

СПОСОБ ФОРМИРОВАНИЯ СТРУКТУРЫ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ТРАФИКОМ В GRID-СИСТЕМАХ

Национальный технический университет Украины «КПИ»

Предложен способ организации структуры распределенной системы управления трафиком в GRID-системах на базе агентной технологии. Множество агентов объединены в оверлейную систему управления. Предложен способ разбиения GRID-системы на множество виртуальных частных GRID-систем

Введение

В настоящее время использование GRID-систем широко распространено в разных областях, например, для совместных научных исследований, расчета финансовых рисков и анализа результатов проектирования. GRID-системы позволяют решать задачи повышенной сложности, которые требуют большой вычислительной мощности. Кроме того, GRID-системы позволяют понизить уровень задержек за счет автоматизированного распределения задач и ресурсов между компонентами системы.

В ближайшем будущем GRID-системы будут использоваться не только крупными организациями, но и конечными пользователями для решения сервис-ориентированных задач, что приведет к появлению новых более жестких требований предъявляемых к GRID-системам. Предполагается обеспечить общедоступный, надежный и сравнительно простой доступ к вычислительным GRID-структурам, а также к устройствам хранения и обработки данных. Если это удастся сделать, влияние GRID-технологий на развитие вычислений окажется очень существенным.

В связи с этим возникает необходимость разработки системы объединения не только локальных ресурсов, но и географически распределенных ресурсов в единое целое.

Обзор существующих решений

В настоящее время выделяют четыре типа инфраструктур GRID-систем, в зависимости от области применения:

- вычислительные,
- информационные,
- сервисные,
- интеллектуальные.

Основной задачей вычислительных GRID-систем является объединение простаивающих вычислительных ресурсов в единое целое для обеспечения высокой вычислительной мощности (например, NEESGrid для изучения моделирования землетрясений).

Информационные GRID-системы призваны обеспечивать доступ к распределенным информационным ресурсам. Особое значение данный тип GRID-систем приобретает в связи с виртуализацией приложений, а также в тех случаях, когда передача приложений потребует меньшего объема трафика, чем передача данных.

В сервисных GRID-системах сочетается физическое соединение узлов с архитектурой *p2p* (программа – к – программе).

Интеллектуальные GRID-системы объединяют соединение узлов с возможностью управления системными и сетевыми ресурсами, а также ресурсами систем хранения данных.

В работе [1] проведен анализ существующих инфраструктур GRID-систем, по результатам которого можно сделать вывод, что одной из наиболее востребованных инфраструктур являются сервисные GRID-системы (рис. 1).

Кроме того, по масштабам конфигурации *GRID*-системы можно разделить на [2]:

- кластерные *GRID*-системы,
- внутрисетевые *GRID*-системы масштаба подразделения (*Intra-GRID*),
- внешнесетевые масштаба предприятия (*Extra-GRID*),
- межсетевые выходящие за рамки предприятия (*Inter-GRID*).

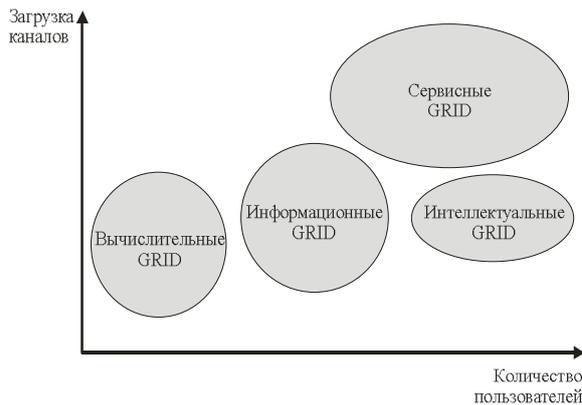


Рис. 1. Анализ загрузки каналов *GRID*-систем различных инфраструктур

Из рис. 1 следует, что наиболее широко используются сервисные *GRID*-системы, в которых количество пользователей существенно превышает пользователей других *GRID*-систем.

Важно отметить, что сервисные *GRID*-систем призваны обеспечивать не только выполнение задач пользователей, но и контролировать функционирование самой системы. В связи с этим, с увеличением количества пользователей и ресурсов системы, возникает задача уменьшения и обеспечения равномерной загрузки каналов передачи данных.

