Fraunhofer Institut
Rechnerarchitektur
und Softwaretechnik

Konzept für das Scheduling von Workflow-Aktivitäten in Instant-Grid

Technischer Bericht

Andreas Hoheisel
andreas.hoheisel@first.fraunhofer.de
Helge Rose
helge.rose@first.fraunhofer.de

Fraunhofer Institut für Rechnerarchitektur
und Softwaretechnik

Stand: 1. Juni 2006

1 Begriffe

- **Workflow:** Automatisierung von Prozessabfolgen. In Instant-Grid werden Workflows durch High-Level Petrinetze modelliert, deren Marken Daten repräsentieren. Die Transitionen der Petrinetze lösen beim Schalten Aktivitäten aus, wie zum Beispiel die Ausführung eines Programms oder eines Web-Service-Methodenaufrufes.
- **Grid Workflow Description Language (GWorkflowDL):** XML-basierte Workflowbeschreibungssprache, die zur Spezifikation von Workflows die Syntax und Semantik von High-Level-Petrinetzen verwendet (siehe http://www.gridworkflow.org/gworkflowdl/)
- **Grid Workflow Execution Service (GWES):** Web Service, der GWorkflowDL-Dokumente auswertet und auf entsprechender Grid-Middleware (z.B. Globus Toolkit 4 oder Web Service) die damit verbundenen Aktivitäten ausführt (siehe http://www.gridworkflow.org/gwes/).
- **Scheduling:** Erstellung eines Ablaufplanes (schedule), der Aktivitäten (zeitlich begrenzt) Ressourcen zuweist. Dabei soll in der Regel ein bestimmtes Kriterium optimiert werden, wie zum Beispiel Durchsatz (=Effizienz), Fairness, Termineinhaltung oder Verweilzeit.

2 Annahmen und Anforderungen

- Die Struktur des Workflows (z.B. zukünftige Aktivitäten) wird beim Scheduling nicht beachtet. Es werden für das Scheduling nur die Aktivitäten betrachtet, die unmittelbar vor der Ausführung stehen. Die Auswahl dieser jetzt zu startenden Aktivitäten erfolgt durch den GWES.
- Die Suche passender Ressourcen (Resource Matching) ist bereits erfolgt (Beispiel: Zu jeder Aktivität Auswahl von Rechnern mit passender Hardware- und Softwareaustattung).
- Es wird (zunächst) keine „advanced Reservation“ unterstützt
- Die Ausführungsdauer von Aktivitäten ist vor ihrer Ausführung nicht bekannt bzw. wird nicht beachtet.
- Co-allocation (garantiertes gleichzeitiges Ausführen) von Aktivitäten wird (zunächst) nicht unterstützt.
- Die Rechner werden nicht exklusiv vom GWES verwendet, sondern es können auch Prozesse von anderen auf den Rechnern ausgeführt werden.
- Es dürfen mehrere Aktivitäten gleichzeitig auf einem Rechner ausgeführt werden.

- Solange die Auslastung pro CPU des Rechners einen bestimmten Schwellwert überschreitet, sollen keine weiteren Aktivitäten auf dem Rechner gestartet werden.

3 Optimierungskriterium

Das Optimierungskriterium für das Workflow-Management in Instant-Grid ist eine möglichst vollständige Auslastung der CPUs der zur Verfügung stehenden Rechner (unabhängig von ihrer Leistungsfähigkeit). Es soll somit die Effizienz und damit auch der Durchsatz über das gesamte Grid optimiert werden und nicht etwa die Verweilzeit einzelner Aktivitäten bzw. Prozesse. Dabei ist zu beachten, dass Aktivitäten im Allgemeinen jeweils nur auf bestimmten Ressourcen (Rechnern) ausgeführt werden können, die von Aktivität zu Aktivität unterschiedlich sein können.

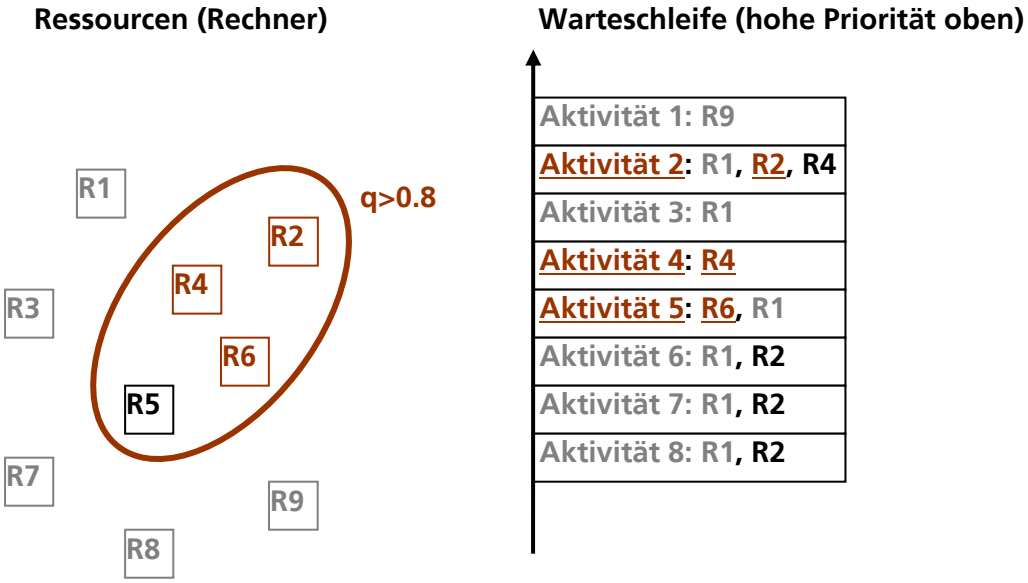
Zusätzlich sollen gesamte Workflows (oder auch einzelne Aktivitäten) mit Prioritäten versehen werden können, die zu einer bevorzugten Abarbeitung führen, falls die entsprechenden Ressourcen momentan ausgelastet sind (Vorrang in der Warteschlange).

