

АРХИТЕКТУРА НА ГРИД РЕСУРСЕН БРОКЕР БАЗИРАН НА МОДЕЛ ЗА ПЛАНИРАНЕ С МЕТАДАННИ

Пламенка Боровска, Хараламбос Докомес

Абстракт. В статията се предлага архитектура на ГРИД ресурсен брокер, базиран на модел за планиране с метаданни. Предложен е сценарий за планирането, при който моделът за планиране обхваща 5 базови модела – на ресурсите, приложението, производителността, метрика за производителността, политика за планиране и програмен модел. Дефинирани са множествата метаданни на входа и на изхода на модела на планиране. Предложеният архитектурен проект на ГРИД ресурсен брокер включва три базови модула: картограф, оценител цена/ресурси и диспечер. Предложен е сценарий за взаимодействието на ресурсния брокер с услугите в ГРИД.

Ключови думи: ГРИД, ГРИД услуги, метаданни, модел за планиране на ресурсите, ресурсен брокер, ресурсен мениджмънт

ARCHITECTURAL DESIGN OF GRID RESOURCE BROKER BASED ON METADATA SCHEDULING MODEL

Plamenka Borovska, Charalampos Ntokomes

Abstract: In this paper a GRID resource broker architectural design has been suggested on the basis of metadata scheduling model. A scenario for scheduling has been built up comprising 5 fundamental modules – resources, application, performance, performance metrics, scheduling policy, and parallel programming model. The metadata sets of the input and the output of the scheduling model have been defined. The suggested architecture of the GRID resource broker includes 3 basic modules: a mapper, evaluator price/resources and dispatcher. A scenario for the interaction of the resource broker with the services in the GRID has been constructed.

Key words: GRID, GRID services, metadata, resource broker, resource management, scheduling model

1. ВЪВЕДЕНИЕ

Управлението и планирането на ГРИД ресурсите са важни компоненти при изграждането на ГРИД [2,8,11]. Тези компоненти са отговорни за селекцията и разпределянето на ГРИД ресурсите за приложенията, и следователно, осъществяват достъпа на потребителите до ресурсите [1,3,13].

Терминът ГРИД ресурсен мениджмънт – (Grid Resource Management - GRM) специфицира множеството функционалности, които са необходими за интегрирането на наличен пул ресурси в средата на Grid и осигуряването на средства за достъп до ресурсите [7,14].

В средата на Grid ресурсите често са хетерогенни по отношение на собствениците/ доставчиците, политиките за достъп, производителностите и цените, така че, системите за ресурсен мениджмънт в ГРИД трябва да вземат предвид тези особености и да осигуряват ефективна използваемост на ресурсите, както и оптимизация на производителността [5,9,10,12].

Много често е необходимо да се разполага с предварителна информация за наличните ресурси и изискваната производителност. Ефективен и популярен подход е да се използват *споразумения за ниво на обслужване (Service Level Agreements – SLA)* за осигуряване на изискваната висока надеждност [15].

2. МОДЕЛ НА ПЛАНИРАНЕТО НА РЕСУРСИТЕ

Важни изисквания за съвременните GRM системи са резервирането на ресурси и качеството на обслужването (QoS). Едно от най-важните качества на GRM системата е способността да осигурява ефективно автоматично разпределяне на ресурсите между задачите. Това се осъществява от планиращата програма, обхващаща множества от софтуерни компоненти или услуги, която на базата на знания или прогнозиране на изчислителните и В/И разходи, генерира „план“ за възлагане на задачи на ресурсите, с цел оптимизация на специфицирана функция на производителността.

Времеви ограничения се предопределят не само от задачата, но и от динамичното разпределение на ресурсите във времето. Изчислителните ресурси се споделят пространствено като достъпът се управлява от система за управление на опашките.

