugebender Intervalle zu sampeln. Es werden verschiedene (erweiterbare) Optimierungs- bzw. Sampling-Algorithmen (Markov-Chain Monte Carlo, Sparse Grid Sampling, Simplex-Optimierung etc.) angeboten. Nach jedem Durchlauf der Subpipeline werden ein oder mehrere neue Parametersätze bestimmt, die dann sequenziell und/oder parallel vom Samplerelement abgearbeitet werden. Liefert die Subpipeline den Wert einer Zielfunktion, wird diese von geeigneten Samplingalgorithmen (z. B. MCMC) für die Entscheidung über die nächsten Samplingpunkte im Parameterraum benutzt. Viele der Algorithmen erzeugen große Generationen von Parametersätzen, die die Ressourcen von Clustern oder Gridsystemen ideal nutzen.

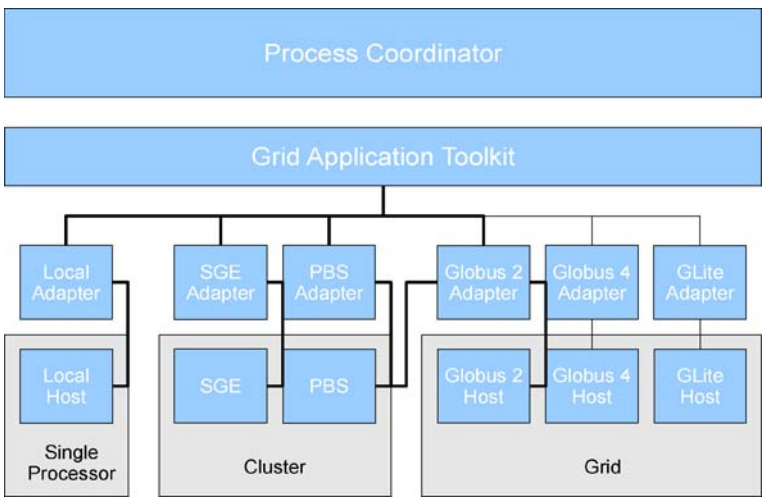

Abb. 6 Aufbau der Schnittstelle zum Scheduling

Scheduling

Der ProC submittiert die ausführbaren Module mithilfe des Grid-Application-Toolkits (GAT), einer generischen Schnittstelle zu verschiedenen Scheduling-Systemen (Abb. 6), über die die eigentliche Jobverteilung erfolgt. Dafür gibt es leicht zu entwickelnde Adaptoren, z. B. existieren für die Ausführung auf dem eigenen Rechner ein „lokaler Adaptor“ und für Clustersysteme Adaptoren zum Portable Batch System (PBS) und der Sun Grid Engine (SGE).

Weiterhin ist eine Gridanbindung möglich über das Globus Toolkit, sowie Glite. Für Unicore ist ein Adapter in Arbeit. Dies ermöglicht es eine Pipeline auf einem Laptop zu entwickeln und zu testen, um sie dann einfach im Produktionslauf auf den Cluster im Rechenzentrum oder auf das Grid zu verteilen.

Datenbank

Das DMC ist die Datenbank-Schnittstelle für den ProC sowie alle Programme (Module), welche innerhalb der Planck-Datenanalyse verwendet werden. Es muss die Reproduzierbarkeit aller in der Datenbank vorhandenen Objekte gewährleisten. Dies ist eine Herausforderung, da z. B. die Resume-Funktionalität des ProCs Abhängigkeiten schafft, welche bei rekursivem Löschen beachtet werden müssen. Für das DMC wurde auch Java als Programmiersprache gewählt um eine möglichst große Flexibilität und Modularität zu gewährleisten.

Das DMC bietet Schnittstellen als Java-Interface sowie für Module in C, C++, Fortran, und IDL als Bibliothek an. Das Java-Interface wird vom ProC genutzt, während Letztere vorwiegend von den Modulen der Wissenschaftler angesprochen werden. Diese greifen mittels JNI auf die Java-Routinen zu. Mithilfe von JDBC ist das DMC schnell an andere Datenbanken angebunden, was innerhalb der Planck-Mission wichtig ist, da die Wissenschaftler nicht nur mit einer einzigen Datenbank arbeiten.

