

УДК 004.724.4(045)

<sup>1</sup>Кулаков Ю.А., д-р тех. наук  
<sup>2</sup>Клименко І.А., канд. техн. наук**ОСОБЕННОСТИ РЕАЛИЗАЦИИ ДИНАМИЧЕСКОЙ GRID СРЕДЫ**<sup>1</sup>Национальный технический университет Украины «КПИ»<sup>2</sup>Институт компьютерных технологий

Национального авиационного университета

*Представлен обзор проектов, различных компонентов и аспектов динамической GRID среды. Исследование сосредоточено в области реализации концепции GRID вычислений в среде с динамической структурой, которая часто упоминается как персональная GRID, настольная GRID, «peer-to-peer» GRID, динамическая GRID. Рассмотрены вопросы адаптации существующих методов реализации GRID для использования в мобильных компьютерных сетях*

**Введение**

Технология *GRID* используется для реализации географически распределенной вычислительной и информационной среды, которая объединяет в единую инфраструктуру ресурсы различных типов и обеспечивает коллективный доступ к этим ресурсам. Технология призвана создать принципиально новый вычислительный инструмент для развития высоких технологий в различных сферах человеческой деятельности [2, 8].

**Постановка проблемы**

В классической реализации *GRID* фокусируется на обеспечении распределенных вычислений в фиксированной среде с заданной конфигурацией. Основное решение – объединение ресурсов в межпрограммном слое. Организации или другие сообщества пользователей, желающие совместно использовать вычислительные ресурсы, обычно создают управляющую инфраструктуру – центральную *GRID*-сеть, в которых узлами и предлагаемыми *GRID*-услугами управляются специальные *GRID*-администраторы. Пользователям предлагают свои услуги отдельные программные системы промежуточного слоя и приложения, опирающиеся на идеи *GRID* – порталы, механизмы поиска, сети распределенных данных, службы авторизации и т. п. Однако все эти службы автономны и не являются интероперабельными, имеющими способность совместного использования или совместной деятельно-

сти. Кроме того качественное внедрение масштабной *GRID* далеко не простое задание, таким образом, необходимость существования специальной инфраструктуры *GRID* обусловлена сложностью установки, поддержки, управления, масштабирования *GRID*-сети, функциональные ограничения программного обеспечения.

Одна из современных задач развития *GRID* это общедоступность и расширение области ее использования, что требует значительного уменьшения сложности ее установки и поддержки. В работах [7, 15] развитие *GRID* рассмотрено с точки зрения «следующего поколения Интернет». Действительно, безграничное распространение Интернет стало возможным благодаря тому, что доступ в Интернет был сделан интуитивно понятным и прозрачным для любого пользователя, что привело к экспоненциальному возрастанию их количества. Таким же образом общедоступная *GRID* может выглядеть не только, как предоставление вычислительных и информационных ресурсов для совместных научных исследований, а как предоставление по требованию инфраструктуры и приложений, ориентированных на использование высоких технологий во всех сферах деятельности человеческого сообщества – медицина, электронные медиа, инженерная сфера, обеспечение коммуникаций для электронного бизнеса и тому подобное для широкого слоя пользователей в хорошо защищенной,

общедоступной, расширяемой и стандартизированной компьютерной среде через Интернет.

Промежуточное программное обеспечение *GRID*, поддерживающее массовое распространение *GRID*, должно обладать многими свойствами инфраструктуры «по требованию», как сформулировано в [4]. Оно должно быть разделяемым между разными пользователями, предоставлять стандартизированные службы и протоколы связи, а также быть гибким и расширяемым. Вычислительная парадигма современной *GRID* [2, 8] строится вокруг основной цели: предоставлять ресурсы (такие, как вычислительная мощность, источники данных, специальные устройства и даже люди) столь же легко, как электричество предоставляется через электрическую сеть.

