

Операционная система Plan9 как реализация идеологии GRID

**М. Н. Геворкян^{1,a}, И. М. Гостев^{2,b}, А. В. Королькова^{1,c},
Д. С. Кулябов^{1,3,d}, Л. А. Севастьянов^{1,4,e}**

¹Российский университет дружбы народов,
Россия, 117198, г. Москва, ул. Миклухо-Макляя, д. 6

²Национальный исследовательский университет «Высшая школа экономики»,
Россия, 101000, г. Москва, ул. Мясницкая, д. 20

³Лаборатория информационных технологий, Объединенный институт ядерных исследований,
Россия, 141980, г. Дубна, Московская область, ул. Жолио-Кюри, д. 6

⁴Лаборатория теоретической физики, Объединенный институт ядерных исследований,
Россия, 141980, г. Дубна, Московская область, ул. Жолио-Кюри, д. 6

E-mail: ^a mngevorkeyan@sci.pfu.edu.ru, ^b igostev@gmail.com, ^c akorolkova@sci.pfu.edu.ru,
^d yamadharm@gmail.com, ^e leonid.sevast@gmail.com

При проведении параллельных вычислений на кластерной системе структура вычислительной система важно для пользователя и должна быть принята во внимание при написании параллельных программ. GRID как идеология вводит дополнительный уровень абстракции и позволяет связать воедино разнородные вычислительные системы. На самом деле, неспособность контролировать операционную среду заставляет разработчиков создавать лишнюю промежуточную инфраструктуру на уровне приложений. Предлагается спуститься ниже по уровню абстракции и интегрировать необходимую функциональность в ядро операционной среды. Например, операционная система Plan9 имеет имманентную структуру, необходимую для внедрения идеологии GRID в распределенные вычисления. При этом основные архитектурные элементы операционной система Plan9 позволяют использовать все ресурсы удаленного компьютера в качестве локальных ресурсов.

Ключевые слова: GRID, Plan9, распределенные вычисления

Работа частично поддержана грантами РФФИ No 14-01-00628, 15-07-08795 и 16-07-00556. Также публикация выполнена при финансовой поддержке Минобрнауки России (Соглашение No 02.а03.21.0008).

© 2016 Мигран Нельсонович Геворкян, Иван Михайлович Гостев, Анна Владиславовна Королькова,
Дмитрий Сергеевич Кулябов, Леонид Антонович Севастьянов

1. Введение

Правильно выбранная идеология имеет крайне большое значение при создании научных и инженерных объектов. Неправильно выбранная идеология (на любом этапе развития, от замысла до реконструкции) приводит к появлению ситуативных решений. Ситуативное решение кажется оптимальным на определенном этапе, но в дальнейшем оно может ухудшить ситуацию.

При появлении вызова редко идут путем усовершенствования и углубления идеологии, предпочитая строить дополнительную систему. В результате получается тяжеловесная конструкция, которая в конце концов разрушается под тяжестью наложенных заплаток.

Примеров можно привести множество. Возьмем, например, семейство операционных систем UNIX и стек протоколов TCP/IP. В обоих случаях развитие шло одинаково: вначале улучшения шли путем добавления новых свойств, далее обе системы стали тяжеловесными и трудно поддерживаемыми. Обе системы пришли к необходимости пересмотра и углубления идеологии (Plan9 и IPv6 соответственно). И в обоих случаях старые системы не только остались живы (по крайней мере пока), но и пытаются ассимилировать новые идеологии.

Примером успешного ситуативного решения можно считать GRID. В качестве идеологической альтернативы ему можно привести Plan9. В данной статье мы хотим тезисно провести параллель между GRID и Plan9.

2. Идеология GRID

Прежде всего, рассмотрим идеологию систем GRID. GRID разрабатывался как способ простого доступа к вычислительным ресурсам [The Grid:..., 1999]. Предшественниками GRID можно считать системы распределенных добровольных вычислений, работающих по принципу SPMD (одна программа, много данных). На данный момент эта идеология представлена в основном платформой BOINC [Karimi, Dickson, ..., 2010].

Вызовом, подстегнувшим появление GRID, стала необходимость обработки данных в рамках проекта LHC (большой адронный коллайдер) [Кореньков, Тихоненко, 2001]. В рамках этой программы выкристаллизовались и основные компоненты GRID-систем [Foster, Kesselman, ..., 2002; Jacob, Brown, ..., 2006].

- х Поиск и мониторинг ресурсов.
- х Выделение ресурсов.
- х Коммуникация.
- х Аутентификация.
- х Доступ к данным.