В связи с возможным изменением состава распределенной системы, вследствие увеличения или уменьшения количества ресурсов, вопрос структуры *GRID*-системы приобретает особую актуальность.

Постановка задачи

В общем случае задача формирования распределенной системы управления трафиком сводится к организации эффективного управления процессами, рас-

пределению задач и ресурсов в сервисных *GRID*-системах.

Основной причиной поставленной задачи является то, что существующие решения не обеспечивают требуемого уровня качества обслуживания *QoS* (*Quality of Service*), что для некоторых классов задач является важнейшим показателем эффективности функционирования всей *GRID*-системы.

Решение задачи

В связи с тем, что в *GRID*-системах широко используется самоорганизация и автоматическое конфигурирование ресурсов, все чаще возникают проблемы с управлением, масштабируемостью и обеспечением заданного уровня *QoS* в *GRID*-системах большой размерности [3, 4].

Для обеспечения эффективной масштабируемости, доступности ресурсов и обеспечения требуемого уровня *QoS* предлагается разбить *GRID*-системы на подсистемы и назначить каждой из них агента, который будет выполнять функции управления, планирования, распределения и выделения ресурсов.

Правильное распределение ресурсов может предотвратить монопольное использование одного ресурса одним и тем же процессом, распределяя выполняемые задачи на различные классы обслуживания.

Под классом обслуживания *GRID*-систем *SC* (*service class*) будем понимать задачи пользователя, к которым выдвигаются одинаковые требования к ресурсам.

В работе [5] под эффективностью использования ресурсов понимается отношение времени затраченного на выполнение задач ко времени простоя в результате ввода-вывода данных.

Кроме того, каждый *SC* должен автоматически получать определенную полосу пропускания и, в дальнейшем, иметь возможность использовать эту полосу, не думая о том, что его полоса может быть занята процессом другого *SC*.

Для эффективного использования ресурсов назначим группе ресурсов или, в частном случае, индивидуально каждому из них управляющий механизм или агента, который будет осуществлять взаимодействие ресурса с другими компонентами *GRID*-системы.

Для осуществления взаимодействия агентов системы между собой выбирается центральный агент системы *CAGS* (*Central Agent of GRID System*). Такой агент *CAGS* централизованно следит за всеми агентами и выполняет задачи управления и мониторинга всей *GRID*-системы.

CAGS назначается, при инициализации системы и его роль может выполнять любой агент системы. При выходе *CAGS* из строя все его функции передаются ближайшему агенту с минимальным временем отклика и максимальной степенью связности.

Для осуществления взаимодействия агентов системы с *CAGS* наиболее оптимально использовать виртуальные каналы передачи данных.

В работе [6] рассматриваются существующие методы построения виртуальных каналов на основе *GRID* (рис. 2), среди них можно выделить:

- виртуальные каналы, построенные поверх *GRID*-систем *Grid over VPN* (*GoVPN*),
- сообщество *GRID*-систем *Grid Community* (*GC*) и *Ad-hoc GRID*-системы *Ad-hoc Grid* (*AG*),
- виртуальные частные *GRID*-системы *Virtual Private Grid* (*VPG*).

По результатам анализа методов построения виртуальных каналов сделан вывод, что для решения поставленной задачи наиболее эффективно использовать *VPG* (рис. 2в).

Важно отметить, что *VPG* представляет собой промежуточный уровень взаимодействия между системой и транспортной средой *GRID*. Таким образом, использование *VPG* позволяет одновременно решить задачу управления сетевы-

ми ресурсами и контролем безопасности *GRID*-систем.