Предъявляемые требования к интеграции и интероперабельности большого количества приложений, обусловили развитие технологии *GRID* в направлении открытой архитектуры *OGSA* (*Open Grid Services Architecture*) [9, 10]. Открытая архитектура призвана интегрировать службы и ресурсы распределенных, гетерогенных, динамических сред и сообществ. Таким образом, в модель *OGSA* введена концепция *GRID*-службы, опирающаяся на принципы и технологии, предложенные как сообществом *GRID*, так и сообществом *WEB*-служб.[3, 5]. Объединение *GRID* и соответствующих стандартов *WEB*-служб – большой шаг в направлении уменьшения сложности использования, управления и поддержки *GRID*. Использование стандарта *WSRF*, который является низкоуровневым описанием инфраструктуры, реализации модели *OGSA*, предлагает возможность предоставлять низкоуровневую виртуализацию доступных ресурсов, что значительно повышает универсальность *GRID*.

Сегодня *GRID* претендует на значительное увеличение базы пользователей, что влечет за собой расширение возможностей и изначальных идей *GRID* – объединения мировых центров вычислений на

суперкомпьютерах с широким сообществом настольных компьютеров. Приток пользователей *GRID* дает возможность использования неиспользуемых процессорных ресурсов, неактивных рабочих станций, присутствующих практически в любой организации, комбинируя их по требованию. В результате основной идеей сегодняшней *GRID* становится формирование динамической среды, состоящей из взаимодействующих неоднородных вычислительных узлов, без определенной фиксированной инфраструктуры и с минимальными административными требованиями [11, 19].

### **Мобильные GRID**

В свою очередь высокий уровень развития и распространения мобильных технологий во всем мире и широкий диапазон их приложения не смогли не оказать влияние на последующее развитие *GRID*. Поэтому наиболее актуальные современные исследования в области *GRID* посвящены необходимости решения проблемы совместимости *GRID* и мобильных сетей. В данном контексте следует сказать, что использование беспроводных устройств, которые имеют достаточно ограниченные ресурсы вычислительной мощности, энергии, адресного пространства, накладывают существенные ограничения при решении вычислительных задач повышенной ресурсоемкости и хранения больших объемов данных, однако за счет привлечения дополнительных ресурсов, доступных при сочетании с *GRID*, такое использование вполне возможно. Более того использование мобильных *GRID* позволило бы обеспечить применение приложений *GRID* в тех местах где это можно лишь представить, а в будущем технология *GRID* могла бы войти в набор обычных услуг для мобильных пользователей.

Мобильные *GRID* в первую очередь характеризуется динамической структурой и допускают как перемещение пользователей *GRID* так и перемещение запрашиваемых ресурсов. Узлы мобильных *GRID* не являются кластерами или супер-

компьютерами, это произвольный диапазон мобильных устройств, которые в первую очередь характеризуются гетерогенными свойствами: вычислительный ресурс процессора, объем памяти, пропускная способность каналов связи, задержка. Кроме того, мобильная сеть подвержена проблемам разрыва связей и выхода узлов из строя или перемещения за пределы досягаемости. Динамическая природа мобильных *GRID* приводит к тому, что мобильные узлы имеют гетерогенное время выполнения подзадачи, и могут останавливать вычисление произвольно в случае непредвиденного перемещения. В связи с этим совместное выполнение задания временно откладывается и возникают проблемы реконфигурации, репликации, резервирования данных, и тому подобное. Такие нестабильные ситуации называются автономными отказами, они приводят к задержке и блокировке выполнения подзадач и как следствие к частичной или полной потере результатов выполнения задания. Отмеченные проблемы осложняют планирование и управление выполняемыми подзадачами.

Таким образом, характеристики и особенности мобильных *GRID* обуславливают необходимость разработки стратегий управления и механизмов планирования которые бы адаптировались к динамической вычислительной среде.

В этой работе мы проводим анализ различных компонентов и аспектов динамической *GRID* среды. Так же мы рассмотрим примеры проектов, целью которых является создание динамичного вычислительного окружения *Grid*-сети.

