Алгоритмы параллельной обработки пакетов для сборочницы дистрибутива.

Игорь Власенко
viy@altlinux.org

ALT LINUX TEAM

СБОРОЧНИЦА

В начале все делалось руками …

- Rsync → Incoming → заботливые руки.

А потом пришел Дмитрий Левин (ldv@) и все автоматизировал.

Проблема в том, что Дмитрий сделал это слишком хорошо.

Hasher — жемчужина программирования, а

Gear — по дизайну велосипед над hasher – тоже получился простым и удачным.

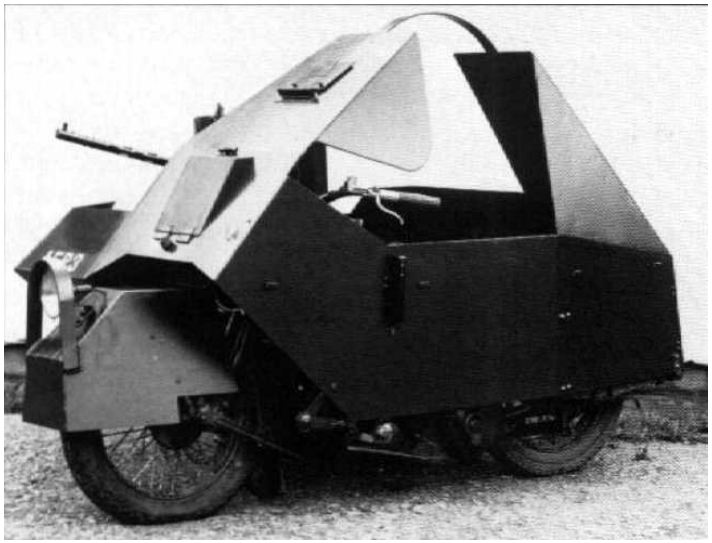
СБОРОЧНИЦА

Gear — в начале простой по дизайну и удобный в применении качественный велосипед над hasher.

СБОРОЧНИЦА

→

→



И поэтому на него стали навешивать все новые и новые функции …

СБОРОЧНИЦА

… пока в монолитном коде не накопились проценты по техническому долгу.

СБОРОЧНИЦА

- Технический долг — метафора программной инженерии, обозначающая накопленные в программном коде или архитектуре проблемы, вызывающие дополнительные затраты труда в будущем. Технический долг связан с недостатками в сопровождаемости, тестируемости, понятности, модифицируемости, переносимости.
  По аналогии с финансовым долгом, технический долг может обрастать «процентами» — усложнением (или даже невозможностью) продолжения разработки, исправления ошибок, сопровождения.

Бьем в набат: Технический долг!

ТРЕВОЖНЫЕ ЗВОНОЧКИ

- Ошибка #39276 – невозможность управлять лимитами hasher для заданий.
- 1. Idle time limit. Пользователи научились обходить

```
--- batik.spec
+++ batik.spec
@@ -270,2 +270,5 @@
 %build
+# zerg's girar armh hack:
+(while true; do date; sleep 7m; done) &
+# end hack
 %mvn_build
```

ТРЕВОЖНЫЕ ЗВОНОЧКИ

Ошибка #39276 — 2. лимит на длительность сборки.

- Началось с libint — пакет по своей природе собирается долго. Дольше лимита сборочницы.
- Свежий пример:

java-9-openjdk не мог быть собран в p9.

Ждал 2 месяца, чтобы администратор руками вмешался в процесс сборки :(

Ошибка #39276

Одинаковые условия для всех пакетов

ТРЕВОЖНЫЕ ЗВОНОЧКИ

- ручное вмешательство (приостановка выпуска, перевод задач в состояние POSTPONED) чтобы провести в Сизиф «тяжелую» сборку.

