**BC/NW 2011; №2 (19):10.4**

**АНАЛИЗ ПОДХОДОВ К ПРОЕКТИРОВАНИЮ КОРПОРАТИВНОГО КОМПЛЕКСА УПРАВЛЕНИЯ ТЕРМИНАЛЬНЫМ ДОСТУПОМ**

Артюхов О.И., Окошкин А.В.

(Москва, НИУ Московский энергетический институт, Россия)

Известно, что «всё новое – это хорошее забытое старое». Изначально, ещё до появления персональных компьютеров в основу организации практически любого корпоративного приложения было заложено использование централизованной архитектуры, предполагающей наличие т.н. мэйнфрейма, (возможно) ряда мини-ЭВМ, и «безинтеллектуальных» внешних устройств-терминалов. При очевидных недостатках, такой подход имел и ряд преимуществ – централизация вычислительных ресурсов, простота и относительная дешевизна эксплуатации [1]. С развитием ПК, следуя известному закону диалектики, разработчики пошли по пути отказа (отрицания) от централизации и максимальной персонализации приложений, «переложив» их на универсальные и, по тем временам, достаточно мощные ПК. Однако, спустя некоторое и достаточно короткое время в качестве ограничений уже этого подхода выступили проблемы нехватки мощностей отдельных компьютеров для стремительно развивающихся приложений, защиты данных при неизбежном взаимодействии последних и, конечно, лицензирования самих приложений при их коммерческом использовании. Позитивным и знаковым моментом этого периода развития явилось появление архитектуры «клиент-сервер», которая, хотя и решила многие задачи, но не сняла проблемы в целом. И «спираль» диалектики развития совершила очередной виток (отрицание отрицания) – возврат, естественно на новом уровне, к централизации.

В настоящее время применяемые, как на ПК, так и на специализированных рабочих станциях способы предоставления терминального доступа используют очень схожие с централизованной архитектурой принципы, основанные на переносе громоздких вычислений и хранения основных данных опять на сторону центра – сервера. При этом типовой механизм работы всего приложения выглядит следующим образом: пользователь запускает на «своей стороне» некую клиентскую часть удалённого доступа и при помощи неё подсоединяется к серверу; после успешной аутентификации он получает возможность запускать в адресном пространстве сервера нужные ему приложения; со стороны клиента на сервер отправляются данные от устройств ввода информации (коды нажатых на клавиатуре клавиш, координаты перемещения позиционного локатора); в обратную сторону поступают данные для изменения текущего вида пользовательского интерфейса приложения («картинки» на мониторе). В рамках данного подхода пользователь ПК зачастую поставлен перед необходимостью самостоятельной настройки клиентской части терминального доступа и обеспечением хранения некоторой локальной информации, т.е. ему необходимо не только частично администрировать сам сервер приложений, но и производить полное локальное администрирование. Одним из общепринятых подходов к решению данных проблем в контексте централизованного управления клиентскими станциями является использование доменов Active Directory [2].

В 2008 году Google начал активно повсеместно внедрять термин «облачных вычислений» (cloud computing) для уже существовавшего тогда подхода обеспечения повсеместного и удобного сетевого доступа по требованию к общему набору вычислительных ресурсов. Здесь «облако» является метафорой, основанной на общепринятом изображении Интернета на диаграммах компьютерных сетей как образа сложной инфраструктуры, за которым стоят и **скрываются** все технические детали реализации. Основными характеристиками облачных вычислений принято считать [4]:

* самообслуживание по требованию - пользователь сам определяет и устанавливает необходимые ему вычислительные параметры;
* универсальный доступ по сети - отсутствие зависимости от используемого терминального устройства;
* объединение ресурсов - поставщик услуг прозрачно для клиентов объединяет ресурсы, необходимые для обслуживания большого числа потребителей, в единый пул для динамического перераспределения мощностей между клиентами;
* «эластичность» предоставляемых услуг;
* учёт потребления услуг.