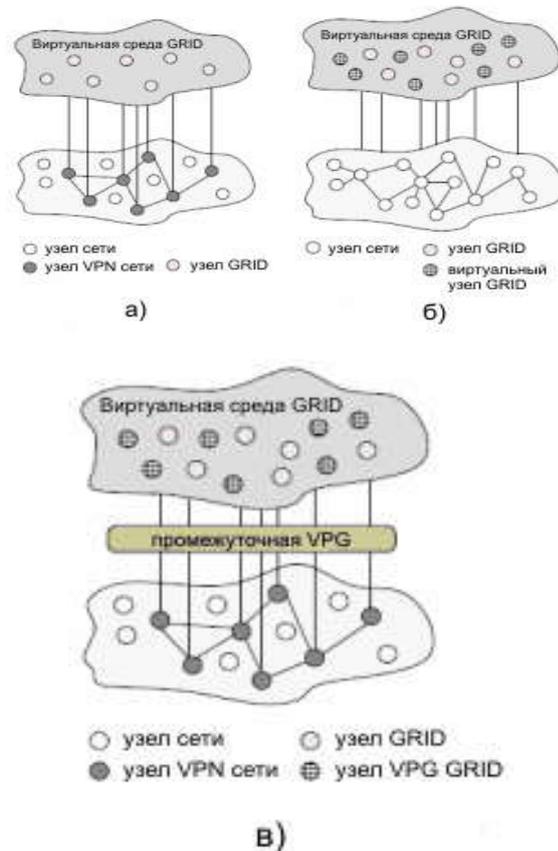


Рис. 2. Методы построения виртуальных каналов на основе *GRID*:
а) *GoVPN*, б) *GC* и *AG*, в) *VPG*

Кроме того, в комбинации с распределением сетевых ресурсов и ресурсов хранения данных, *GRID*-система может быть представлена как динамическое объединение оверлейных *GRID*-систем (*VPG*), где каждый из классов *VPG* представляется как *GRID*-система.

Множество объединенных между собой *VPG* и управляемых с помощью *CAGS* можно рассматривать в качестве оверлейной системы, которая представляет собой единую логическую систему управления поверх существующей физической.

В работе предлагается способ организации структуры распределенной системы управления трафиком в *GRID*-системе на основе агентов с использованием оверлейной технологии.

Рассмотрим *GRID* - систему представленную в виде графа $G = \{V, E\}$, где V – множество ресурсов *GRID*-системы, а E – множество связей между ресурсами V .

Алгоритм формирования структуры:

1. На первом этапе производится инициализация системы, в процессе которой определяются все доступные ресурсы и каналы передачи данных.

2. На следующем этапе определяется количество агентов *GRID*-системы N_a , где в качестве агентов управления *GRID*-системой назначаются информационные ресурсы, состоящие из одного или нескольких ресурсов хранения, в зависимости от количества ресурсов. Агент управления выбирается по формуле:

$$\frac{V}{2} \leq N_a \leq V. \quad (1)$$

3. Для решения задачи определения месторасположения агентов заполним матрицу инцидентности MI для графа G , и с ее помощью определим максимальную степень вершин графа v_m по формуле:

$$S = \max(v_m), \forall m \in E, \quad (2)$$

где m – связь графа G , v – вершина графа G , m_e – степень связности вершины v .

4. В качестве условия выбора абонентской системы v_m агентом a_i определим следующее условие:

$$a_i = \{v_m \mid (1 - p_i) \rightarrow \max \forall v_m \in V\} \quad (3)$$

где p_i – вероятность удаления вершины или ее перемещения, которое приводит к изменению структуры графа.

5. Для определения месторасположения *CAGS*, для вершины $v_m \in V$ определим также плотность сетевого окружения k -го порядка $\delta_k(v_m)$ по формуле:

$$\delta_k(v_m) = \frac{S(v_m) + \sum_{v_i, v_j \in \Gamma^k(v_m)} e_{i,j}}{S(v_m)}, \quad (4)$$

где $S(v_m)$ – максимальная степень связности вершины v_m ; $\Gamma(v_m)$ – соответствие первого порядка, множество всех вершин, связанных с вершиной v_m , $\Gamma^2(v_m) = \Gamma(\Gamma(v_m))$; $e_{i,j}$ – связь между смежными вершинами v_i и v_j с максимальной степенью; k – порядок сетевого окружения вершины v_m .

Из всех полученных $\delta_k(v_m)$ случайным образом выбирается вершина с максимальным значением первого и второго порядка, и такая вершина становится *CAGS*.

6. На следующем этапе формируется множество виртуальных каналов *VPG*, которое представляется в виде подграфа $G^0 = \{V^0, E^0\}$ исходного графа $G = \{V, E\}$, где $V^0 \subseteq V$ – множество ресурсов системы, в которых расположены агенты управления, $E^0 \subseteq E$ – множество ребер, создающих виртуальные каналы оверлейной служебной подсети.

В качестве примера рассмотрим распределенную систему, состоящую из 14 узлов-ресурсов, представленную на рис. 3.