ТРЕВОЖНЫЕ ЗВОНОЧКИ

Оба примера суть Windows-way:

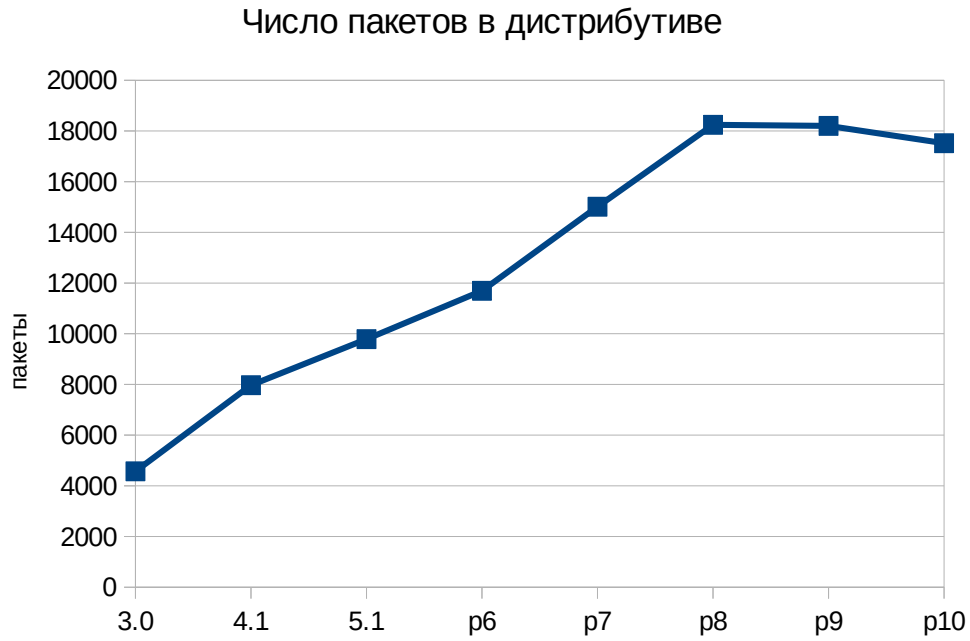
каждый раз делаем руками, вместо того, чтобы один раз (Unix-way) исправить в коде.

ТРЕВОЖНЫЕ ЗВОНОЧКИ

- Другие проблемы: рапортуются, но не чинятся, даже тривиальные.

ТРЕВОЖНЫЕ ЗВОНОЧКИ

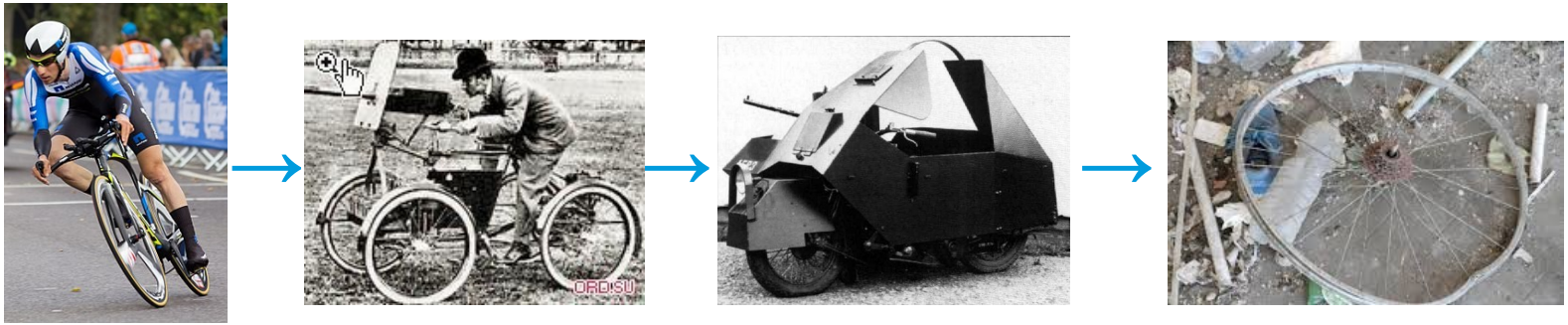
Стагнация репозитория, начиная с p8.



Технический долг

Что делать?

- Понять и простить.
- Долги платят только трусы.
- Пусть оно как-то само к чему-то придет:



ТЕХНИЧЕСКИЙ ДОЛГ

На технический долг можно и забить, но только в случае, если код -черный ящик «просто работает» без проблем.