Облачные вычисления можно считать логическим продолжением развития клиент-серверной архитектуры. Самой распространённой моделью обслуживания «в облаках» является концепция SaaS (Software-as-a-Service, программное обеспечение как услуга) [5]. Это эффективный инструмент повышения прибыли и расширения каналов продаж для независимых производителей программного обеспечения, пользователи производят оплату по факту и сами регулируют объём необходимых им ресурсов в зависимости от текущих потребностей. В данном случае облака могут быть как платформенные (Microsoft Azure), так и независящие от ОС (например, такие различные веб-ресурсы, как почта или веб-графические редакторы). Также плюсом облачных вычислений является отказоустойчивость при прозрачном доступе к услугам, ресурсам, так как обычно в этом случае серверы объединяют в кластер. Таким образом, в облачных вычислениях пользователь платит только за услугу, его не интересуют и никак не затрагивают технические особенности реализации сервиса. Зачастую при организации корпоративных информационных систем выдвигается требование использования защищённых бездисковых рабочих станций. Также важными являются экономические аспекты эксплуатации автоматизированных систем, удобство и простота администрирования рабочих мест.

Учитывая всё сказанное выше, концепция облачных вычислений в сочетании с известными клиент-серверными механизмами предлагается в качестве архитектуры для проектирования комплексного системного решения «Ками-Терминал». Данный комплекс предназначен для создания защищённых автоматизированных систем и для обеспечения эффективного и безопасного доступа к сетевым информационным ресурсам за счёт переноса вычислений на сторону сервера. В структуре комплекса можно выделить 3 основных элемента: терминальное устройство (попросту, терминал), терминальный сервер и сервер приложений. Принцип их взаимодействия схематично представлен на рис.1.

Сервером приложений может быть любой выделенный сервер, предоставляющий услуги и приложения, необходимые для работы операторам терминалов. Именно на нём запускаются серверные части классических терминальных систем, таких как RDesktop, VNC, Citrix ICA, OpenSSH или X11 [3].

Терминал представляет собой бездисковую и маломощную рабочую станцию, исключающую локальное хранение информации, под управлением специально разработанной «облегчённой» ОС на базе ядра Linux. Таким образом, в качестве терминала может выступать, как специально укомплектованные дешёвые x86 или arm совместимые платформы, так и любой «старенький ПК со склада».

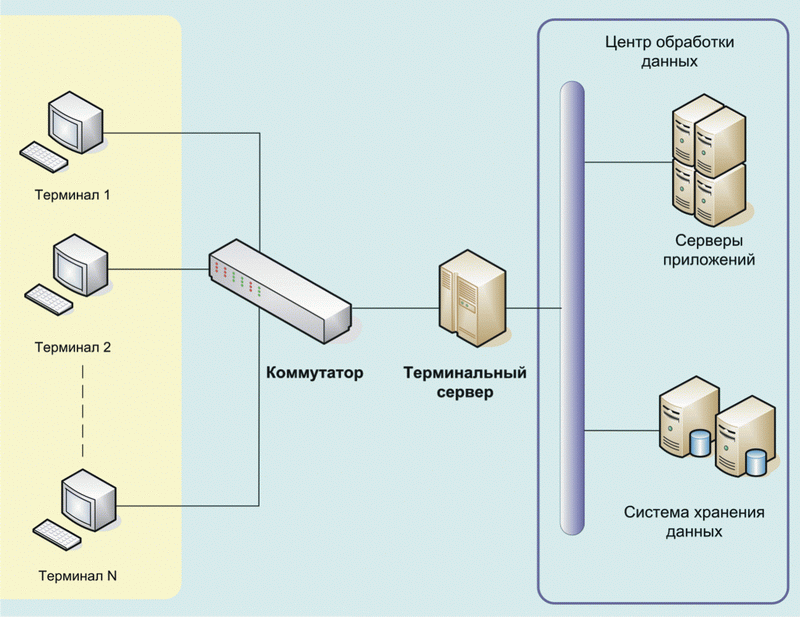


Рис. 1. Структура разрабатываемого комплекса "Ками-Терминал"

