Resilience Engineering and incident response

2021

Владимир Федорков

Learning

Monitoring

Responding

Adapting

slurm.io

У кого не падал прод?

Прод падает постоянно

- Отказы железа и сети
- Баги в коде
- Баги в софте
- Ошибки человека
- Природные катастрофы
  - (крайне редко)

Отказ системы это страшно

- Ничего не работает
- Что сломалось – не ясно
- Как чинить – не понятно
- Хорошо, если руководства нет

Немного о стрессе

- Непонятная ситуация рождает стресс
- Стресс блокирует работу мозга
  - Бей – беги это не про подумать
    - Замереть и подождать пока все закончится
    - Убежать как можно быстрее
    - Наорать на кого-нибудь («победить» в «битве»)
- Что умеет делать заблокированный стрессом мозг ?
- ПАНИКОВАТЬ!!!

Как-то не очень все позитивно

- Эмоции мешают
- Голова не работает
- Алкоголь повышает вероятность ошибки
  - И убивает последние рабочие нейроны
- Замкнутый круг
- Как было бы хорошо, если бы отказов вообще не было!

Отказ – ШТАТНАЯ ситуация

- Чем больше система – тем больше компонент
- Допустим вероятность отказа сервера за день 1/500
- 1 сервер:
  ожидаемое время наработки на отказ около года
- 1000 серверов:
  сегодня упадут два сервера

Закомьтесь, это писец, он приходит.

Хорошие новости!

- Мы не в медицине!
- Мы не в авиации!
- Мы не в атомной энергетике!
- И даже не в разработке беспилотных автомобилей!

Страшно, когда НЕ падает

- Совершенно не ясно, что делать
- Без понятия как чинить, если сломалось
- Как отказ одного компонента отразится на остальных – вообще не ясно
- Разработка знает как код работает
  - И совершенно точно не знает как он падает