В зависимост от възприетата функция на производителността, планиращите програми се разделят на следните категории:

✓ *Ориентирани към задачи* – функцията на производителността е свързана с общата системна производителност при изпълнение на изчислителния товар;

✓ *Ориентирани към ресурси* – оптимизират използването на ресурсите или налагат критерий за справедливост, фокусът е върху оптимизирането на производителността на ресурса;

✓ *Ориентирани към приложения* – фокусът е върху оптимизирането на производителността на приложението;

В зависимост от нивото, на което функционират, планиращите програми се разделят на следните категории:

✓ *На ниво процеси* – планиращата програма включва компоненти на ниво операционна система, които координират изпълнението на локалните процеси и осъществяват ефективното използване на локалното процесорно време; тези планиращи програми в общия случай не се считат като част от системата GRM;

✓ *На ниво задачи* – координират изпълнението на отделните задачи;

✓ *На ниво ресурси* – управляват използването на ресурсите;

✓ *Метапланиращи програми* – координират работата на други, обикновено локални планиращи програми. Анализират потенциални планове за изпълнение на приложенията на различни ресурси и изграждат цялостния план.

В средата на ГРИД съществуват всички изброени видове планиращи програми, които преследват различни, в някои случаи конфликтни цели. Така например, подобряването на производителността на дадено приложение може да влоши производителността на другите приложения, както и на системната производителност, и обратното. Аналогично, планиращи програми, които оперират на различни нива, може да не вземат предвид адекватно дейностите на друга планираща програма. Следователно, съществува необходимост от явно или неявно взаимодействие между планиращите програми за ефективното изпълнение на приложенията в средата на ГРИД.

При създаването на сценарий за планирането се използват 5 градивни базови модела:

1. *Модел на ресурсите* – описва характеристиките, производителността, наличността и цената на ресурсите;

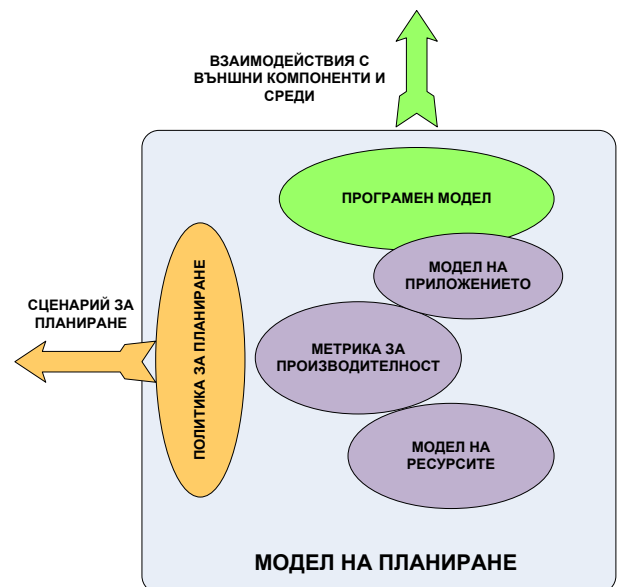
2. *Модел на приложението* – предоставя се на потребителите за описание на приложенията. Базира се на формализмите на потока данни или на работния поток, като фактически представлява граф на зависимостите по данни.

3. *Метрика за производителността* – с цел измерване и подобряване;

4. *Политика за планиране* – обхваща два компонента: *процес на планиране* (последователност от дейности, напр., поемане на ново задание, наличност на ресурс) и *алгоритъм за планиране* – дефинира начина, по който задачите се разпределят на ресурсите;

5. *Програмен модел* – предоставя се на външни компоненти или среди за взаимодействие с планиращата програма и описване в детайли на качествата на приложното програмиране.

Концепцията на предложения модел за планиране е представена на фиг. 1.



Фиг. 1. Модел на планиране на ресурсите в ГРИД

Моделът за планиране използва метаданни под формата на наредени форми атрибут/стойност. Метаданните се асоциират с описанието на ресурсите и приложенията.

Моделът на ресурсите осигурява абстракции при описанието на характеристиките и производителността на хостове, набори данни, софтуерни компоненти, и мрежови връзки.

✓ Хост се моделира като двойка {хост ID, описание}.

✓ Набор данни се моделира като тройка {данни, описание, хост ID}.

