

Meilensteinbericht

D-Grid HEP-CG

Arbeitspaket 3: Verteilte Datenanalyse unter Verwendung von Grid Ressourcen

Johannes Elmsheuser, Günter Duckeck
Ludwig-Maximilians-Universität München

1 Einleitung

Die verteilte Datenanalyse unter Verwendung von Grid Ressourcen ist eine der wichtigsten Anwendungen der experimentellen Hochenergiephysik, die in den nächsten Jahren zur Praxisreife entwickelt werden muss [1]. Eine effektive Analyseumgebung und das Know-how diese zu nutzen und weiterzuentwickeln, sind für die Community unabdingbar, um wissenschaftlich von den hohen Investitionen in Beschleuniger und Detektoren zu profitieren.

Die Anforderungen an das Management der Ressourcen sind dabei sehr hoch. In jedem Experiment werden mehrere 100 bis 1000 Physiker Jobs in das Grid submittieren. Damit dies auch für Anwender ohne weit reichende Grid-Expertise nutzbar ist, sind geeignete Benutzerschnittstellen und Hilfsprogramme erforderlich. Diese erweitern die Nutzung des Grids von wenigen Produktionsmanagern auf potentiell alle beteiligten Physiker der Experimente.

2 Gap-Analyse: Automatischer Job-Manager

Eine ausführliche Gap-Analyse zum Bereich *Automatischer Job-Manager* im Rahmen des ATLAS-Experiments [2] und des LCG-Grids [3] wird im Folgenden dargestellt. Zunächst wurde eine grundlegende Liste mit Spezifikationen

erarbeitet, die ein Automatischer Job-Manager erfüllen sollte:

Interface zur Job-Konfiguration: Ein einheitliches Interface zur Konfiguration der Parameter der verwendeten Programme, z.B.: Softwareversion, Konfiguration des Programmkomponenten, Datensets, u.v.m.. Dieses Interface muss einerseits auf die spezifischen Bedürfnisse der Anwendungsprogramme abgestimmt sein, andererseits aber auch flexibel auf unterschiedliche Anwendungen anzupassen sein.

Job-Submission Interface für Grid/Batch Systeme: Die Analyseprogramme sollen auf unterschiedliche lokale bzw. entfernte Computersysteme geschickt werden können. Hierbei sollten sowohl verschiedene Batch- bzw. Grid-Systeme, als auch experiment-spezifische Produktionssysteme berücksichtigt werden. Job-splitting und paralleles Abschicken (sog. bulk-submission) der Jobs muss unterstützt werden.

Integration mit Datenmanagement: Die Analyseprogramme greifen auf ein experiment-spezifisches Datenmanagement-System zu. Mit dessen Hilfe werden die Analysejobs im Grid zum Ort der Datenspeicherung gesendet, um das Kopieren großer Datenmengen zu vermeiden. Außerdem ist bei Job-splitting eine Aufteilung des Datensets und Zuordnung an die jeweiligen Sub-Jobs erforderlich.

Ressource estimation: Es wird ermittelt wieviel Ressourcen, z.B. Speicherbedarf oder CPU-Zeit, ein Job vorraussichtlich benötigen wird. Gleichzeitig werden die im Grid verfügbaren Ressourcen geschätzt.

Job-Monitoring: Während des Ausführens der Analyseprogramme informiert ein kontinuierliches Monitoring über den Status der Jobs. Es wird ermittelt, an welches entfernte Computersystem im Grid das Programm gesendet wurde, ob es sich im Warte- bzw. Ausführungszustand befindet oder ob es beendet wurde.

Job-Error Checking: Es wird überprüft, ob ein Job erfolgreich beendet wurde und bei evt. aufgetretenden Fehlern Informationen gesammelt bzw. kategorisiert. Es besteht die Möglichkeit, fehlerhaft beendete Programme automatisch wieder zu verschicken.

Collecting and Merging of the Results: Nach erfolgreicher Beendigung des Programms sollten die Programm- und Fehler-Ausgaben automatisch auf den Ausgangscomputer übertragen werden. Gleichzeitig existiert bei paralleler Ausführung mehrerer Programme eine Möglichkeit Ergebnisse zusammenzufügen.

Job-Archiv: Es existiert ein sog. Job-Archiv, mit dessen Hilfe bereits beendete Programme überprüft bzw. als Vorlage für neue Programme verwendet werden können. Wünschenswert wäre ausserdem ein Interface zu einem Meta-Data Storage System.

Bulk Operations: Viele der aufgezählten Funktionen werden im sogenannten bulk Modus ausgeführt. Werden Jobs gestartet, auf ihren Zustand überprüft oder die Resultate gespeichert, geschehen diese Aktionen immer parallel bzw. im Hintergrund, so daß die Wartezeit des Benutzers minimiert wird.