Однако все эти компоненты реализуются поверх уже существующих вычислительных сред. Фактически, GRID является промежуточным программным обеспечением (middleware):

- х GRID является посредником между подзадачами, выполняемыми на удаленных друг от друга компьютерах. Доступ к вычислительным ресурсам, данным, устройствам является прозрачным и виртуальным (предоставляется доступ не к компьютерам, а к сервисам).
- х GRID представляет собой набор программных инструментов для обмена сообщениями, вызова удаленных процедур, управления доступом к ресурсам на уровне сетей.
- х GRID создает единую виртуальную среду для выполнения распределенного приложения. Эта среда не должна зависеть от сетевых служб, аппаратных платформ, операционных сред и географической удаленности компьютеров.

На данный момент существует несколько реализаций идеологии GRID.

- х • Globus Toolkit [<http://www.globus.org/>] — является стандартом де-факто на реализацию GRID-систем. Достаточно тяжеловесный и сложный в развертывании и обслуживании.

- х • gLite [<http://glite.cern.ch/>]. Являлся более простой и легковесной альтернативой Globus Toolkit. На настоящий момент полностью влился в проект EMI.
- х • EMI [<http://www.eu-emi.eu>] (European Middleware Initiative, Европейская инициатива по развитию промежуточного программного обеспечения). Данный проект объединил несколько других проектов: gLite, ARC, UNICORE, dCache.

3. Идеология Plan9

Две диалектические сущности идеологии UNIX — файлы и процессы. Однако, по историческим причинам, эти сущности не являлись тотальными. Как результат, возникла необходимость в ситуативных дополнениях, вызывавших коррозию идеологии UNIX. Система разрасталась, становилась слишком сложной, плохо поддерживаемой. Plan9 представляет собой попытку вернуться к чистоте идеологии UNIX [Pike, Presotto, ..., 1990; Mullender, Presotto, 2001].

Plan9 базируется на крайне ограниченном количестве принципов [Mirtchovski, Simmonds, ..., 2004].

- х Все ресурсы являются файлами (тотальность понятия файла).
- х Все файлы доступны в рамках единой иерархической файловой системы.
- х Для доступа к файлам используется стандартный протокол 9P.
- х Локальные и удаленные ресурсы не различаются.
- х Процессы обладают пространством имен, основанном на файловой иерархии.

3.1. Тотальность понятия файла

Понятие файла действительно становится тотальным. В классической системе Unix большинство сущностей представлено файлами, в том числе и устройства (иерархия /dev). Однако, например, сетевые интерфейсы представляют из себя отдельную сущность. Не говоря уже об оконной системе X Window.

В операционной системе Plan9 все является файлом, например:

- х • /dev/cons — содержимое консоли;
- х • /prog — программы (вне зависимости от того, на каком хосте они выполняются);
- х • доступ к удаленным хостам по разным сетевым протоколам, например ftp или http;
- х • разрешение доменных имен

```
% echo www.google.com!http > /net/dns
```
- х • использование удаленной машины для соединения (например, снимается необходимость в отдельном механизме NAT)

```
% import otherhost /net
% telnet tcp!www.google.com
```
- х • архивы и прочие контейнеры;
- х • оконная система Plan9;
- х • сетевые интерфейсы.

Здесь можно заметить, что современные UNIX, например Linux, активно заимствует элементы файловой тотальности из Plan9 (например, иерархия /prog, файловые системы для работы с архивами и контейнерами). Однако все это реализуется дополнительным внешним инструментарием.

Для прозрачности исполнения процессов в распределенной среде важное значение имеет понятие пространства имен. UNIX поддерживает глобальное пространство имен, что приводит к низкой масштабируемости в распределенных системах. Plan9 же предоставляет для каждого процесса локальное пространство имен.

Для управления файлами служит стандартизованный механизм, протокол 9P (текущая версия — 9P2000). Фактически он создает распределенную файловую систему в рамках операци-

онной системы. Причем он спроектирован в крайне минималистичной манере: он содержит всего 17 сообщений: 3 для аутентификации, 14 для совершения действий с файлами.

Распределенная аутентификация является составной частью операционной системы Plan9. Такой подход к аутентификации позволил отказаться от понятия суперпользователя.

Интерфейс пользователя в Plan9 тоже претерпел идеологические изменения. Командная строка — одна из сильнейших сторон UNIX. Поэтому UNIX-подобный командный интерфейс был сохранен в Plan9. Однако интерфейс стал мышечентричный. Но это не интерфейс WIMP (Window-Icon-Mouse-Pointer). Это текстовый графический интерфейс, подобный Оберону Н. Вирта. Кроме того, было введено понятие плюмбинга, расширяющее понятие каналов на графическое окружение.

3.2. Варианты Plan9

При отсутствии основного разработчика, у операционной системы Plan9 возникло несколько ответвлений (форков), большинство из которых не демонстрирует признаков жизни.