### **Особенности и проблемы реализации динамической *GRID***

Чтобы достигнуть поставленной цели формирования динамической *GRID*-среды во внимание должен быть принят ряд проблем, которые обусловлены самой природой динамического окружения [1, 11].

– Расширение *GRID* за счет неспециализированных ресурсов, способствует возрастанию сложности.

– За счет большого количества узлов, которые добавляется в *GRID* на динамической основе, центральное администрирование становится неосуществимым.

– Стремительно возрастает неоднородность системы, а из-за перезагрузок и неисправностей, причиняемых пользователями узлов уменьшается их надежность.

– Природа динамической *GRID* приводит к частым автономным отказам – остановкам вычислений в случае непредвиденного отключения узла. В связи с этим совместное выполнение задания временно откладывается и возникают проблемы реконфигурации, репликации, резервирования данных [1].

– Традиционные инфраструктуры безопасности *GRID* фокусируют внимание на сообщении уровня безопасности и контроля доступа со статичной ориентацией исключительно на используемые ресурсы [6]. Масштабность и спонтанность динамической *GRID* повышает требования к безопасности. Необходимы новые защитные механизмы, чтобы гарантировать, что вредоносный код не сможет повредить законным службам, выполняющимся на *GRID*.

– Сложность динамических *GRID*-приложений и их специфичность еще больше затрудняет непрофессионалам разработку распределенных приложений, когда они пытаются применить *GRID*-вычисления в своей экспертной области.

– Традиционное рассмотрение инициализации прикладных служб по требованию как утилиты, предусматривает разные модели хостинга – размещенный, выделенный, коллективный [3, 5], которые зависят от утомительного ручного администрирования predetermined инфраструктуры.

– Традиционные разработки промежуточного программного обеспечения, которые используют служебные протоколы *WEB* для взаимодействия с традиционным представлением о *GRID*, как о predetermined и статическом соединении высокопроизводительных вычислительных ресурсов, доступ к которым осуществ-

вляется через их рабочие очереди [5], не учитывают особенностей динамической среды.

– Методы и средства разрабатываемые для осуществления динамических *GRID*-вычислений сосредоточены на решении описанных проблем, таким образом перед ними ставятся следующие требования.

– Сама система должна быть способной справиться с динамическими изменениями топологии базовой сети и неоднородностью узлов, чтобы автономно формировать динамическую *GRID*.

– Промежуточное программное обеспечение должно поддерживать *GRID*-приложения в динамически сформированной среде, которая хорошо справляется с частыми изменениями в инфраструктуре.

– Однако, идея *GRID*-вычислений не ограничивается понижением сложности создания и обслуживания рабочей среды, поддерживающей *GRID*-приложения. Специализированная среда *GRID* также должна стимулировать простую и быструю разработку новых *GRID*-приложений с целью сделать возможным применение *Grid*-вычислений в большем количестве предметных областей.

– Система *GRID*-приложений должна быть гибкой и автономной, легкой для понимания, которая скорее помогает разработчикам, чем ошеломляет их функциональностью, и даже может частично помочь в достижении быстрой разработки приложений в различных предметных областях.

– Среда разработки, должна базироваться на классических концепциях разработки программного обеспечения *GRID*, интегрироваться в существующую *GRID* и отвечать основной цели – предоставлению поддержки в разработке как специалистам промежуточного программного обеспечения, так и экспертам в разных предметных областях, не имеющим глубоких знаний о разработке распределенных систем или промежуточного программного обеспечения.

– Способствовать направлению к истинным *GRID*-вычислениям, уменьшающий препятствия на пути к признанию парадигмы *GRID* более широким сообществом пользователей.

– Идея динамической *GRID* должна поддерживать динамическую ограниченную инсталляцию с минимальным администрированием. Число неспециализированных узлов в такой *GRID* намного больше, чем в традиционных *Grid*-системах, требующих неагрессивной работы промежуточного программного обеспечения *GRID*. Логическая структура, сформированная промежуточным программным обеспечением динамической *GRID*, должна предоставить большинство основных инфраструктурных служб, которые нужны для предоставления сервисов большому и изменяющемуся множеству пользователей.