Die Operations-Datenbanken sind in die oben erwähnten „Level“ unterteilt, d. h. es existieren separate Datenbanken, welche die Rohdaten des Satelliten, die produzierten Karten sowie alle Zwischenprodukte auf dem Weg zur finalen Himmelskarte beinhalten. Die Level-1-Datenbank ist mit Ausnahme der dateninjizierenden Programme nur für Lesezugriffe freigegeben. Dies soll die Rohdaten vor Manipulationen schützen. Außerdem werden später, wenn die Daten freigegeben werden, viele Wissenschaftler die Daten in eine lokale Datenbank auf ihrem eigenen PC speichern. Damit die Wissenschaftler dabei nicht an denselben Datenbanktyp wie im Operationsbetrieb gebunden sind und auch freie DBMS einsetzen können, wurde JDBC gewählt. Als Persistenzschicht fiel die Wahl auf JDO und hier speziell auf die zum Entscheidungszeitpunkt viel genutzte Kodo-Implementierung. Kodo übernimmt alle Aufgaben vom Bytecode-Enhancing der persistenten Klassen bis hin zur automatischen Erzeugung des Datenbank-Mappings gemäß den definierten Datenobjekten. Da Kodo allerdings keine freie Software ist, ist für den Zeitpunkt der Softwareveröffentlichung geplant, das DMC auf eine freie Persistenzschicht (Hibernate, OpenJPA, o. Ä.) umzustellen. Diese Umstellung würde jedoch nicht mehr in Datenverarbeitungszentren eingesetzt, da die Sicherheit der Daten oberste Priorität hat und die Umstellung des Persistenz-Managers mit viel Sorgfalt im Hinblick auf das Datenbankmapping, welches mit Beginn der Satellitenmission nur noch kompatible Änderungen erfahren darf, erfolgen müsste.

Innerhalb des Planck-Projekts existieren viele hundert persistente komplexe Datentypen, welche mittels einer abstrakten XML-Beschreibung definiert werden. Diese Datentypen umfassen u. A. die Definition der Rohdaten, der Himmelskarten oder der Leistungsspektren. Die XML-Beschreibung wird von einem Parser in die entsprechenden Java-Klassen, inklusive der zugehörigen Metainformationen für Kodo, überführt. Mithilfe dieser Metainformationen wird dann von Kodo automatisch das Mapping erstellt. Der Wissenschaftler braucht sich also bei der Beschreibung der Datentypen keine Gedanken um datenbankspezifische Gegebenheiten wie Abbildung der einfachen Datentypen oder Indizes zu machen. Die Definition der komplexen Datentypen folgt dem objektorientierten Ansatz. So ist es möglich, dass Datentypen voneinander abgeleitet werden und somit ein logischer Zusammenhang zwischen diesen erstellt wird. Gleichzeitig verringert sich der Beschreibungsaufwand, da nur noch die „neuen“ Attribute eines Typs definiert werden müssen. Innerhalb der Planck-Mission existieren drei selbstdefinierte Grunddatentypen, welche den jeweiligen Einsatzzweck definieren. Neue Typen können von diesen abgeleitet werden und sind dann au-

tomatisch fähig in die Datenbank persistiert zu werden, Metadaten oder aber richtige Datenspalten zu enthalten. Die Datenspalten können die grundlegenden Datentypen (float, double, int, etc.) enthalten. Bei den Metadaten wird noch zwischen obligatorisch und nichtobligatorisch unterschieden. Letztere können zur Laufzeit frei definiert werden.