- Спрашивает сынишка программиста:
- - Папа, почему солнышко каждый день встаёт на востоке, а садиться на западе?
- - Ты это проверял? - Проверял.
- - Хорошо проверял? - Хорошо.
- - Работает? - Работает.
- - Каждый день работает? - Да, каждый день.
- - Тогда ради бога, сынок, ничего не трогай и не меняй.

ТЕХНИЧЕСКИЙ ДОЛГ

- А что в нашем случае?

  Работает ли, как надо?

  Проверим шахту канарейками.

  (канарейки в шахте умирают первыми, предупреждая остальных шахтеров об опасности)

КАНАРЕЙКИ

Одна говорливая канарейка уже есть, это докладчик.

Но чтобы не говорили, что его проблемы со сборочницей — это его его проблемы, сам взялся сопровождать perl и java — сам и мучайся :) начну с другой канарейки — молчаливой.

КАНАРЕЙКИ

- Молчаливая канарейка, точнее, змейка, точнее, питон.
- Крест, который сейчас молча несет Григорий Устинов <grenka@basealt>.
- Молчание, которое страшнее любого слова…

Последнее обновление питона заняло 1 Месяц.

Техническая пересборка. 1 Месяц жизни.

УЖАС

- Как это работает:

  1. <разработчик>: А не взяться ли мне за упаковку библиотек/модулей/ для X? …
- 2. <разработчик вспоминает>. Чтобы пересобирать Python требуется 1 месяц? Ужас!

  3. Ну его нафиг! Не возьмусь.

  4. Силы сэкономлены. Profit!

УЖАС

- Пример из жизни:

  Малъ Скрылевъ <Majioa@> в марте обращался ко мне по проводу node.js пакетов. Когда-то я разрабатывал средства автоматизации для сопровождения node.js пакетов, до сих пор в autoimports сопровождается около 1000 node.js пакетов, чтобы не утратить компетенции. Так и не рискнул тогда идти в Сизиф, и сейчас боюсь. Посоветовал Sisyphus Node Team.

- Sisyphus. Node Team. 28 пакетов. Мудрое решение.

Поэтому так мало канареек. Java, Python, Perl…

Остальные могли бы быть, но или не хотят вылупляться из яйца, или так и умерли не родившись.

УЖАС СБОРОЧНИЦЫ

У
Ж
А
С

Кладбище языковых и других модулей.

Вернемся к первой канарейке.

JAVA

1.просто медленно,
не так страшно.