Als Job und Scheduling-Manager wird das gemeinsam von den LHCb- und ATLAS-Experimenten entwickelte Programm GANGA [4] verwendet. Dieses Programm bietet eine einheitliche Umgebung zur Konfiguration verschiedener experimentspezifischer Analyseprogramme (“Athena“ im Fall des ATLAS-Experiments) oder generischer Programme zum Start auf lokalen Batch-Systemen oder verschiedenen Grid-Typen. Es kann sowohl per Kommandozeile, Skripten oder einer grafischen Benutzeroberfläche bedient werden. Die Grundfunktionalität der oben erwähnten Eigenschaften sind in einigen Punkten schon vorhanden:

- In GANGA können die Startparameter generischer Programme bzw. Athena interaktiv, mit einer grafischen Benutzeroberfläche oder mit Python-Skripten konfiguriert werden. Die auszuführenden Programme können u.a. auf folgende Systeme gesendet werden: lokales Computersystem, LSF- bzw. PBS-Batch-Systeme, LCG- und gLite-Grid-Umgebungen. Weitere denkbare zukünftige Umgebungen sind das ATLAS-Produktionssystem bzw. andere Grid-Typen.
- Das Datenmanagement war zum Zeitpunkt des Projektstarts nur auf einfachem Niveau implementiert: die zu prozessierenden Daten werden entweder mit dem auszuführenden Programm verschickt oder es kann auf Daten auf einem entfernten Computer (SE) zugegriffen werden. Bei letzterem wird das Analyse-Programm aufgrund der LCG-Grid Funktionalität zu demjenigen LCG Computing System (CE) geschickt, das den Daten am nächsten liegt. Es existiert nun ein Interface zum neuen Daten-Management-System von ATLAS. Die Möglichkeit der Parallelisierung der Analyseprogramme wurde während der ersten Projektmonate implementiert.
- Das Job-Monitoring ist in Grundzügen implementiert. Detailliertere Informationen zum Status wartender bzw. laufender Jobs wären sinnvoll und eine Erweiterung des Monitoring via Web Interface wünschenswert.

- Error checking existiert nur auf Ebene der Grid- bzw. Batch-System Kommunikation. Ein Erfassen bzw. Kategorisieren von Problemen und Fehlern bei der Ausführung der Jobs fehlt. Es existiert kein automatischer Neustart-Mechanismus für fehlgeschlagene Jobs.
- Nach Beendigung der Analyseprogramme werden Programm- und Fehlermeldungen automatisch gespeichert. Bei paralleler Ausführung existiert nun ein einfacher Mechanismus zum Zusammenfügen der Ergebnisse, der während des Projektstarts noch nicht existierte.
- Ein Job-Archiv existiert in einfacher Form.

Die Münchner Gruppe hat die Funktionalität GANGAs erfolgreich in verschiedenen Bereichen erweitert. Zu den wichtigsten Komponenten zählt hierbei die Job Parallelisierung und die Integration des ATLAS Datenmanagementsystems DQ2/DDM mit dem direkten Zugriff auf die Dateien des Eingabedatensatzes. Diese Komponenten sind von besonderer Bedeutung, denn nur durch eine intelligente Parallelisierung und ein robustes Datenmanagement kann ein erfolgreiches System zur verteilten Analyse aufgebaut werden. Eine genauere Beschreibung dieser Erweiterungen wird im Folgenden gegeben:

- Es wurden Funktionen zur Job Parallelisierung und Job Splitting für ATLAS Analyse Jobs in GANGA implementiert. Dabei wird ein Eingabedatensatz auf mehrere Jobs verteilt, die gleichzeitig parallel abgearbeitet werden. Dies beschleunigt die Prozessierung großer Datenmenge erheblich und ist ein wichtiger Bestandteil der verteilten Analysefunktionalität.
- Der Zugriff auf das neue ATLAS Datenmanagementsystem DQ2/DDM in GANGA und innerhalb der jeweiligen Grid Jobs wurde integriert. Damit ist es dem Benutzer möglich, nur unter Angabe eines Datensatznamens die zugehörigen Dateien vollständig auf dem Grid mit automatischer Aufteilung auf mehrere parallele Jobs prozessieren zu lassen.
- Ein wesentlicher Bestandteil der Integration des ATLAS Datenmanagement Systems DQ2/DDM ist eine Domain-Erkennung für LCG Jobs.
- Für LCG Grid Jobs wurde der direkte Zugriff (Posix I/O) auf Eingabedatendateien, die auf einem nahen dCache- oder Castor-SE liegen, mittels POOL/ROOT ermöglicht. Damit können also sehr große Datenmengen prozessiert werden, ohne dass diese zuvor heruntergeladen werden müssen.