Рассмотрим основные (планируемые) принципы работы комплекса. На терминальном сервере запущен Kami-Terminal Server – кроссплатформенный комплекс специальных системных сервисов, обслуживающих парк терминалов. Он обеспечивает терминалы настройками и дополнительными бинарными утилитами, необходимыми для безопасной организации терминального доступа к серверу приложений. Все пользовательские настройки, включая перечень так называемых разрешённых пользователю функций, хранятся на терминальном сервере. В классической централизованной терминальной архитектуре доступа данный элемент отсутствовал, так как пользователь сам производил настройку своего клиента удалённого доступа на конкретной рабочей станции. Разрешённые пользователю функции – это абстрактные понятия, отвечающие за предоставляемый конкретному пользователю конкретный перечень возможных для него действий на терминале без привязки к конкретному рабочему месту. К примеру, если пользователю разрешена функция использования внешних носителей данных, то на терминал будут дополнительно загружены необходимые системные настройки и бинарные драйверы строго разрешённых устройств. Такое поведение обосновано требованиями безопасности и тем, что изначально присутствующая на терминале ОС должна уметь только показать пользователю диалог аутентификации (как паролем, так и с использованием Smart Card) в корпоративной среде и предоставить выбор из разрешённых ему удалённых сессий на строго определённых серверах приложений. Таким образом, от пользователя требуется только ввести логин и пароль (или, приложив свой личный аппаратный идентификатор, ввести PIN-код), выбрать необходимые терминальные сессии и приступить к работе. Каждая клиентская сессия запускается на отдельной виртуальной консоли (в UNIX-терминологии) терминала, количество же одновременно запущенных на терминале удалённых сессий зависит только от производительности сетевого оборудования.

Вместо локального администрирования каждого рабочего места (как это было в классической терминальной технологии) администратор централизованно управляет всем парком терминалов через терминальный сервер. Очевидно, что терминальный сервер чем-то идеологически напоминает MS Active Directory. Но, в отличие от последнего, Kami-Terminal Server ориентирован на автоматизированное или полуавтоматическое администрирование предоставляемого именно каждому пользователю терминального доступа к серверу приложений. Администратор просто регистрирует учётные записи пользователей, указывая каждой разрешённые функции и доступные сессии для серверов приложений. Для пользователя остаётся прозрачным получение доступа к ресурсам и приложениям, не зависимо от выбранного им рабочего места.

В итоге, можно выделить следующие аспекты разрабатываемого комплекса:

*безопасность:*

* исключение варианта хранения данных на клиентском рабочем месте и возможности несанкционированного использования на терминале самостоятельно подключённых пользователем дисководов и периферийных устройств для ввода-вывода информации (внутренние и внешние дисководы, принтеры, USB-устройства и накопители), это обеспечивается за счёт того, что терминальное устройство не содержит встроенных носителей данных, и загружаемая на терминале ОС не поддерживает их на программном уровне, а использование различных USB-накопителей определяется политикой безопасности;
* исключение возможности по установке программных средств с клиентского рабочего места, что позволяет снизить риски нарушения целостности информации от воздействия вредоносных и деструктивных программ «внутри» вычислительной сети.

*управляемость:*

* перенос функционала клиентского рабочего места на серверную сторону, что позволяет обеспечить централизованное управление информационными ресурсами, системой защиты информации и антивирусной защиты, практически исключить постоянные затраты сил и средств на локальное администрирование рабочих мест;
* предоставление пользователю терминального устройства возможности одновременного запуска Windows, Linux сессий с переключением между ними в процессе работы, каждая сессия запускается на отдельной виртуальной консоли.

*экономические аспекты:*

* больше производительности при минимальных затратах – стоимость терминала ниже стоимости полноценной рабочей станции, при этом пользователю будет доступна вычислительная мощность сервера приложений, к которому эти терминалы и подключаются;
* более эффективное в затратном смысле лицензирование ПО, т.к. за основу взяты open source решения;
* снижение затрат на последующую модернизацию корпоративной информационной системы – достаточно модернизировать только серверы приложений.

В данный момент проводится работа по реализации комплекса в соответствии с описанными принципами, а также изучение направлений его дальнейшего развития и усовершенствования. Последние предполагают добавление поддержки новых технологий терминального доступа, создание более гибкой системы доступных для использования пользователями функций, разработку подсистемы мониторинга действий пользователей и состояний терминалов. Интересными представляются аспекты адаптации комплекса под arm архитектуру.

**Литература**

1. Эбберс М. Введение в современные мейнфреймы: основы z/OS – M.:IBM,2007. – 635c.
2. Дэн Холме, Нельсон Рэст Настройка Active Directory. Windows Server 2008. Официальные издания Microsoft Press – Изд-во «Русская Редакция»,2009. -960с.
3. Елманова Н. Терминальный доступ для малых и средних предприятий. – КомпьютерПресс, 2005 №11.
4. Владимир Романченко. Облачные вычисления на каждый день. – 06.09.2009. http://www.3dnews.ru
5. Клементьев И.П., Устинов В.А. Введение в Облачные вычисления – Екатеринбург:Изд-во УГУ, 2007. -233с