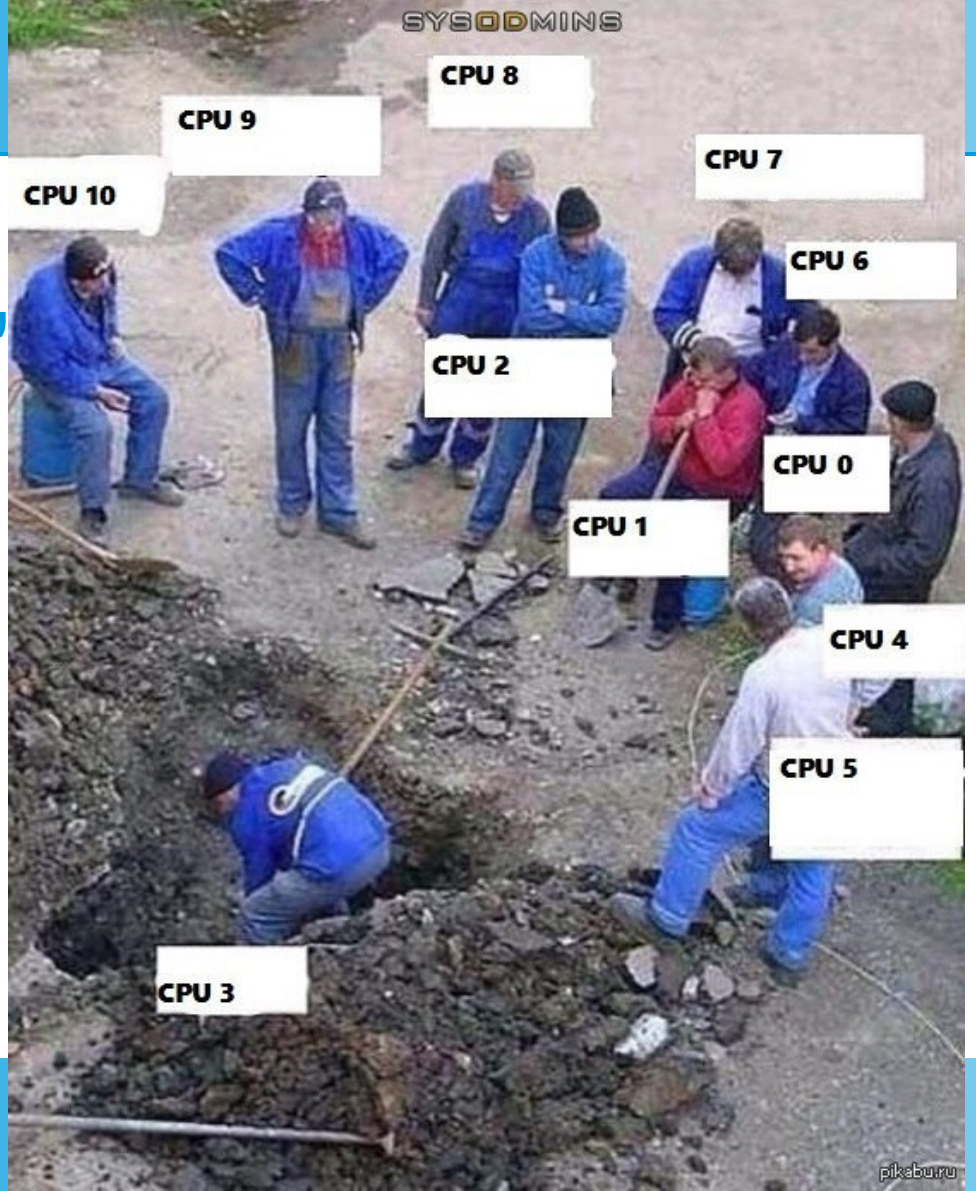
9 июня. 36 пакетов (perl)

7 минут сборки локально =

3 часа сборки в сборочнице.

переезд на java11 =

очередь длинной в месяцы?



JAVA

2. Пакет может запросто завязнуть в сборочнице.

Посреди обновления. С концами.

- Ошибка #39276 — 2. лимит на длительность сборки
- Свежий пример от 14 июня.
- [#274507] FAILED srpm=scala-2.10.6-alt3_17jpp8.src.rpm : aarch64/i586/ppc64le/x86_64 build OK

  [armh] hasher-priv: master: time elapsed limit (28800 seconds) exceeded

2. Пакет может запросто завязнуть в сборочнице.

Посреди обновления. С концами.

Но там хоть есть надежда когда-нибудь полностью переехать на java11.

Perl.

И здесь уже начинается ужас.

Ужасы с Perl. Предисловие. Круговорот сборки.

СБОРОЧНИЦА.

- Багфича — эмуляция последовательной сборочницы.

  Если во время сборки пакета A пересобрался пакет libB, входивший в сборочное окружение пакета A, то опять пересобираем пакет A.
- В обычном случае не нужно, так как для этого есть инварианты — версионированные зависимости.
- Для патологических случаев ничего не гарантирует, так как и A и libB заливаются независимо, и ничего не мешает A собраться первым, и тогда libB его сломает. Ведь мы не пересобираем затронутые пакеты при заливке libB?

Багфича от последовательной сборочницы

- Правильное лечение проблемы — выявлять пакеты со сломом интерфейса, проставлять жесткие зависимости
- Время от времени (перед бранчеванием, например) устраивать (можно двойную) пересборку всех со всеми.