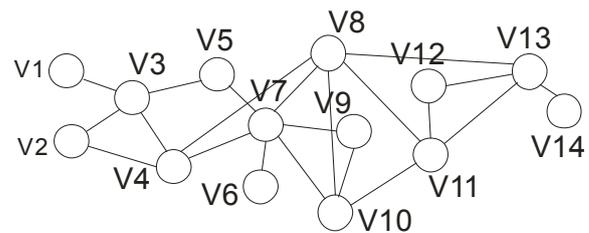


Рис. 3. Граф распределенной системы

Для распределенной системы, представленной на рис. 3, степени связности вершин рассчитаем по формулам (1) – (3), и представим в виде таблицы (табл. 1).

Таблица 1. Степени связности вершин графа

№ вер.	Связ.	№ вер.	Связ.	№ вер.	Связ.
V1	1	V6	1	V11	4
V2	2	V7	6	V12	2

продолжение табл.

V3	4	V8	5	V13	4
V4	4	V9	2	V14	1
V5	2	V10	4		

По результатам анализ значений максимальной связности вершин, можно сделать вывод, что агентами будут 7 вершин, а именно V3, V4, V7, V8, V10, V11, V13. Для полученных вершин рассчитаем плотность сетевого окружения первого и второго порядков по формуле (4). По результатам расчетов получим, что *CAGS* станет вершина V7. Далее формируем множество виртуальных каналов в виде оверлейной системы из 7 узлов, представленной на рис. 4.

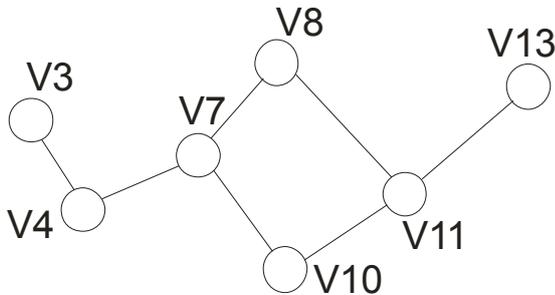


Рис. 4. Граф оверлейной системы

Основное преимущество оверлейных сетей заключается в том, что они позволяют разрабатывать и эксплуатировать новые крупномасштабные распределённые сервисы без внесения каких-либо изменений в физическую структуру *GRID*-системы.

Если разработать оверлейную сеть, которая бы имела некоторый вид механизма резервирования канала, то возможно будет устанавливать логические маршруты по физической среде. Тогда пользовательские связи будут устанавливаться через логические маршруты.

Основная особенность оверлейной сети – ее гибкость. Логическая сеть также может использоваться как механизм защиты от сбоев, то есть некоторые логические маршруты могут быть установлены как резервные копии.

Также, важным моментом является алгоритм маршрутизации, при помощи

которого осуществляется передача по сети управляющей информации.

Например, наиболее распространенным является алгоритм лавинной маршрутизации, который не требует знания топологии сети. Узел-источник широковещательно посылает пакет всем своим соседям. Каждый получивший пакет узел передает его дальше по всем своим выходным каналам связи, кроме того, по которому пакет прибыл, и так до тех пор, пока пакет не достигнет всех узлов сети.

В качестве примера использования предложенного подхода на *GRID*-системе рассмотрим стандартную *GRID*, представленную на рис. 5.

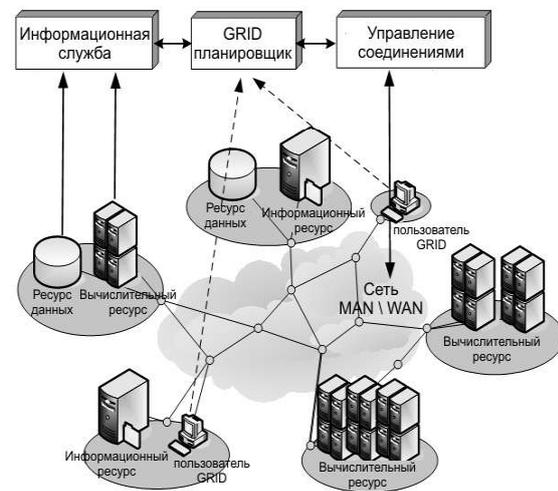


Рис. 5. Стандартная *GRID*-система

В стандартной *GRID*-системе в качестве ресурса данных представлен ресурс, который обеспечивает непосредственный доступ к хранению информации. Вычислительный ресурс обладает мощностями для вычисления сложных математических задач и представляет собой мультипроцессорную систему. В качестве информационного ресурса используют систему взаимодействия с данными, состоящую из одного или нескольких ресурсов хранения.