Таким образом, представление динамической среды *GRID*, обусловленное включением в себя неоднородных, спонтанно добавляемых вычислительных ресурсов, выходит за рамки существующих сегодня predetermined инфраструктур *GRID*. В динамическую среду входят, как рабочие станции в организациях, так и разрозненные персональные компьютеры. Например, таким образом реализована идея распределенного вычислительного проекта SETI@Home [14].

Динамическое формирование *GRID* также подчиняется парадигмам «по требованию» и «*utility computing*», выдвинутым компанией *IBM* [17]. По этому сценарию, наиболее требовательные вычисления перенаправляются организациям, предлагающим вычислительную мощность или требуемые ресурсы. В этих проектах также развивается идея динамической аренды ресурсов с минимальными административными издержками. Таким образом, реализуется возможность приобретать и конфигурировать нужные ресурсы без участия администратора, который вручную контролирует каждую транзакцию.

### **Особенности известных проектов реализации динамической Grid**

Представим примеры известных завершенных проектов, целью которых является создание динамического вычислительного окружения *GRID*.

Эти проекты исследуют более динамичные формы вычисления *GRID*, чем существующий в настоящий момент стандартные решения. В литературе можно встретить множество различных названий таких *Grid*-сетей: персональные *GRID*, настольные *GRID*, «peer-to-peer» *GRID*, динамические *GRID*.

Авторы проекта [20] предложили облегченную среду *GRID* для разрешения биоинформационных проблем на небольших, «конфиденциально управляемых» *GRID*. При разработке проекта было принято решение не использовать стандартное промежуточное программное обеспечение *GRID*-решений подобно *GLOBUS*, обусловленное тем, что они слишком громоздки и трудны, чтобы использовать в динамичной среде, и недостаточно гибкие, чтобы обеспечить нужды исследователей. Основные критерии проектирования системы – децентрализация, обеспеченная концепцией интервала записи, и платформенная независимость, обеспеченная реализацией в *JAVA*. Исходя из этого, предложенная инфраструктура разрабатывалась с целью обеспечения решения следующих задач: универсальность, распределение ресурсов и планирование, автоматическое распространение прикладного кода к рабочим вычислительным узлам, взаимодействие процессов и управление ресурсами. В работе были представлены так же проблемы проекта – отсутствие безопасного совместного использования ресурсов, пользовательской системы уведомлений и низкое качество сервисных механизмов. Все взаимодействие процессов производится через сокеты и упорядоченные объекты *JAVA*. Это приводит к усложнению конфигурации инфраструктуры во время расширения границ сети и ограничивает примене-

ние предложенной инфраструктуры в других исследовательских проектах. В итоге, отбрасывание стандартных инструментов и протоколов, для достижения специализированных преимуществ, усложнило использование разработанных компонентов в других проектах, и внедрение в этот проект новых разработок из других проектов.

Настольная вычислительная *GRID* для коммерческих решений представлена в работе [16]. В каждом узле *GRID* выполняется программа рабочей станции *VMware* с операционной системой *LINUX* в качестве гостевой операционной системы. На каждой гостевой операционной системе выполняется *WebSphere Application Server Aes* от *IBM*, чтобы принимать услуги *WEB*-сервисов, встроенные в *GRID*-приложениях. Система ориентирована на предприятия и реализует малую, но безопасную среду *GRID* эффективную для решения внутренних заданий. Легкость использования и автономность управления не рассматриваются вообще, так как это не является критической проблемой в данной реализации. Предложенная система имеет проблемы интероперабельности и интеграции в существующую *GRID*.

В работе [18] предложены алгоритмы распределения ресурсов, которые могут справиться с отказами узла в динамичной *GRID*-среде. *GRID* в данном случае состоит из динамических узлов, и приложение моделируется как направленный ациклический граф. Авторы обратились только к алгоритмическим проблемам, программное обеспечение к их прикладному сценарию в работе не обсуждается.