Багфича от последовательной сборочницы

Выглядит как

#274008 AWAITING #1.2 sisyphus srpm=perl-DateTime-Format-MySQL-0.0701-alt2.src.rpm
#274006 AWAITING #1.2 sisyphus srpm=perl-DateTime-Format-ISO8601-0.16-alt2.src.rpm
#274005 AWAITING #1.2 sisyphus srpm=perl-DateTime-Format-ICal-0.09-alt3_31.src.rpm
#274004 AWAITING #1.2 sisyphus srpm=perl-DateTime-Format-IBeat-0.161-alt3_35.src.rpm
#274002 AWAITING #1.3 sisyphus srpm=perl-DateTime-Format-Flexible-0.33-alt2.src.rpm
#274001 BUILDING #1.3 [locked] sisyphus srpm=perl-DateTime-Format-Excel-0.31-alt3_27.src.rpm
#274000 AWAITING #1.3 sisyphus srpm=perl-DateTime-Format-Epoch-0.16-alt2_14.src.rpm
#273999 AWAITING #1.3 sisyphus srpm=perl-DateTime-Format-DBI-0.041-alt2_16.src.rpm

Бывали случаи и AWAITING #1.30, сейчас легче, но не с perl.

Багфича от последовательной сборочницы

- Правильная починка сборочницы — проверять готовность task-а не по анализу сборочных окружений, а по unmets и symbols db.
- Уйдет большинство AWAITING #1.2

Ужасы с Perl.

- Первый ужас: вечный круговорот сборки

  В старой чисто последовательной сборочнице Perl собирался за 1 раз. Протестировав локально сборку, можно было быть уверенным, что она (тогдашняя) соберется за 15-16 часов.

  В новой сборочнице Perl попал в круговорот пересборок.

  BUILDING #1 теперь почти сутки.

- 1) Добавился практически обязательный шаг

  AWAITING #1 → AWAITING #1.2 (еще сутки)

Ужасы с Perl.

2) Далее крутим рулетку.

AWAITING #1.N → AWAITING #1.(N+1)

происходит, если в репозитории за сутки что-то изменилось, что затронуло сборочное окружение транзакции.

Ужасы с Perl.

2) Далее крутим рулетку.

AWAITING #1.N → AWAITING #1.(N+1)

За сутки что-то изменилось? Вероятность хоть и < 100%, но > 50%.

Ужасы с Perl.

Второй ужас: тесты-шатуны.

- Пусть у тестов пакетов вероятность сбоя 0.1%. Какова вероятность, что сбой случится в транзакции из 430 пакетов, еще и запущенной на 5 архитектурах?

Ужасы с Perl.

