**Глава 3. Частные вопросы управления ИТ-инфраструктурой**

**Примеры инфраструктурных решений, применяющихся в крупных сетевых проектах**

* 1. **Пример реализации инфраструктуры в Google**

Google визуализирует свою инфраструктуру в виде трехслойного стека:

* Продукты: поиск, реклама, электронная почта, карты, видео, чат, блоги и т.п.
* Распределенная инфраструктура системы: GFS, MapReduce и BigTable.
* Вычислительные платформы: множество компьютеров во множестве дата центров.

**GFS (Google File System)** — большая распределенная файловая система, способная хранить и обрабатывать огромные объемы информации. Она является основной платформой хранения информации в Google.

GFS строилась исходя из следующим критериев:

* Система строится из большого количества обыкновенного недорогого оборудования, которое часто дает сбои. Должны существовать мониторинги сбоев и возможность в случае отказа какого-либо оборудования восстановить функционирование системы.
* Система должна хранить много больших файлов.
* Приложения, критические по времени исполнения, должны накопить определенное количество таких запросов и отсортировать их по смещению от начала файла. Это позволит избежать при чтении блужданий вида назад-вперед.
* Часто встречаются операции записи большого последовательного куска данных, который необходимо дописать в файл.
* Система должна реализовывать строго очерченную семантику параллельной работы нескольких клиентов, в случае если они одновременно пытаются дописать данные в один и тот же файл.
* Высокая пропускная способность является более предпочтительной, чем маленькая задержка.

Файлы в GFS организованы иерархически, при помощи каталогов, как и в любой другой файловой системе, и идентифицируются своим путем.

GFS поддерживает резервные копии, или снимки (snapshot).

* 1. **Пример реализации инфраструктуры для проекта Flickr**

Примечательно, что на проекте Flickr используется практически только свободное программное обеспечение.

**Использующиеся программные компоненты**

* Платформа GNU/Linux (RedHat)
* СУБД MySQL
* Web-сервер Apache
* Скрипты программной логики, написанные на языке PHP и Perl
* Средства сегментирования (Shards) (прим.: разбиение системы на части, обслуживающие каждая свою группу пользователей)
* Memcached для кэширования часто востребованного контента
* Squid в качестве обратного прокси-сервера для html-страниц и изображений
* Шаблонизатор Smarty
* PEAR для парсинга e-mail и XML
* ImageMagick для обработки изображений
* SystemImager для развертывания элементов конфигурации
* Ganglia для мониторинга распределенных систем
* Subcon для хранения важных системных конфигурационных файлов в SVN-репозитории для легкого развертывания на машины в кластере
* Cvsup для распространения и обновления коллекций файлов по сети.

**Типовое оборудование для серверов**

* EMT64 под управлением RHEL 4 с 16 Gb оперативной памяти.
* 6 жестких дисков с 15000rpm, объединены в RAID-10.
* Размер для пользовательских метаданных достигает 12 терабайт (это не включает фотографии).
* Используются 2U корпуса.

**Системная архитектура**

* Входные запросы поступают на сдублированные контроллеры приложений Brocade ServerIron ADX. Они обеспечивают коммутацию приложений и балансировку трафика, основываясь на принципе виртуальных ферм серверов
* Выбор “лучшего” сервера производится на основании механизма Real-Time Health и наличия требуемой производительности
* Последовательно повышается коэффициент использования для всех серверов
* Интеллектуальное распределение загрузки осуществляется для всех доступных ресурсов
* Метод конфигурируется и выбирается пользователем
* Обеспечивается защита серверной фермы от атак и от неправильной эксплуатации
* Клиенты подсоединяются к серверам приложений используя виртуальный IP (VIP). VIP адреса настраиваются на коммутаторе приложений
* Коммутатор приложений осуществляет трансляцию адресов после выбора нужного сервера, причем сами адреса серверов скрыты.Обслуживание сессий ведется согласно последовательности:Запись о каждой пользовательской сессии создается в таблице
* Каждая сессия назначается определенному серверу
* Все сообщения в рамках сессии посылаются к одному серверу
* Таблицы сессий синхронизируются между двумя коммутаторами
* За счет дублирования коммутаторов нет простоя сервиса когда коммутатор вышел из строя: второй коммутатор обнаруживает отказ и начинает обслуживать сессии пользователей

Структура Dual Tree является индивидуальным набором модификаций для MySQL, позволяющим масштабировать систему путем добавления новых мастер-серверов без использования кольцевой архитектуры.

Центральная база данных включает в себя таблицу пользователей, состоящую из основных ключей пользователей и указатель на сегмент, на котором может быть найдена остальная информация о конкретном пользователе.

Все, за исключением фотографий, хранится в базе данных.