GUI

Die GUI des DMC (vgl. Abb. 7) ist primär darauf ausgelegt nach Objekten in der Datenbank zu suchen und einen schnellen Überblick über die Inhalte dieser Objekte zu gewinnen. Sie kann jedoch auch für detailliertere Informationen wie die Erzeugungshistorie eines Objekts (Abb. 8) oder für eine grafische Darstellung der wissenschaftlichen Daten genutzt werden. Die Historie eines Datenprodukts ist ein zentraler Punkt, da die Erfahrung früherer Satellitenmissionen gezeigt hat, dass spontane Kreativität der Wissenschaftler selten mit guter Verfolgbarkeit der einzelnen Analyseschritte einhergeht. Das DMC wertet die Metainformationen, welche vom ProC geschrieben werden, aus und man kann bei jeglichen Datenprodukten auf einfache Weise nachverfolgen, aus welchen Daten, mit welchen Programmen, in welcher Pipeline und mit welchen Parametern diese Objekte erzeugt wurden. Gleichzeitig ergibt dies

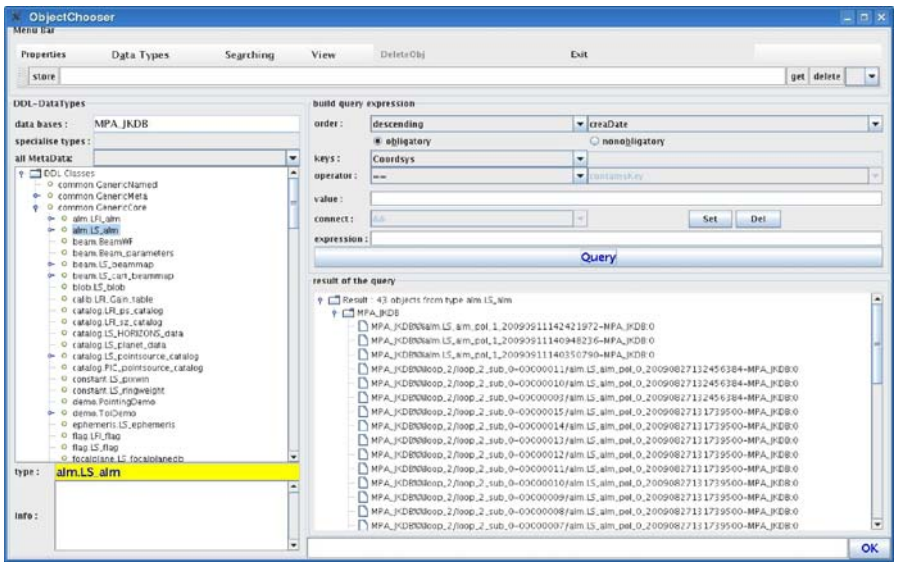

Abb. 7 DMC GUI (links: Hierarchische Liste aller definierten Datentypen; rechts: Ergebnis eines Querys)

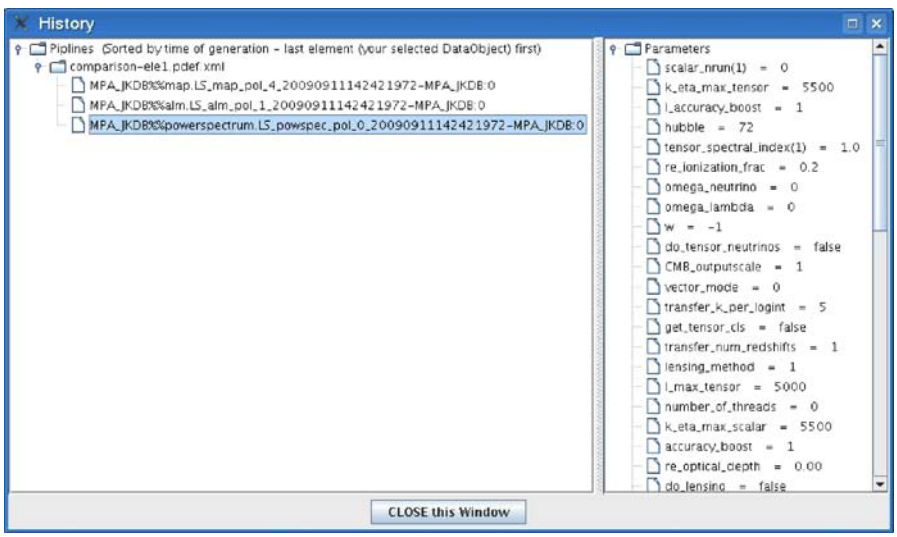

Abb. 8 Historie eines Datenobjekts (links: Liste der zur Erzeugung benutzten Datenobjekte; rechts: benutzte Parameter des erstellenden Programmes)