✓ Копие на набора данни (реплика) се моделира като тройка {копие_данни, описание, хост ID}.

✓ Софтуерен компонент се описва като тройка {софтуер ID, описание, хост ID}.

Моделът на приложението се представя с метаданните на задачите му, както следва:

{задача_ID, данни_ID_IN, данни_ID_OUT, софтуер_ID, параметри, описание, хост_ID}

Една задача на приложението е абстрактна, ако се реферират абстрактни ресурси. Приложението се нарича абстрактно ако съдържа поне една абстрактна задача. Преди планирането се създават инстанции на абстрактни задачи, които подлежат на планиране.

Входът на модела за планиране съдържа:

(1) три множества D_{input} , S_{input} , и H_{input} , съдържащи описанието на входните данни, софтуерните компоненти и виртуалните (абстрактните) хостове, изисквани да се изпълни заданието;

(2) описание на абстрактното задание job_{input} {задача_ID, данни_ID_IN, данни_ID_OUT, софтуер_ID, параметри, описание, хост_ID};

(3) функция за оптимизация на производителността $Perf_Opt$.

Изходът на модела за планиране съдържа:

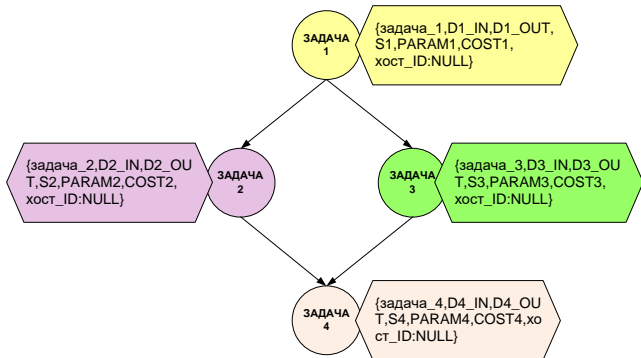
(1) три множества D_{output} , S_{output} , и T_{output} , съдържащи описанието на входните данни, софтуерните компоненти и задачите;

(2) времеви график, показващ стартовото време за всяка задача на заданието съобразен със зависимостите между задачите в рамките на заданието.

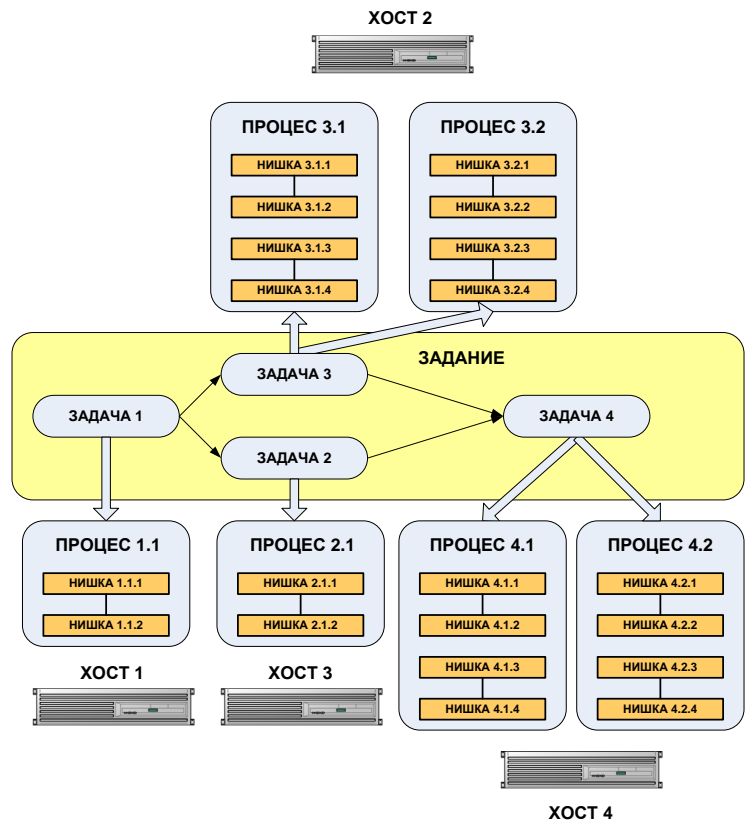
Множествата D_{output} , S_{output} и T_{output} , описват инстанции на елементите на задачите от множеството job_{input} .