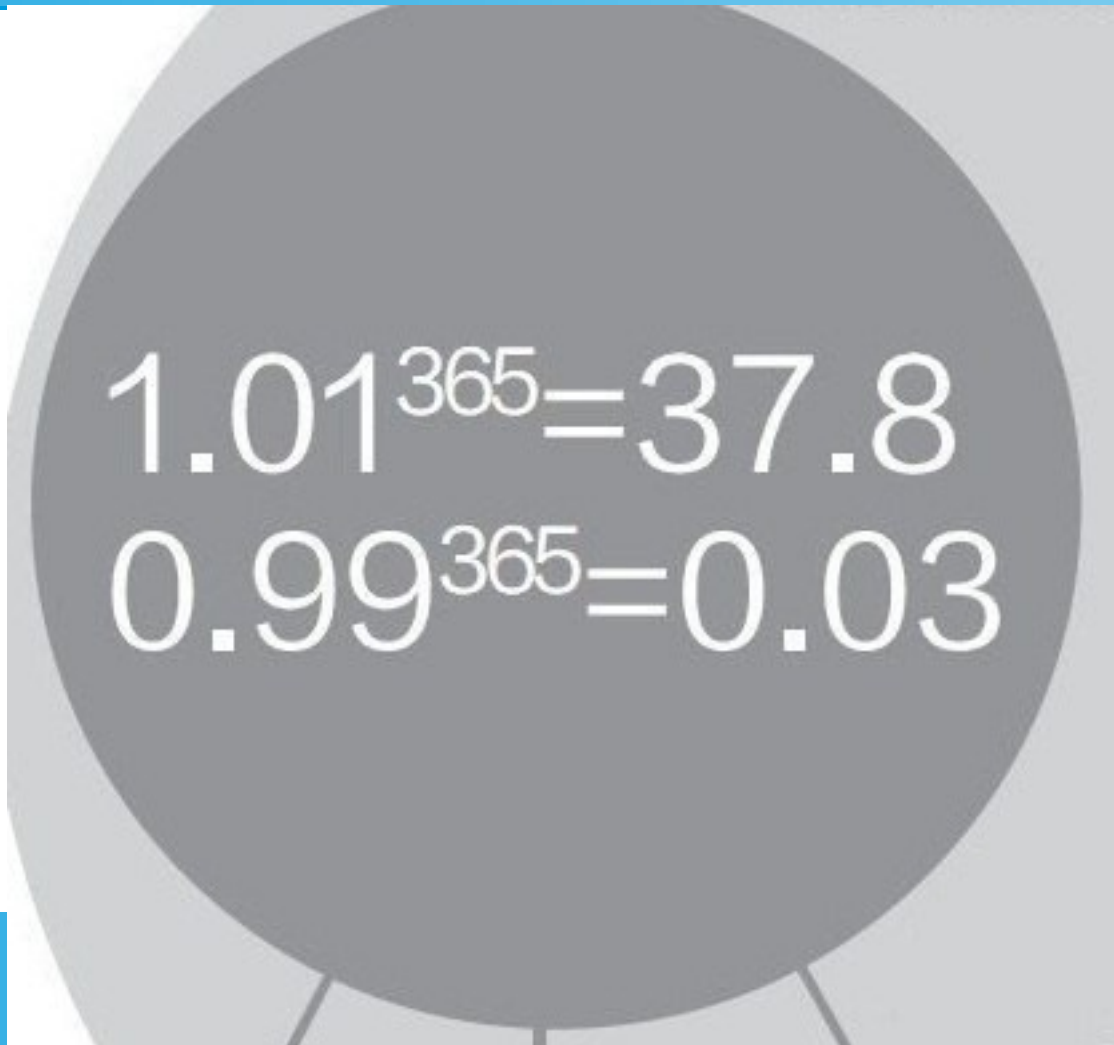
Второй ужас: тесты-шатуны.

- Ответ: > 25%!

Ужасы с Perl.

Тесты-шатуны.

- Аналог - Микрокредит «под всего 1%»



Ужасы с Perl.

Тесты-шатуны. Историческая справка

- Ранее тестами-шатунами можно было пренебречь (вероятность < 10%). Связанно было с меньшим размером транзакции (2xx,3xx против 430) и архитектур (2 против 5).

Ужасы с Perl.

Тесты-шатуны. Python.

- Питон, похоже, пока серьезно не страдает от тестов-шатунов, несмотря на больший в 1.5 раза размер транзакции. Объяснить можно меньшим покрытием тестами (37.7% против 99.67%) и тем фактом, что питон в нашей сборочнице прошел гораздо больше кругов ада, и большую часть тестов-шатунов ему уже отключили.

Ужасы С Perl. Тесты-шатуны.

В текущей сборочнице с тестами-шатунами работать очень затратно. К примеру, в транзакции тест-шатун выбил в 300-м пакете транзакции из 430.

Придется заново запускать транзакцию на сборку и вхолостую опять пересобирать эти 300 уже собранных пакетов.

Ужасы с Perl. Тесты-шатуны.

Правильная починка сборочницы — проверять готовность task-а не по анализу сборочных окружений, а по unmets и symbols db — помог бы и против тестов-шатунов. Раз мы перестанем гнаться за самым последним репозиторием, то ничто не помешает создать команду task run —continue.

Засбоил 300-й пакет? task run —continue и продолжаем заново не с первого пакета, а с 300-го.

И питону task run —continue здорово бы пригодился.

Ужасы с Perl.

