



Как облака изменяют Linux?

Peter Lemenkov

Russian Fedora / Fedora Project.

Как было раньше?

Типичная система



- Заботливо “скомпилирована из сорцов”
- Один или несколько машин (см. “админ локалхоста”).
- Не перезагружается годами (см. “большой аптайм” -> см. “красивый скриншот” -> см. “почет и уважение на линуксовке”)
- Много “магии” (собранные вручную, незадокументированные конфиги и скрипты)
- Девушки нет, но зато машина названа анимешным, фэнтезийным или взятым из sci-fi сериалов именем

Внезапно, проблемы!

- Попробуй уволить админа, злодей!
- Да он и сам уже ничего не помнит.
- Долгий аптайм == (почти всегда) проблемы в безопасности
- Долгий аптайм == (почти наверняка) признак, что за системой никто не присматривает и не обновляет
- Попробуй обновить какое-то конкретное ПО, не обновляя ничего больше.
- ...

Большой аптайм!



- Большой аптайм – скрытая угроза для бизнеса/организации.
- Если у вас есть компьютер, который вот **прямо сейчас** нельзя выключить, то это мина с часовым механизмом.
- “Если для бизнеса падение ...сервера фатально, то надо сразу выходить из бизнеса”. Max Lapshin, CTO at Flussonic, LLC.

Поэтому систем с большим аптаймом
надо избегать, но как?

Как исправить?

- Избыточность ресурсов (балансировка по серверам, кластеры)
- Возможность замены на лету (“hot swap”, “hot code reload”)
- Бесшовные и надежные обновления

ТАК ЭТО ЖЕ ОБЛАКА!

Облака, это...

- Балансировка ресурсов
- Возможность замены на лету (облако находится “над железом”)
- Обновления (почти) без остановки

“Pets and cattle”

- У системы имени больше нет – теперь это просто бездушный рабочий юнит с ID
- Не “удобство в настройке под себя” и “богатый набор USE-флагов”, а быстрое развертывание систем с предсказуемыми свойствами из повторно используемых компонентов (бинарных пакетов).

А что нужно?

- Загружаться как можно быстрее
- Контроль над процессами
- ~~Надежные обновления (почти) без~~
~~остановки~~
 - Ок, хотя бы обновления без поломок и с возможностью “отката”, когда что-нибудь ломается

Как это решают?

- Загружаться как можно быстрее – `systemd` / контейнеры (LXC, OpenVZ)
- Контроль над процессами – `cgroups` / `systemd`
- Обновления без поломок – пересоздание `rootfs` на каждый минимальный апдейт.

Почему systemd?



- Быстро загружает (загрузка контейнера с Arch на Fedora 19 занимает 50 миллисекунд до ssh без DNSР)
- Контроль над процессами – зависимости, статус, ограничение ресурсов с помощью cgroups
- Небывалая доселе прозрачность и уровень мониторинга

Почему контейнеры?

- Компромисс между изоляцией и расходом ресурсов
- ~~Оверселлинг, это выгодно!~~

А как обновляться?



- Пересоздавать rootfs каждый раз, при апдейтах
- Запускай новый рабочий юнит с новой rootfs
- Если все ок, то убивай старый юнит
- Это скотный двор, и любимцев тут нет

Как это повлияло
на дистрибутивы?

Переход на systemd

- Унификация с облачными системами
- Упрощение жизни для разработчиков встраиваемых систем
- Полный контроль над работой системы
- “Конкуренты” начали копипастить идеи. И это здорово!

Контейнеры и VM

- Гостевая ОС это такой-же процесс (или группы процессов), как и прочие
- Контейнер/KVM - один из процессов.
- А почему бы не поместить другие процессы в контейнеры / VM?
 - Qubes OS (Joanna Rutkowska), app within VM
 - Docker
 - systemd-nspawn

Контейнеры в DE



- Контейнеры будут использоваться прямо для запуска критических приложений (Firefox, Google Chrome)
- Контроль (и, следовательно, интеграция) с помощью systemd (systemd-nspawn)

Как обновляться?

- Don't upgrade, redeploy!
- Обновления должны быть надежны. Поэтому транзакционны (либо туда, либо сюда)
 - Атомарность (либо получилось, либо нет)
 - Консистентность (rollback при ошибках)
 - Изолированность (параллельность обновлений)
 - Надежность (выключив питание мы ничего не потеряем – см. “большой аптайм”)

Если по-старому, то...

- Нужна FS со снапшотами и роллбэками
- Нужен пакетный менеджер с транзакциями (ближе всего – rpm)
- Нужно уметь Checkpoint/Restore In Userspace для произвольных системных демонов. Ну ок, хотя-бы для некоторых
- И все-равно придется перезагружаться в общем случае

Если по-новому, то...

- Две rootfs – одна текущая, другая резервная (как вариант, три).
- При апдейте пересоздать резервную fs
- Перезагрузиться используя ее, как основную
- Бывшую основную пометить, как резервную
- Всё

Вот поэтому redeploy

- ChromeOS работает так
- Mac OS X работает ***почти*** так
- Облачные appliances так работают.
- Чем же Linux хуже?

Поэтому теперь при апдейтах будем перезагружаться.

Новый дистрибутив?

- Минимум компонентов – только для запуска самой системы, и для запуска других систем в контейнерах.
- Обновлений нет вообще (ну или почти нет). Если приспичит, то мигрируй всех на соседнюю машину, и переустанавливай.
- Для прикола включим Portage.
- Встречайте - CoreOS

Более новая ОС - OSv

- Не Linux
- Еще более лучше спроектируем
- Удалим “лишний код”, раз у нас супервизор на супервизоре
- Внесем ряд ограничений – несовместимо со стандартами (POSIX)
- W.I.P.

А как же UNIX-Way?

- Был хорош для прототипирования, но сколько можно пробовать?
- A tightly-coupled core vs. a world of many implementations
- Технологический стандарт против философии

Questions?



Contact:
lemenkov@gmail.com