Пользователь *GRID*-системы представляет собой конечного пользователя, задачи которого выполняются в системе.

Корректную работу *GRID*-системы обеспечивают системы планирования задач и информационная служба.

Вся система об'єднується з допомогою *MAN* (*Metropolitan Area Network*) или *WAN* (*Wide Area Network*) мереж и управляється з допомогою служби управління соединениями.

На рис. 6 приведена модифікована *GRID*-система з використанням *VPG* структури.

VPG структура являється динамічною и не має постійних ресурсів, однак она реагує на зміну структури всієї системи.

Наприклад, при створенні нових маршрутів *VPG* во время перерозподілення ресурсів *GRID*-системи, структура оверлейної мережі динамічно перестраюється.

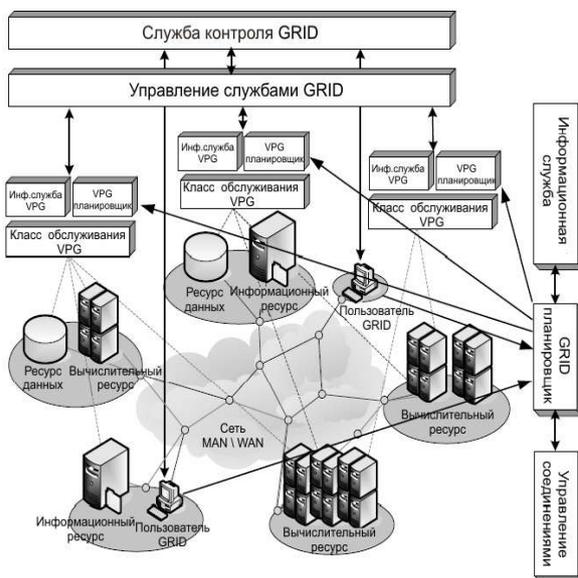


Рис. 6. Модифікована *GRID*-система з використанням *VPG* структури

Перерозподілення ресурсів в системі може викликати більші зміни в характеристиках управління и моніторингу *SC*. Наприклад, при створенні нових *SC* з більш високими пріоритетами, може виникнути ситуація когди на забезпечення виможаного рівня *QoS* система виділить більше ресурсів чем необхідно для рішення задач виконуваних *GRID*-системою. В зв'язі з цим, формування структури розподіленої системи управління трафіком являється актуальною задачею.

Выводы

1. Предложен способ организации структуры распределенной системы управления трафиком в *GRID*-системе на основе агентов с использованием оверлейной технологии, при котором множество агентов объединены в оверлейную систему управления.

2. Предложен способ разбиения *GRID*-системы на множество виртуальных частных *GRID*-систем (*VPG*), который позволяет оперативно реагировать на изменение структуры всей системы не загружая ресурсы *GRID*-системы задачи обслуживания.

Список литературы

1. Turck F., Vanhastel S., Volckaert B., Demeester P. Generic middleware-based platform for scalable cluster computing. *Elsevier Journal on Future Generation Computer Systems*, Vol.18, 2002. – P. 549–560.

2. Thysebaert P., Volckaert B., Leenheer M., Turck F., Dhoedt B., Demeester P. Resource partitioning algorithms in a programmable service Grid architecture. *Lecture Notes in Computer Science. 5th International Conference on Computational Science – ICCS 2005. – Atlanta, 2005. – P. 250–258.*

3. Kephart J.O., Chess D.M. The Vision of Autonomic Computing. *IEEE Computer*, Vol. 36(1), 2003. – P. 41–50.

4. Ganek A.G., Corbi T.A. The dawning of the autonomic computing era. *IBM Systems Journal*, Vol. 42, 2005. – P 5–18.

5. Munoz J.L., Pegueroles J. Security Issues in Virtual Grid Environments. *Computational Science – ICCS 2004. – Poland, Krakow, 2004. – P. 174–178.*

6. Bubak M., Geert D. Van A., Sloot M. A., Dongarra J.J. 4th International Conference on Computational Science – ICCS 2004. – Poland, Krakow, 2004. – P. 174–180.