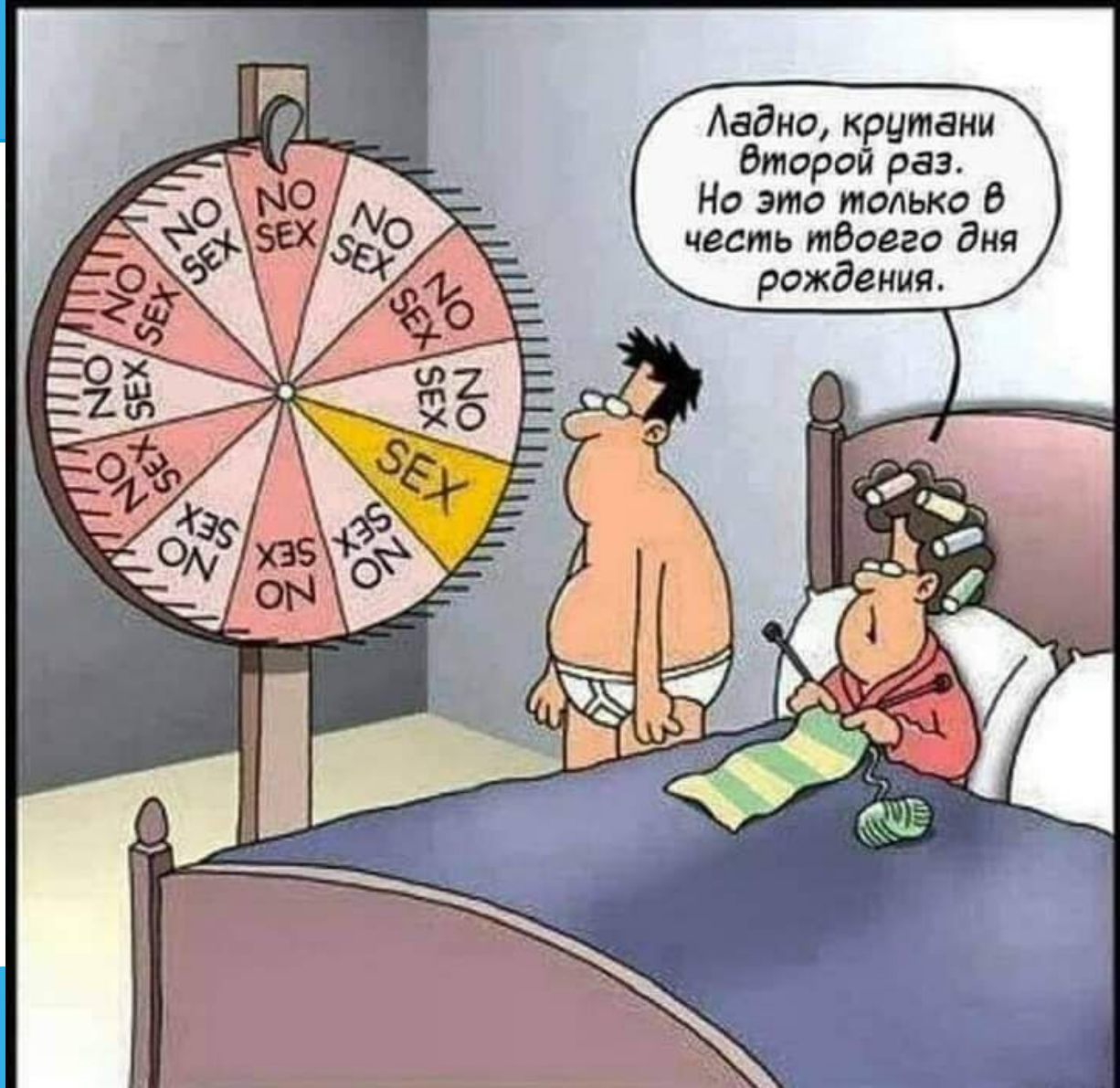
Казино сборочницы.

С вероятностью 25% сбой,

с вероятностью > 50%
идем на следующий круг.

Ужасы с Perl.

По счастью, помогает вмешательство администратора, который приостанавливает (PENDING) другие задания в сборочнице и не дает ей уйти на следующий круг.



Ужасы с Perl.

Казино сборочницы.

Ужас в том, что волей-неволей транзакция perl растет, и вероятность нормально собраться в текущей сборочнице для нее потихоньку падает по направлению к нулю.
На глазах perl превращается во второй (по мучениям со сборкой) python3 :((

Ужасы с Perl.