die Möglichkeit Daten, die bspw. durch fehlerhafte Programme erzeugt wurden, und alle daraus abgeleiteten Daten mit einem Klick aus der Datenbank wieder zu entfernen. Außerdem bietet die GUI einfache Plotting-Möglichkeiten um „mal eben schnell" die Plausibilität der Daten zu überprüfen, ohne dabei bestehende, spezialisierte Visualisierungssoftware zu ersetzen. Die zentrale Aufgabe der DMC-GUI ist aber das Auffinden von Daten. Da beliebige Datenbanken an das DMC angebunden werden können, und viele von diesen einen eigenen SQL-Dialekt mit sich bringen, wird zur Beschreibung eines Querys die durch JDO/Kodo definierte JDOQL benutzt. Diese Querysprache ist, wie die Datentypen auch, objektorientiert, sodass recht einfach nach den Werten eines bestimmten Attributs eines Datentyps gesucht werden kann.

Informatik und Kosmologie – Rückblick und Ausblick

Die Vorbereitungen der Planck-Satellitenmission und ihrer Datenauswertung starteten von mehr als zehn Jahren. Bei ProC und DMC wurden daher die Softwareanforderungen zwar recht früh definiert aber auch sehr allgemein gehalten. Dies hat die Entwicklung etwas erschwert. Die Wünsche der prospektiven Softwarenutzer waren nicht immer präzise, da diese als Wissenschaftler kaum zuvor mit einem Datenverarbeitungsproblem dieser Komplexität konfrontiert gewesen waren. Die Softwareentwickler mussten die möglichen Arbeitsabläufe der Datenverarbeitungszentren erahnen, um daraus selber auf die essenziellen Anforderungen zu schließen. Auch wurden manche Entscheidungen über die verwendeten Softwarekomponenten teilweise von der Zeit überholt und müssen nachträglich revidiert werden. Dennoch hatte die langfristige Softwareplanung den Vorteil, dass eine kleine Gruppe von Entwicklern die Software sorgfältig planen und ohne großen Zeitdruck implementieren konnte. Wie diese Verbindung von Informatik und Kosmologie sich weiter entwickelt muss beobachtet werden.

Ein System, welches die wesentlichen Anforderungen für die nachvollziehbare Datenanalyse kosmologischer Experimente durch eine größere Gruppe von Wissenschaftlern erlaubt, wurde nun entwickelt. Aber wie wird es genutzt werden? Die ersten Daten des Planck-Satelliten durchlaufen bereits heute ProC-Pipelines, die von einer kleinen Zahl von Experten, mehr Informatikern als Kosmologen, erstellt wurden. Aber auf die Endprodukte der verschiedenen Level der zentralen Planck-Datenverarbeitung werden sich Hunderte von Wissenschaftlern stürzen. Für diese ist eine dank Vorkonfiguration leicht zu installierende Variante des Systems, *IDIS in a box*, in Vorbereitung. *IDIS in a box* soll neben dem ProC, dem DMC auch die Planck-Simulationspipeline, sowie eine Reihe von bereits existierenden Analysemodulen enthalten. Eine wesentliche Komponente in dem soziologischen Prozess, die große Zahl von Planck-Wissenschaftlern an diese Technologie heranzuführen, werden Beispiele von wissenschaftlichen Anwendungen sein, die die Stärken des ProC und des DMCs herausstellen, wie die Bestimmung grundlegender kosmologischer Eigenschaften mittels des Samplingelements. *IDIS in a box* wird auch die Vorbereitung des öffentlichen Releases der Software sein, die von der Entwicklergruppe am Max-Planck-Institut für Astrophysik in Garching sowie vom Deutschen Zentrum für Luft- und Raumfahrt (DLR), durch welches das Projekt finanziert[1] wird, angestrebt ist. Ob Informatik und Kosmologie in stärkere Wechselwirkung treten ist eigentlich nicht die Frage, sondern nur mit welchem Tempo.

[1] Aus Mitteln des Bundesministeriums für Wirtschaft und Technologie unter dem Förderkennzeichen 50 OP 0901