На фиг.2 са представени абстрактните задачи в рамките на примерно приложение под формата на насочен ацикличен граф на приложението.

На фиг. 3 е представено планиране на ресурсите за примерното приложение от фиг. 2.



Фиг.2. Представяне на абстрактни задачи в рамките на примерно приложение под формата на насочен ацикличен граф на приложението



Фиг. 3 Планиране на ресурсите за изпълнение на примерното задание от фиг. 2.

3. АРХИТЕКТУРЕН ПРОЕКТ НА РЕСУРСНИЯ БРОКЕР

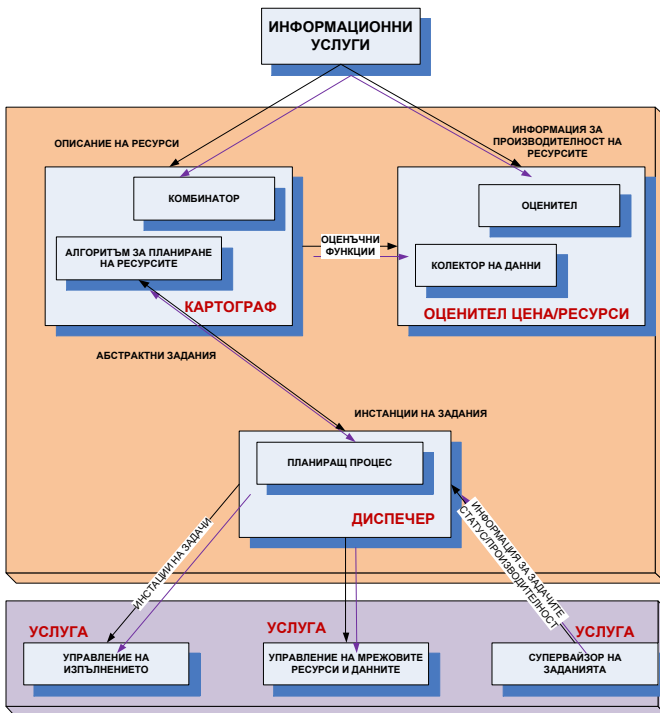
Архитектурата на предложения ресурсен брокер включва три базови модула: картограф, оценител цена/ресурси и диспечер (фиг.4).

Предназначението на *картографа* е да конструира план на базата на абстрактните задачи. Картографът включва два основни компонента: алгоритъм за планиране на ресурсите и комбинатор. Ролята на комбинатора е да открие най-добрата комбинация от предлаганите ресурси за удовлетворяване на изискванията на конкретната заявка. При вземането на решение картографът използва описанието на ресурсите и оценките, изготвени от оценителя цена/ресурси.

Оценителят цена/ресурси получава информация за наличните ресурси от колектора за данни относно тяхната настояща и бъдеща наличност, производителност и цена за използване. Оценителят изчислява стойностите на оценъчните функции на базата на получената информация от колектора и необходимото време за изпълнение на заявката.

Диспечерът осъществява дейностите по планирането. Той получава абстрактните задачи, изисква съответните планове от картографа, и комуникира с услугите за управление на изпълнението за стартиране изпълнението на

заявката. По време на изпълнението на заявката диспечерът получава известията от услугите за мониторинг на заданията за важни събития и също така, при необходимост пренасочва неизпълнените части от заданияето.



Фиг. 4 Архитектура на ресурсния брокер

За всяка планирана за изпълнение задача, диспечерът осъществява следните дейности:

1. Събира информация за характеристиките и производителността на наличните ресурси;
2. Прави селекция на ресурсите, които могат да се използват за изпълнението на задачите, на основата на ресурсното описание и информацията за производителността;
3. Конструирание на възможни варианти за планиране;
4. Избор на оптимално планиране от създадените варианти по отношение на производителността.
5. Подава заявка за изпълнение на задачата.
6. Осъществява мониторинг на изпълнението на задачите;
7. Осъществява пренасочване на изпълнението на задачите в съответствие с новопостъпила информация за статуса на задачата и наличните ресурси.