- Позапрошлый perl 5.30.1— уже готовая транзакция намотала 8 кругов, в том числе сбои тестов-шатунов, 8 суток.
- Прошлый perl 5.32.1— уже готовая транзакция намотала 3 круга, 3 суток
- Текущий perl 5.34.0— уже готовая транзакция напоролась на тест-шатун, намотала 3 круга, 3 суток, вовремя подоспело вмешательство адиминистратора.

ПОМОГИТЕ! ТОНУ!

Что делать?

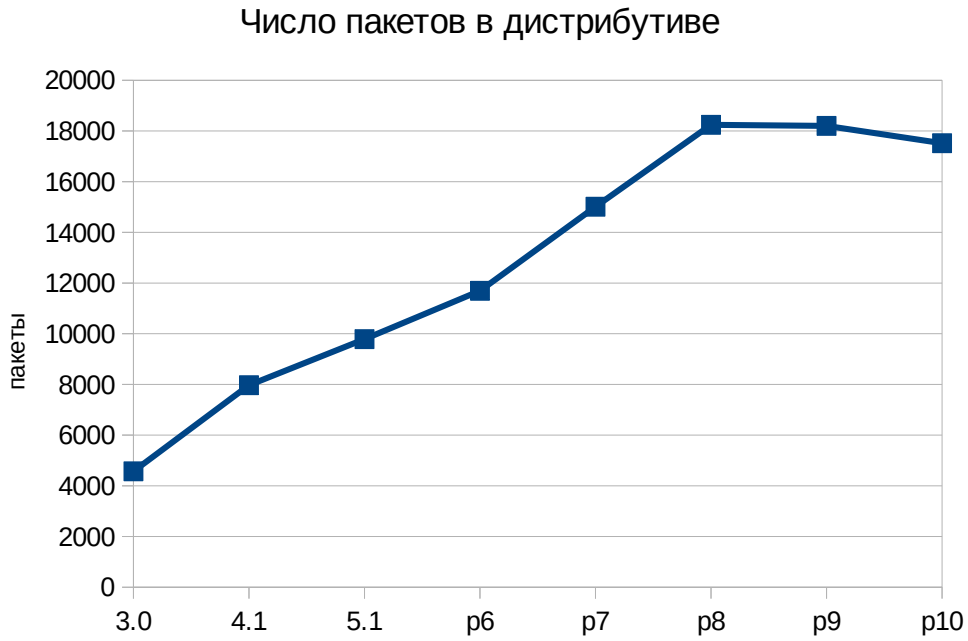
- Можно и дальше игнорировать проблему.

УЖАС СБОРОЧНИЦЫ

У
Ж
А
С

Кладбище языковых и других модулей.

Стагнация репозитория, начиная с p8.



ЧТО ДЕЛАТЬ?

Горькое, но нужное лекарство — Рефакторинг.

ЧТО ДЕЛАТЬ?

Рефакторинг:

- Разделение сборочницы на независимые части
- инкапсуляция task’ов
- вынос части компонент сборочницы в пространство пользователя:
  - Утилиты создания и обслуживания каталога task
  - Сборщик пакетов в каталоге task
  - Мержер собранного taska с репозиторием.

ЧТО ДЕЛАТЬ?

Горькое, но нужное лекарство —

Рефакторинг.

- Привлечет пользователей к разработке (смогут разрабатывать и испытовать локально)
- В перспективе даст «карманы» и дистрибутивную сборочницу.

Алгоритмы параллельной обработки пакетов

За весь доклад я так и не рассказал о собственно алгоритмах параллельной обработки пакетов. Да, есть такие алгоритмы, позволят существенно поднять производительность сборочницы, как на обычных заданиях (уйдут очереди), так и для тяжелых транзакций, таких, как perl с модулями или kernel с модулями.

Но в процессе подготовки доклада понял, что говорить о них преждевременно, пока сама возможность рефакторинга сборочницы под вопросом. Да, я могу написать прототип, но зачем писать «в стол»?

Останется ли сборочница прежней?

Кто-то в ней еще плавает,
а кто-то уже тонет.

У меня:

- Java — пакеты клинит.
- Perl — без пригляда администратора сборка стремится войти в бесконечный цикл.