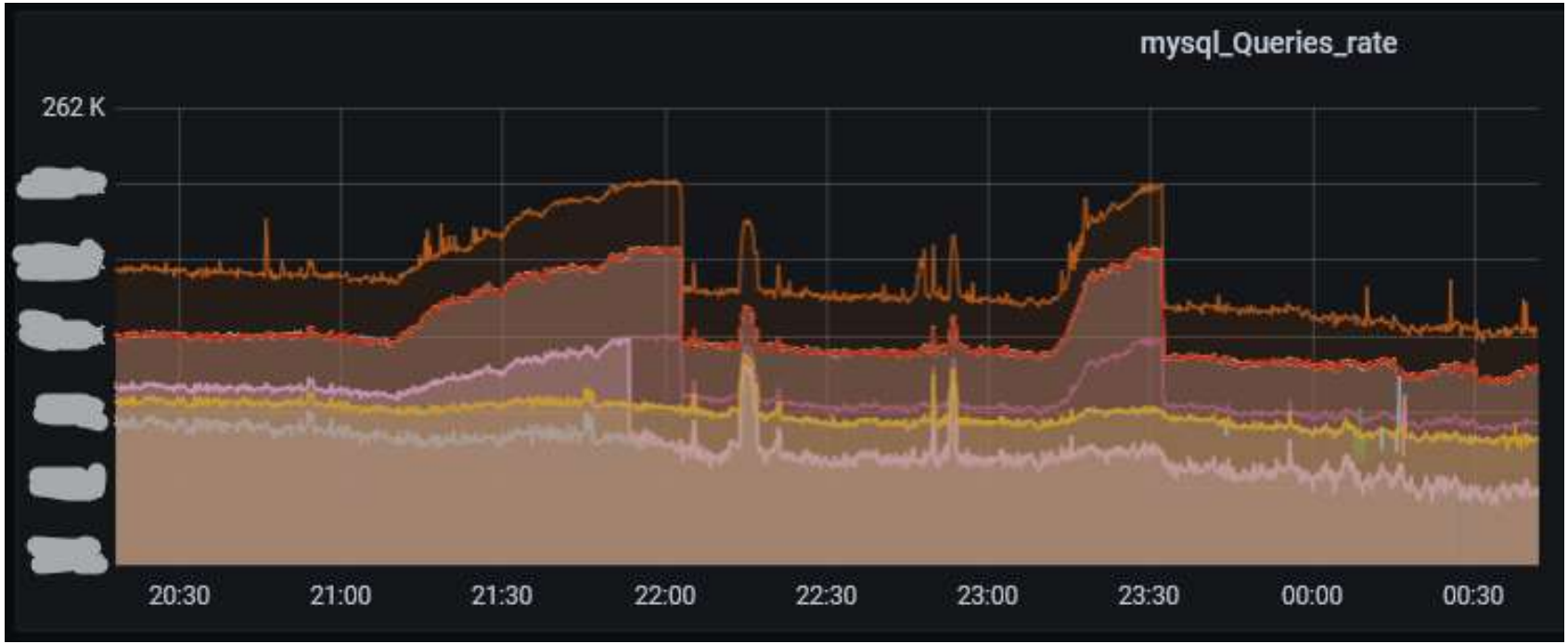
Если не падает – нужно уронить

- Нагрузочное тестирование
- Стресс тестирование
- Chaos engineering

Нагрузочное тестирование

- Эмулирует пользовательскую нагрузку
  - Пока система не упадет
- Позволяет выявить узкие места системы
- И конечно ее уронить!
  - Бонусом посмотреть как она восстанавливается
    - Или нет 

Выглядит примерно так



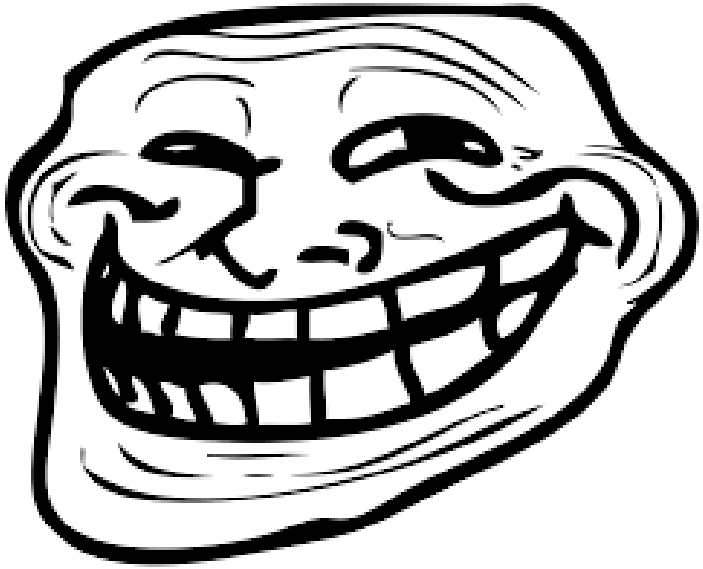
Стресс тестирование

- Нагружается отдельный компонент
  - Синтетической нагрузкой
  - Или приближенной к продовой
- До тех пор, пока не умрет
  - Совсем и до конца
- После этого нагрузка еще увеличивается
- В процессе снимаем метрики
- Можем делать на стенде, можем в проде

Стресс тестирование выглядит так

Chaos engineering

- Выключаем любой сервер
- Выдираем любой кабель
- Убиваем ресурсы
  - Заполняем диск
  - Забиваем память
  - Убиваем CPU
- И смотрим как система будет себя чувствовать



Зачем все это?!

- Системы стали слишком сложными
- Научиться решать контролируемые инциденты

Системы слишком сложные

- Legacy накапливается
- Зависимости усложняются
- Как это работает – никто не знает
- Даже те, кто писали и строили

Отказ

- Дает понимание работы систем
- Выявляет скрытые взаимосвязи
- Показывает где нужно внимание
- Сплачивает команду
- Outage-Driven Development
  - Очень эффективно
  - Но очень дорого

За одного битого двух небитых дают

- Учимся действовать в нештатных условиях
- Борем стресс опытом и наработками
- Доводим умения до рефлексов
- Рефлексы сильнее страха
- Пока действуем – мы эффективны

Как изучить кунг-фу?