4. СЦЕНАРИЙ ЗА ВЗАИМОДЕЙСТВИЕТО НА РЕСУРСНИЯ БРОКЕР С УСЛУГИТЕ В ГРИД

Трите базови модула на предложения ресурсен брокер комуникират посредством P2P технология, тъй като в общия случай могат да бъдат ситуирани на различни локации. И трите базови модула комуникират със следните ГРИД услуги:

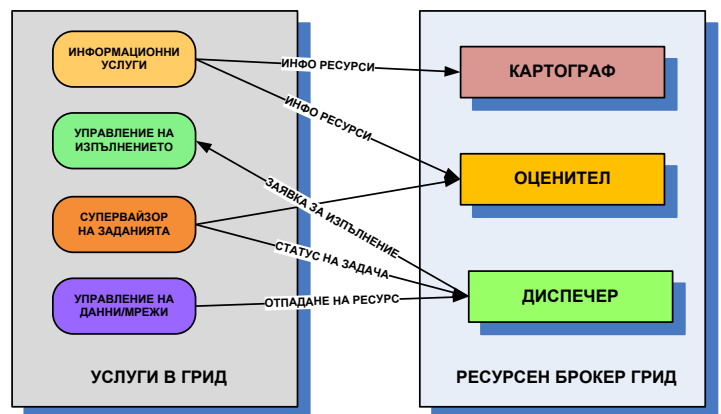
- ✓ информационните услуги за ресурсите, тъй като им е необходима информация за наличните ресурси;
- ✓ услугите за управление на изпълнението за подаване на заявките за изпълнението на задачите;
- ✓ Услугите на супервайзора на заданията за мониторинг на изпълнението на заданията;
- ✓ Услуги за управление на данните/мрежовите ресурси.

Информационните услуги в ГРИД взаимодействат с ресурсния брокер като му предоставят информация за наличните ресурси, като тази информация се консумира от ресурсния оценител и картографа.

Диспечерът на ресурсния брокер подава заявки за изпълнение на задачи (инстанции) към услугите за управление на изпълнението. Той взаимодейства, също така, със услугите за надзор на задачите, за да може да осъществи мониторинг над изпълнението на задачите.

Диспечерът взаимодейства и с услугите за управление на данните/мрежовите ресурси като получава актуална информация за динамиката на наличността на ресурсите в ГРИД и в случай на отказ на ресурс, да направи пренасочване на изпълнението на проблемната задача.

На фиг.5 е показан сценарий за взаимодействието на ресурсния брокер с услугите в ГРИД като технологията за комуникация е P2P предвид ситуирането на услугите и брокера на различни локации.



Фиг. 5. Сценарий за взаимодействието на ресурсния брокер с услугите в ГРИД

Предложената архитектура на ресурсния брокер лесно се вписва в модела OGSA и кореспондира с релевантния подход за изграждане на ГРИД мидълуер, базиран на услуги, при който логиката за планиране на заданията и картографирането за разпределени между няколко услуги като Услугата за планиране на заданията и Селекция на потенциалните ресурси.

Услугата за планиране на заданията взема решение на кой ресурс и в кой момент да се изпълни задачата.

Услугата Селекция на потенциалните ресурси изготвя селекция от предварително подбрани ресурси, отговарящи на изискванията на заявка за ресурс. Тази услуга обаче, не извършва планиране и не взаимодейства с никоя от услугите за планиране.

5. ЗАКЛЮЧЕНИЕ

В статията се предлага архитектура на ГРИД ресурсен брокер, базиран на модел за планиране с метаданни. Предложен е сценарий за планирането, при който моделът за планиране обхваща 5 базови модела – на ресурсите, приложението, производителността, метрика за производителността, политика за планиране и програмен модел. Моделът за планиране използва метаданни под формата на наредени форми атрибут/стойност. Метаданните се асоциират с описанието на ресурсите и приложенията. Моделът на приложението се представя с метаданните на задачите му. Дефинирани са множествата метаданни на входа и на изхода на модела на планиране.