- × Plan 9 from Bell Labs [<http://plan9.bell-labs.com/plan9/index.html>]. Исходный вариант системы. На данный момент представляет исторический интерес.
- × Plan 9 from User Space [<https://swtch.com/plan9port/>] — попытка переноса утилит Plan9 на UNIX.
- × 9legacy [<http://www.9legacy.org/>]. Набор патчей к Plan 9 from Bell Labs.
- × 9front [<http://9front.org/>] — интересное и достаточно бурно развивающееся ответвление Plan9.
- × NIX [<http://lsub.org/ls/nix.html>] — переработанное ядро Plan9 для целей облачных вычислений.
- × 9atom [<http://www.quanstro.net/plan9/9atom/>]. Экспериментальное ответвление без большого сообщества. Содержит в том числе и ядро от NIX.
- × Harvey [<http://harvey-os.org/>]. Plan 9 с поддержкой gcc и clang.
- × Inferno [<http://www.inferno-os.info/inferno/>]. Отдельная операционная система, разработанная в рамках идеологии Plan9. Основная особенность — исполнение кода в виртуальной машине.

4. Заключение

Авторам статьи представляется, что идеология операционной системы Plan9 полностью соответствует идеологии GRID, и даже превосходит ее. В систему Plan9 изначально встроены средства распределенной коммуникации, аутентификации, доступа к распределенным вычислительным ресурсам и к распределенным данным. Конечно, замена инфраструктуры GRID с помощью Plan9 представляется на данный момент утопией. Однако авторы считают, что расширение идеологии и инструментария GRID за счет идеологии и инструментария Plan9 может принести радикальные улучшения.

Список литературы

- Korenkov V. V., Tikhonenko E. A.* The concept of GRID and computer technology in the LHC era // Physics of Elementary Particles and Atomic Nuclei (PEPAN). — 2001. — V. 32, № 6. — PP. 1458–1493, [in Russian].
- Foster I., Kesselman C., Nick J. M., Tuecke S.* The physiology of the grid: An open grid services architecture for distributed systems integration // Globus Project. — 2002. — P. 31.
- Jacob B., Brown M., Fukui K., Trivedi N.* Introduction to Grid Computing // USA: IBM Red Books International Technical Support Organization. — 2006. — P. 352.

- Karimi K., Dickson N. G., Hamze F.* High-Performance Physics Simulations Using Multi-Core CPUs and GPGPUs in a Volunteer Computing Context // *The International Journal of High Performance Computing Applications*. — 2010. — Vol. 25, no. 1. — P. 61–69.
- Mirtchovski A., Simmonds R., Minnich R.* Plan 9 - an integrated approach to grid computing // *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings*. — 2004. — P. 273–280.
- Mullender S. J., Presotto D.* Programming Distributed Applications using Plan 9 from Bell Labs // *4th European Research Seminar on Advances in Distributed Systems (ERSADS), Bertinoro, Italy*. — Bertinoro: Univ. di Bologna, 2001. — P. 115–132.
- Pike R., Presotto D., Thompson K., et al.* Plan 9 from Bell Labs. — 1990. — P. 1–9.
- The Grid: Blueprint for a New Computing Infrastructure, Foster I., Kesselman C. (ed.)*. The Elsevier Series in Grid Computing. — Morgan Kaufmann, 1999. — P. 675.

Operating system Plan9 as the implementation of the GRID ideology

**M. N. Gevorkyan^{1,a}, I. M. Gostev^{2,b}, A. V. Korolkova^{1,c}, D. S. Kulyabov^{1,3,d},
L. A. Sevastianov^{1,4,e}**

¹RUDN University (Peoples' Friendship University of Russia),
6 Miklukho-Maklaya street, Moscow, 117198, Russia

²National Research University Higher School of Economics,
20 Myasnitskaya street, Moscow, 101000, Russia

³Laboratory of Information Technologies, Joint Institute for Nuclear Research,
6 Joliot-Curie street, Dubna, Moscow region, 141980, Russia

⁴Bogoliubov Laboratory of Theoretical Physics, Joint Institute for Nuclear Research,
6 Joliot-Curie street, Dubna, Moscow region, 141980, Russia

E-mail: ^amngevorkyan@sci.pfu.edu.ru, ^bigostev@gmail.com, ^cakorolkova@sci.pfu.edu.ru,
^dyamadharma@gmail.com, ^eleonid.sevast@gmail.com

When we organize parallel computations on a cluster system, the computer system structure is not hidden from the user, and should be taken into account while writing parallel programs. GRID ideology introduces an additional level of abstraction and makes it possible to link together heterogeneous computing systems. In fact, the inability to control the operating environment makes developers to create superfluous infrastructure at the application level. We offer to go down to the below level and to implement the necessary functionality within the operating environment kernel. Plan9 operating system has the immanent structure necessary for the implementation of GRID ideology. The main architectural elements of the operating system allow you to use all the resources of the remote computer as the local resources.

Keywords: GRID, Plan9, distributed computing

The work is partially supported by RFBR grants No's 14-01-00628, 15-07-08795, and 16-07-00556. Also the publication was supported by the Ministry of Education and Science of the Russian Federation (the Agreement No 02.a03.21.0008).