- Хреновые новости – мы, кажется, не в матрице
  - Поэтому только многократным повторением
  - В доброжелательной обстановке
  - Например на мастер классе в Слёрме!
- Боремся с хаосом

Планируем работу на инциденте

- Оцениваем ситуацию
  - И влияние на конечных пользователей
- Зовем помощь
- Распределяем обязанности
- Поддерживаем коммуникацию

Правила работы на инциденте

- «Поспешай не торопясь!»
- Не гадаем – проверяем
- Не верим домыслам, верим фактам
  - Графики, логи, вывод команд, скрины
- Озвучиваем свои действия
  - Но не очень подробно
- Кратко сообщаем находки
- Мало говорим, внимательно слушаем
- Никого не обвиняем!

Создаем общее пространство

- «WarRoom», «IncidenRoom»
- Говорить быстрее, чем печатать
  - Артефакты сбрасываем в канал, пригодятся
- Возможность расшарить экран экономит много времени
    - Zoom, GoogleMeet, Skype, Discord, YouNameIt.
- Никого не ждём – сразу идем смотреть
- Забираем лидерство в делах
  - Не пытаемся быть генералом
  - И не забываем о вежливости

Пример

- 16:11 [1] @here кажется, лежим, пошли в зум!
- 16:12 [2] Солр упал, поиск отвалился и пятисотит, остальное живо
- 16:12 [1] @Петр, глянь что с ним
- 16:15 [Пётр] Много данных приехало, висим
- 16:15 [1] Переключаемся на резервный?
- 16:15 [2] Пошел переключать
- 16:16 [1] Сколько данных приехало, надолго индексация?
- 16:17 [2] Переключил
- 16:17 [Пётр] часа на полтора в прошлый раз было
- 16:17 [3] пойду колл-центр предупрежу, что часть товаров не в поиске

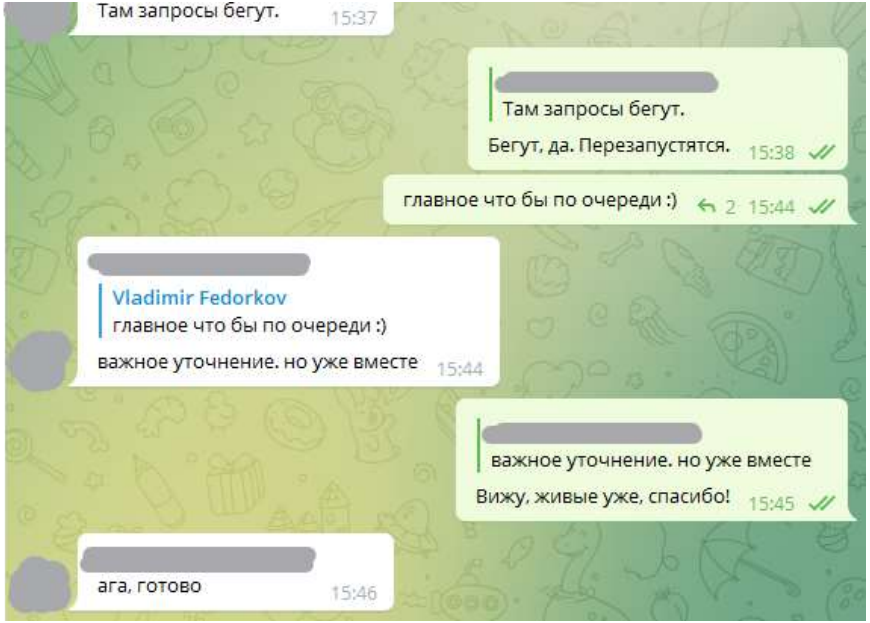
Что делает лидер?

- Поддерживает темп
- Запускает потоки
- Приглашает людей
- Модерирует процесс
- Держит эфир чистым
- Передает лидерство или завершает

От импровизации к процессу

- Вырабатываем привычки
- Экономим время
  - Аккуратные алерты
  - Ранбуки
  - Визуализация состояния системы (i.e. grafana)
  - Логгирование и поиск по логам (i.e. kibana)
- Один отказ – вынужденная ситуация
  - Два отказа – роскошь
    - Три отказа – халатность
- Анализируем причины
- Пишем PostMortem’ы

Ошибка человека – вопрос времени.



Вопросы?

СЛЁРМ

Спасибо!

СЛЁРМ