Предложен е архитектурен проект на ГРИД ресурсен брокер включващ три базови модула: картограф, оценител цена/ресурси и диспечер. Предложен е сценарий за взаимодействието на ресурсния брокер с услугите в ГРИД.

Насоките за бъдещи изследвания обхващат симулационни експерименти за верификация и валидиране на модела на планиране и архитектурния проект.

ИНФОРМАЦИОННИ ИЗТОЧНИЦИ

1. OGF OGSA Resource Selection Services WG: <https://forge.gridforum.org/sf/projects/ogsa-rss-wg>
2. OGF Grid Scheduling Architecture Research Group: <https://forge.gridforum.org/sf/projects/gsa-rg>
3. The gLite website, <http://glite.web.cern.ch/glite/>
4. Job Submission Description Language: <http://www.ggf.org/documents/GFD.56.pdf>
5. Abishek Singh, *GRID Resource Broker*, 2016: <https://www.cse.buffalo.edu/faculty/miller/Courses/Grid-Seminar/GridResourceBroker.pdf>
6. Srikumar Venugopal, Krishna Nadiminti, Hussein Gibbins and Rajkumar Buyya, *Designing a Resource Broker for Heterogeneous Grids*: <http://www.cloudbus.org/reports/gridbus-broker-design.pdf>
7. Enis Afgan, *Role of the Resource Broker in the Grid*: <https://pdfs.semanticscholar.org/2b1a/f88bb37986f0bdae938c1b1d8a3dc7458be1.pdf>
8. Arshad Ali, Ashiq Anjum, Atif Mehmood, Richard McClatchey, Ian Willers, Julian Bunn, Harvey Newman, Michael Thomas, Conrad Steenberg, *A Taxonomy and Survey of Grid Resource Planning and Reservation Systems for Grid Enabled Analysis Environment*.

<https://pdfs.semanticscholar.org/a81b/0d7e50d20e15a9536d63a9c37d2f158dff71.pdf>

9. Susmita Singh, Madhulina Sarkar, Sarbani Roy, Nandini Mukherjee, *Genetic Algorithm based Resource Broker for Computational Grid*, Part of special issue: First International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013, Procedia Technology, Volume 10, 2013, Pages 572-580

10. Manoj Kumar Mishra, Yashwant Singh Patel, Yajnaseni Rout, G.B. Mund, *A Survey on Scheduling Heuristics in Grid Computing Environment*, I.J. Modern Education and Computer Science, 10, 2014, pp. 57-83.

11. Access Grid <http://www.accessgrid.org>

12. Manoj Kumar Mishra, Yashwant Singh Patel, "The Role of Grid Technologies: A Next Level Combat with Big Data", Chapter Techniques and Environments for Big Data Analysis, Springer series Studies in Big Data, Springer International Publishing, 2016, pp. 181-191.

13. Manoj Kumar Mishra, Yashwant Singh Patel, Moumita Ghosh, G. B. Mund, *A Review and Classification of Grid Computing Systems*, International Journal of Computational Intelligence Research ISSN 0973-1873 Volume 13, Number 3 (2017), pp. 369-402

14. Jia Yu and Rajkumar Buyya, *A Taxonomy of Workflow Management Systems for Grid Computing*, <http://www.cloudbus.org/reports/GridWorkflowTaxonomy.pdf>

15. Service Level Agreement based Job Scheduling, <http://www.grid scheduling.org/>

ЗА АВТОРИТЕ:



Пламенка Боровска е професор, доктор на науките, в катедра Информатика, Факултет по приложна математика и информатика, Технически университет – София pborovska@tu-sofia.bg



Хараламбос Докомес е докторант в катедра Компютърни системи, Факултет по компютърни системи и технологии, Технически университет – София ch_dokomes@yahoo.gr