Проекты в сфере медицинских *Grid*-вычислений обсуждены в работах [12, 13]. Работа [13] отображает механизм технологического процесса для медицинских приложений с высокой производительностью обработки данных, в работе [12] в свою очередь предложена *GRID*, основанная на методе анализа изображения. Авторы работ заявляют, что расту-

щим интересом в медицинском сообществе является интеграция рабочих станций персональных компьютеров в настольные *GRID*-системы. В отличие от проектов, упомянутых выше, решения, представленные в этих исследованиях, основаны не на легковесных заказных развиваемых *GRID*-системах, а на стандартном сервис-ориентированном промежуточном программном обеспечении для *GRID*. В работе [13] отмечено, что осуществление медицинских приложений на стандартном промежуточном программном обеспечении *GRID* далеко от тривиального благодаря сложной природе как приложения, так и промежуточного программного обеспечения непосредственно. Тут отмечена важная особенность реализации медицинской специфики *GRID* – интеграция с приложениями существующей *GRID* для эффективного участия в мировых программах и проектах.

В работах [11, 19] исследована идея использования одноранговых (*P2P*, *peer-to-peer*) сетевых механизмов для *GRID*-вычислений путем применения первой интеграции инфраструктуры *P2P*-вычислений общего назначения в промежуточном программном обеспечении стандартизированной сервис-ориентированной *GRID*. В этом проекте используются коммуникационные возможности сетей *P2P*, чтобы сделать возможным взаимодействие через сегменты сети в среде Интернет. В то время, как традиционные инфраструктуры безопасности *GRID* фокусируются на безопасности уровня сообщений и контроле доступа к межузловому взаимодействию в среде *GRID*, в этом проекте рассмотрены межузловые угрозы безопасности, свойственные для динамического и изменяющегося сообщества *GRID* [11]. Чтобы противостоять угрозам, являющимся результатом динамических возможностей *GRID* в области промежуточного программного обеспечения, представлен ряд новых изоляционных технологий служебного уровня. В проекте представляет новый модельно-ориентированный подход к разработке служб *GRID* и поддержке раз-

работчика и администратора в интегрированной среде в области сервис-ориентированных *Grid*-вычислений. Внедрение системы имело положительный эффект от внедрения, способствовало понижению расходов на *GRID*-вычисления и дало возможность удобной для пользователя разработки сложных *GRID*-приложений.

### **Выводы**

В результате определения современной концепции развития *GRID* основным направлением является формирование динамической среды *GRID*, состоящей из взаимодействующих неоднородных вычислительных узлов, без определенной фиксированной инфраструктуры и с минимальными административными требованиями.

Анализ существующих проектов, целью которых является создание динамического вычислительного окружения *GRID*, показал, что интеграция рабочих станций в организациях и разрозненных персональных компьютеров в динамические *GRID* является актуальной задачей. Одной из проблем представленных проектов является реализация динамического вычислительного окружения за счет специализированных изолированных *GRID* с отсутствием интеграции с существующей *GRID*-средой.

В статье приведен обзор проблем, которые возникают при реализации динамической среды *GRID*. Представлен ряд требований для реализации динамической *GRID*, которая рассматривается в контексте настольных *GRID*, «*peer-to-peer*» *GRID*, динамических *GRID*. В статье так же рассмотрены особенности реализации мобильных *GRID*. Проведенное исследование показало, что характеристики и особенности динамических *GRID* и мобильных *GRID* по своей природе сходны, имеют ряд особенностей и проблем, которые обуславливают необходимость разработки новых стратегий управления и механизмов планирования которые бы адаптировались к динамической вычислительной среде и обладали свойствами ин-

теграції в существуюче GRID-пространство, інтеперабельности, мінімізації сложности и управління, простоты разработки программного обеспечения для специалистов и доступности для експертів в различных предметных областях.