4 Lösungsweg

1 Aktivierte, aber noch nicht in Ausführung befindliche Aktivitäten werden von dem GWES in eine Warteschlange gestellt, die gemäß der (nutzerspezifischen) Priorität sortiert ist. Jede der Aktivitäten in der Warteschlange ist mit einer Liste passender Ressourcen versehen.

2 Für alle beteiligten Ressourcen wird eine Qualitätszahl $q \in [0..1]$ bestimmt, die sich aus der Anzahl CPUs und der letzten bekannten Auslastung (load) berechnet. Diese Qualitätszahl ist zunächst (in der ersten Ausbaustufe) unabhängig von der Aktivität. In einer späteren Ausbaustufe könnte diese Qualitätszahl aber auch von der Aktivität abhängen, zum Beispiel wenn man den angeforderten mit dem jeweils aktuell zur Verfügung stehenden Arbeitsspeicher vergleicht.

3 Möglichst viele Ressourcen, deren Qualitätszahl einen bestimmten Schwellwert überschreitet (z.B. q>0.8), werden nun den Aktivitäten in der Warteschleife zugeordnet, beginnend bei Aktivitäten mit hoher Priorität. Pro Iteration wird jede Ressource höchstens einer Aktivität zugeordnet, obwohl mehrere Aktivitäten pro Ressource erlaubt sind. Es ist jedoch zunächst die Veränderung der Auslastung in Folge einer neuen Aktivität abzuwarten, bevor eine weitere Aktivität auf demselben Rechner gestartet wird.
4 Die Aktivitäten der Warteliste, die mit einer Ressource versehen wurden, werden nun aus der Warteliste entfernt und vom GWES gestartet.
5 Goto **1.**



Beispiel für Scheduling in Instant-Grid: Das Grid besteht aus den Rechnern R1 bis R9. Von diesen Rechnern haben vier Rechner (R2, R4, R5, R6) eine Qualitätszahl > 0.8 (d.h. eine Auslastung < 0.2 pro CPU). Es sollen 8 Aktivitäten gestartet werden. Aktivität 1 kann nur auf R9, Aktivität 2 nur auf R1 oder R2 oder R4 gestartet werden, etc. Aktivität 1 hat die höchste und Aktivität 8 die niedrigste Priorität. Der Scheduler legt nun fest, dass Aktivität 2 auf R2, Aktivität 4 auf R4 und Aktivität 5 auf R6 gestartet wird. Aktivität 1 kann nicht gestartet werden, da die einzig zulässige Ressource R9 noch eine zu hohe Auslastung vorweist, analog Aktivität 3. Die Aktivitäten 6 bis 8 werden in diesem Schritt nicht gestartet, weil R2 schon durch die höher priorisierte Aktivität 2 belegt ist. Zur freien Ressource R5 findet sich keine passende Aktivität.

5 Schnittestellen

Der Scheduler wird zunächst als Java-Bibliothek in den GWES eingebunden. Der Scheduler verwendet die D-GRDL-Datenbank und den Instant-Grid Data Dispatcher Dienst, um Informationen über die aktuelle Rechnerauslastung zu erhalten. Die Schnittstelle zwischen GWES und Scheduler besteht aus einem Java Interface mit folgender Methode:

```
ArrayList<Transition> schedule(ArrayList<Transition> transitions)
```

Der Eingabeparameter ist eine sortierte Liste, welche alle aktivierten Transitionen enthält, die jeweils mit einer Aktivität verknüpft sind, welche jetzt gestartet werden kann. Die Reihenfolge der Transitionen in der Liste ergibt sich über die Priorität (hohe Priorität zu erst). Der Rückgabewert ist wieder eine sortierte Liste, welche alle Transitionen enthält, für deren Aktivitäten eine Ressource ausgewählt werden konnte. Die Java-Klasse „Transition" wird durch das Paket `net.kwfgrid.gworkflowdl.structure` bereitgestellt. Die Klasse Transition entspricht dem XML-Element <transition> der GWorkflowDL.

```
<transition ID="cat1">
  <description>concatenate two files</description>
  <inputPlace placeID="d25" edgeExpression="input1"/>
  <inputPlace placeID="d26" edgeExpression="input2"/>
  <outputPlace placeID="d25-26" edgeExpression="stdout"/>
  <operation>
    <pe:programClassExecution
     xmlns:pe="http://www.gridworkflow.org/gworkflowdl/programclassexecution"
     softwareClass="http://fhrg.first.fraunhofer.de/fhrg/grdl/sw/cat.xml">
      <pe:programExecution software="/home/grid/fhrgbin/cat/cat.sh"
        hardware="https://countess.first.fhrg.fraunhofer.de:8443/wsrf/services/ManagedJobFactory
        Service" quality="0.85" selected="true"/>
      <pe:programExecution software="/home/grid/fhrgbin/cat/cat.sh"
        hardware="https://quadro.first.fhrg.fraunhofer.de:8443/wsrf/services/ManagedJobFactorySe
        rvice" quality="0.3"/>
    </pe:programClassExecution>
  </operation>
</transition>
```

XML-Beispiel einer Transition. Der Scheduler hat bei den beiden Elementen <programExecution> die Attribute „selected" und „quality" hinzugefügt.