

УДК 004

## АРХИТЕКТУРЫ GRID СИСТЕМ

Жагдагцэрэн Н.-Э., Саклаков В. М.  
Научный руководитель: Иванов М.А.

*Национальный Исследовательский Томский политехнический университет,  
634050, Россия, г. Томск, пр. Ленина, 30  
E-mail: romanov\_ky@mail.ru*

*The article describes the classification grid systems, methods of scheduling and resource monitoring*

**Key words:** *grid-architecture, distributed computing, heterogeneous data.*

**Ключевые слова:** *grid-архитектура, распределенные вычисления, гетерогенные данные.*

На сегодняшний день в большом количестве организаций сконцентрировано множество вычислительных средств. Они имеют распределенный характер, гетерогенную структуру и, зачастую, не всегда используются по назначению, либо просто простаивают. С другой стороны, существует множество задач достаточно большого масштаба, для решения которых такие ресурсы будут востребованы. На данном разрыве и появились технологии Grid Computing, т. е. технологии распределенной обработки данных в гетерогенных вычислительных кластерах. Анализ Grid-архитектур посвящена настоящая работа. Так же стоит упомянуть параллельно развивавшиеся технологии: Cloud computing (облачные вычисления), jungle computing (вычислительные джунгли) [1].

Классическая схема Grid-вычислений строится на основе распределенных систем хранения информации и процессорных мощностей. Использование данного метода позволяет эффективно задействовать простаивающие вычислительные ресурсы, которые могут быть глобально распределены. Однако на сегодня Grid-системы часто используются внутри предприятий, в первую очередь по соображениям безопасности.

Grid-система прежде всего должна управлять отношениями между ресурсами любых потенциальных участников планируемого проекта. Одной из главных проблем здесь является обеспечение взаимодействия между различными языками, программными средствами, платформами. Для осуществления этого взаимодействия в контексте сети необходимы общие протоколы. Именно поэтому Grid-архитектура – это прежде всего система протоколов, определяющих механизмы взаимодействия между элементами сети. На основе стандартных протоколов можно строить инструментальные средства разработки (SDK) и интерфейсы прикладного программирования (API).

Grid-системы имеют следующую классификацию [2]:

- Вычислительные (computational Grid);
- Интенсивной обработки данных (data Grid);
- Сервисные (service Grid);
- Обработки знаний (knowledge Grid);
- Семантические (semantic Grid).

Важным элементом является система диспетчеризации задач (job scheduling) и управления ресурсами Grid. В настоящее время существует три базовые архитектуры. Их особенности представлены на рис. 1. Так же различают [3] два вида диспетчеризации в зависимости

от способа объединения ресурсов для решения задач: много-сайтовое и одно-сайтовое. В первом случае задача может выполняться одновременно на нескольких сайтах. При этом она выходит за границы параллельных систем. Во втором случае задача выполняется лишь в пределах одного сайта. Она не выходит за пределы границ параллельной системы.



Рис. 1. Диспетчерская система управления задачами и ресурсами Grid

Для использования ресурсов сети необходим ее мониторинг. Стратегия мониторинга зависит от того, какая диспетчерская система используется. Централизованная система мониторинга является наиболее простым решением. В ней используется лишь один сервер, задачей которого является опрос и контроль всех серверов вычислений. Однако у нее есть один серьезный недостаток – резкий рост служебной информации. Для поиска ресурсов, в системе с малым их количеством, необходимое время близко к константному. Таким образом, использование данной системы является целесообразным, лишь при небольшом количестве ресурсов в системе. При использовании распределенной системы мониторинга информацию о ресурсе клиент получает непосредственно в нем самом, что позволяет решить задачу непротиворечивости. При этом для поиска нужного ресурса необходимо сканировать все ресурсы Grid-системы, что создает большую нагрузку на каналы связи. При использовании иерархической системы мониторинга используется множество серверов, соединенных в виде древовидного графа. Как клиенты, так и сервера вычислений производят подключение к самым нижним серверам мониторинга в иерархии. Клиент отправляет запрос на вычислительные ресурсы который поднимается по уровням снизу вверх, до тех пор пока на одном их уровней они не будут предоставлены. Клиент останавливает поиск и продолжит его после некоторой задержки если корневой сервер не может выделить ресурсов. Очевидной проблемой является затраченное время на поиск ресурсов. По мнению авторов [4], иерархическая система более эффективна.

### Список литературы

1. Саак А. Э. Прогноз ресурсной оболочки при диспетчеризации в grid-системах с централизованной архитектурой // Известия ЮФУ. Технические науки. – 2014. – № 7. – С. 239–248.
2. Саак А. Э. Управление ресурсами и заявками пользователей в GRID-системах с централизованной архитектурой // XII всероссийское совещание по проблемам управления ВСПУ-2014. – 2014. – С. 7489–7498.
3. Hamscher V., Schwiegelshohn U., Streit A., Yahyapour R. Evaluation of job-scheduling strategies for grid computing // In: Proceedings of the 7th International Conference on High Performance Computing HiPC-2000. India. Springer. – 2000. Lecture Notes in Computer Science. – Vol. 1971. P. 191–202.
4. Симоненко В.П., Щербина О.В. Выбор архитектуры системы мониторинга ресурсов в глобальных GRID системах // Вестник Национального технического университета Украины «Киевский политехнический институт». Серия «Информатика, управление и вычислительная техника». – 2012. – № 57. – С. 25–29.