Weitere GANGA Erweiterungen sind:

- Mit Hilfe eines weiteren neuen Plugins ist es möglich die Skripte und Transformationen des offiziellen ATLAS Monte Carlo (MC) Produktionssystem in GANGA einfach zu konfigurieren und für eine 'kleinere' MC Produktion im Grid zu verwenden.
- Grid Jobs können mit Condor-G Interface gestartet werden. Dies bietet den Vorteil einer sehr effizienten Bulk-Job-Submission. Damit ist es möglich, beliebige Programme wie auch ATLAS-Athena Analyse Jobs mittels CondorG im LCG Grid oder auch in anderen Grids auszuführen. Die Job Submission Zeit wird durch CondorG erheblich verkürzt. Zum Beispiel verringert sich die Submission Zeit für 100 Jobs von etwa 30 Minuten auf wenige Sekunden.
- Es wurde ein weiteres Eingabe-Datensatz-Plugin für Dateien auf lokalen Dateisystemen integriert. Dies ermöglicht das einfache Testen von Programmen auf einem lokalen Computer bevor diese ins Grid abgeschickt werden.
- Eine verbesserte Verwaltung der Ausgabedatensätze im GANGA Job Repository wurde eingeführt. Dabei werden Jobs nach ihrer Beendigung automatisch überprüft und die Speicherorte der Ausgabedateien registriert.
- Eine Erweiterung für das Zusammenfügen von Ausgabe-Dateien aus mehreren Parallel-Jobs wie z.B. ROOT-Dateien bzw. Text-Log-Dateien wurde integriert. Ausgabedateien auf entfernten Dateisystemen können automatisch im Hintergrund heruntergeladen werden und auf dem lokalen Dateisystem zur Weiterverarbeitung gespeichert werden.
- Eine verbesserter Mechanismus zum Abfangen von Fehlern während des Ausführens von ATLAS Analyse Jobs im Grid wurde implementiert. Der sog. Fehlerwert wird an GANGA zurückgeliefert und im Job Repository gespeichert.
- Zahlreiche Tests dieser neuen Funktionalitäten in GANGA wurden lokal in München und im LCG Grid in Karlsruhe, Lyon und CERN durchgeführt. Einige ATLAS DQ2 Datensätze wurden nach Karlsruhe repliziert und die Funktionlität des DQ2/DDM Systems getestet.

Erste Ergebnisse der Gap-Analyse über Distributed Analysis und Erfahrungen mit dem Job Scheduler GANGA wurden im Rahmen einer Posterpräsentation und Contributed Paper auf der CHEP 2006 Konferenz in Mumbai,

Indien vorgestellt [5]. Eine ausführliche Beschreibung der neuen Funktionalitäten in GANGA wurden in den ATLAS-Computing-Wiki [6] aufgenommen und mit zahlreichen Beispielen beschrieben. Im Rahmen mehrerer Tutorials während der ATLAS Trigger and Physics Wochen und an der LMU München wurden diese vorgestellt.

Die meisten aufgezählten Punkte der Gap-Analyse, die ein automatischer Job Manager erfüllen sollte, wurden in GANGA verbessert. So bieten die folgende Punkte die notwendige Basisfunktionalität, müssen aber noch in Details verbessert, besser integriert und robuster gemacht werden:

- Interface zur Job Konfiguration
- Job Submission Interface für Grid/Batch Systeme mit Parallel-Jobs
- Integration mit dem Datenmanagementsystem
- Collecting and Merging of the Results
- Job Archiv

Zunächst soll aber eine intensive Testphase unter Einbeziehung einer größeren User Community stattfinden. Die folgenden Punkte sind nur teilweise vorhanden bzw. bedürfen noch einer größeren Verbesserung und Erweiterung:

- Resource estimation
- Job Monitoring
- Job Error Checking

3 Gap-Analyse: Interaktive Analyse

Die Gap-Analyse zur *Interaktiven Analyse* wird in Zusammenarbeit mit dem Projektpartner GSI Darmstadt erarbeitet.

4 Zusammenfassung

Es wurde eine ausführliche Gap-Analyse zum Thema des Automatischen Job-Managers dargestellt. Dabei wurden die verschiedenen gewünschten Funktionalitäten im Rahmen des computing models des ATLAS-Experiments dargestellt und mit schon existierenden Programmen verglichen. Dabei hat sich der Job-Scheduler GANGA als sehr guter Kandidat erwiesen, so daß schon

eingeliste Punkte im Arbeitsplan der Projektmonate 12 bis 24 vorgezogen wurden. Hierbei wurde die Funktionalität GANGAs in wesentlichen Punkten der Gap-Analyse-Ergebnisse verbessert.

Literatur

- [1] D.Baberis *et. al.*, Common Use Cases for a HEP Common Application Layer for Analysis, LHC-SC2-2003-032
- [2] ATLAS Computing Wiki Seite:
<https://uimon.cern.ch/twiki/bin/view/Atlas/AtlasComputing>
- [3] LCG Projekt WWW-Seite:
<http://lcg.web.cern.ch/LCG/>
- [4] GANGA Projekt WWW-Seite:
<http://ganga.web.cern.ch/ganga/>
- [5] CHEP 2006 Konferenz WWW-Seite:
<http://www.tifr.res.in/~chep06/>
- [6] GANGA ATLAS Wiki-Seiten:
<https://twiki.cern.ch/twiki/bin/view/Atlas/DistributedAnalysisUsingGanga>
<https://twiki.cern.ch/twiki/bin/view/Atlas/GangaTutorial420>
<https://twiki.cern.ch/twiki/bin/view/Atlas/GangaUpdates420>
<https://twiki.cern.ch/twiki/bin/view/Atlas/GangaTutorial415>