### Список литературы

1. Avilés-López E., García-Macías J.A. Providing Service-Oriented Abstractions for the Wireless Sensor Grid. – GPC, 2007. – P. 710–715.
2. Berman F., Fox G., Hey T. Grid Computing: Making the Global Infrastructure a Reality. – Chichester: John Wiley & Sons, 2003. – 1060 p.
3. Box D., Ehnebuske D., Kakivay G. et al. Simple Object Access Protocol (SOAP) – 2003. – <http://www.w3.org/TR/soap/>.
4. Chang K., Dasari A., Madduri H. et al. Design of amenable ment process for on demand applications. IBM Systems Journal. – 2004. – № 43(1). – P. 190–203.
5. Christensen E., Curbera F., Meredith G., Weerawarana S. Web Services Description Language (WSDL). – 2001. – <http://www.w3.org/TR/wsdl>.
6. Foster I, Kesselman C., Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organization // Int. Journal of High Performance Computing Application. – 2001. – № 15 (3). – P. 200–222.
7. Foster I., Iamnitchi A. On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing // Proc. of the 2<sup>th</sup> Int. Workshop on Peer-to-Peer Systems. – 2003. – P. 118–128.
8. Foster I., Kesselman C. The Grid2: Blueprint for a New Computing Infrastructure. Elsever, 2003. – 777 p.
9. Foster I., Kesselman C., Nick J., Tuecke S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration // In Open Grid Service Infrastructure WG, Global Grid Forum. – 2002. – P. 131.
10. Foster, I., Kishimoto, H., Savva et al. The Open Grid Services Architecture, Version 1.5. – Open GRID Forum, 2006.
11. Friese T., Smith M., Freisleben B. Hot Service Deployment in an Ad Hoc Grid Environment // Proc. of the 2<sup>th</sup> Int. Conf. on Service-Oriented Computing. – New York, USA, 2004. – P. 75–83.
12. Germain C., Breton V., Clarysse P. et al. Grid-Enabling Medical Image Analysis // Proc. of the Bio Grid Workshop at CC Grid. – Cardi, UK, 2005. – P. 487–495.
13. Glatard T., Montagnat J., Pennec X. Grid-Enabled Workflows for Data Intensive Applications // Proc. of the 18<sup>th</sup> IEEE Symposium on Computer-Based Medical Systems. – Dublin, Ireland, 2005. – P. 537–542.
14. Korpela E., Werthimer D., Anderson D. et al. SETI@home-Massively Distributed Computing for SETI // Computing in Science and Engineering. – № 3(1), 2001. – P. 79.
15. Lohr S. IBM Making a Commitment to the Next Phase of the Internet. – New York Times, 2001. – <http://www.nytimes.com/2001/08/02/technology/02BLUE.html>.
16. Naik V., Sivasubramanian S., Bantz D., Krishnan S. Harmony: A Desktop Grid for Delivering Enterprise Computations // Proc. 4<sup>th</sup> Int. Workshop on Grid Computing. – 2003. – P. 25–33.
17. Rappa M.A. The Utility Business Model and the Future of Computing Services // IBM Systems Journal. – № 43(1), 2004. – P. 32–42.
18. Shiple S., Siegel H., Maciejewski A. et al. Mapping of Subtasks with Multiple Versions in a Heterogeneous AdHoc Grid Environment. Algorithms // Models and Tools for Parallel Computing on Heterogeneous Networks. – 2004. – P. 380–387.
19. Smith M., Friese T., Freisleben B. Towards a Service-Oriented AdHoc Grid // Proc. of the 3<sup>rd</sup> Int. Symp. Parallel and Distributed Computing. – Cork, Ireland, 2004. – P. 201–209.
20. Sterck H.D., Markel R., Knight R. A Light weight, Scalable Grid Computing Framework for Parallel Bioinformatics Applications // Proc. of the 19<sup>th</sup> Int. Symp. on High Performance Computing Systems and Applications. – 2005. – P. 251–257.