

# Почему мы быстро считаем в банке

Антон Батяев

Deutsche Bank TechCenter



**HighLoad**<sup>++</sup>  
Siberia 2019

Профессиональная конференция  
для разработчиков высоконагруженных  
систем





Антон Батяев

# Почему мы быстро считаем в банке

#PositiveImpact

[anton.batiaev@db.com](mailto:anton.batiaev@db.com)

# Почему мы быстро считаем **риски**



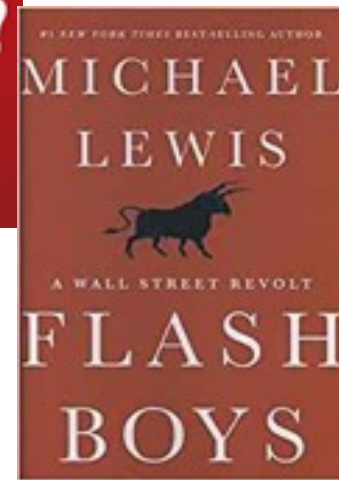
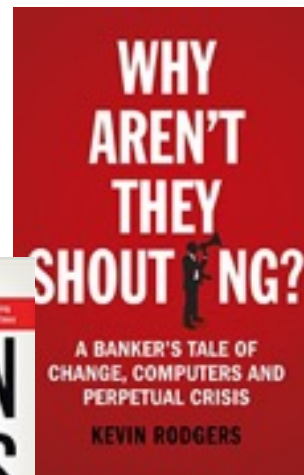
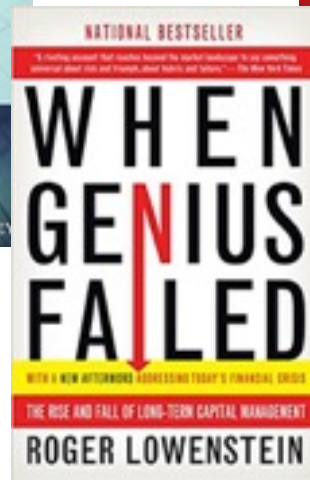
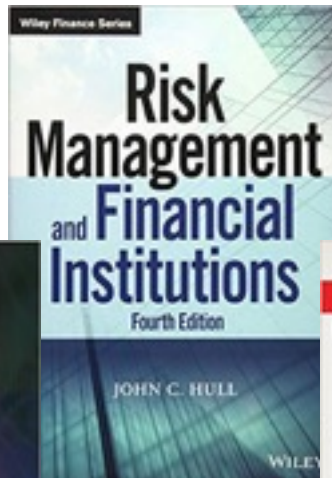
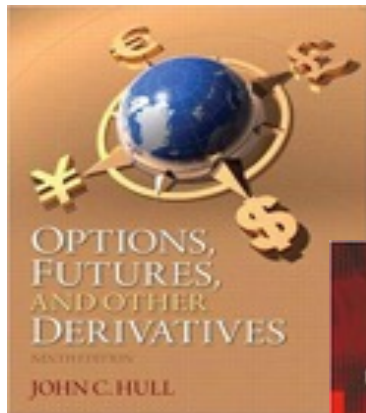
LTCM (1998)

Lehman Brothers (2008)

# О рисках в кинематографе



# О рисках и деривативах в литературе



# Fixed Income & Currencies



FX



Rates



Credit

# Расчет рисков по деривативам



olegchir 30 апреля 2019 в 11:03



## «Предложили ознакомиться с расчетами одного показателя, а там два листа с интегралами и вторыми производными»

Блог компании JUG.ru Group, Блог компании Технологический Центр Дойче Банка, Высокая производительность, Java, Финансы в IT

Это интервью с Антоном Батяевым (@batiaev) из Технологического Центра Дойче Банка.

<https://habr.com/ru/company/dbtc/blog/449630/>

# Производные финансовые активы (деривативы)

**Дериватив** — финансовый инструмент, договор в котором вы обязуетесь купить, продать, обменять базовый актив в будущем

## Виды Деривативов

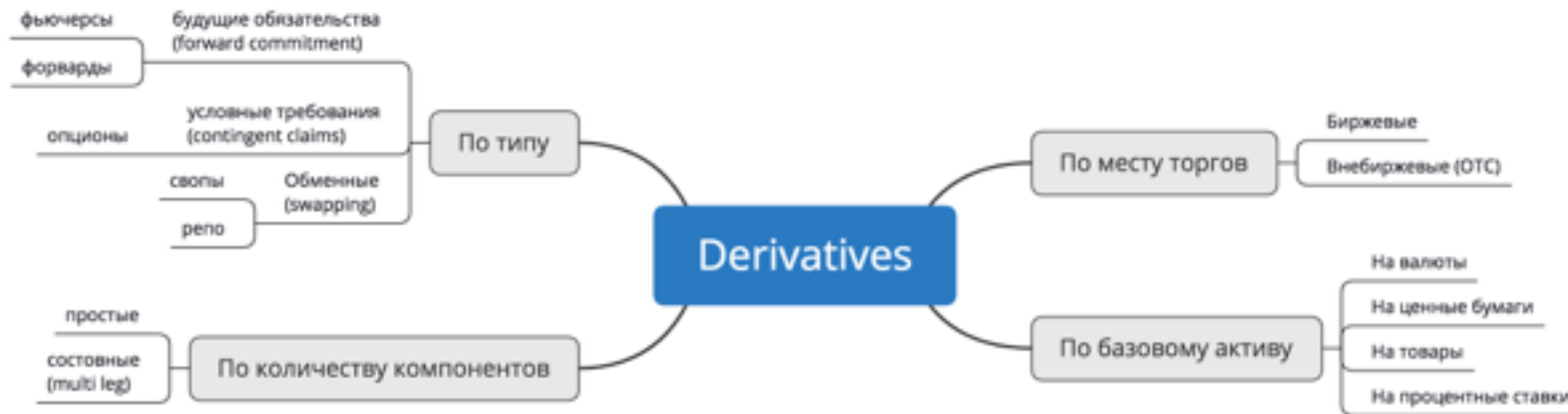




# Бывают разных видов



Производных финансовых активов ([Derivatives](#)) существует большое разнообразие:



# Некоторые показатели считать просто



Фьючерсная цена — по какой цене вы сейчас договоритесь с банком купить доллары в будущем

$$F = S * (1 + r)^{T/365}$$

где

- F - фьючерсная цена (будущая котировка доллара)
- S - цена базового актива (текущая котировка)
- T - количество дней до даты в будущем
- r - ставка дисконтирования

# Другие показатели сложнее



Оценка опциона — величина премии, например за возможность купить доллары в июле по 60 руб.

$$C(S_t, t) = N(d_1)S_t - N(d_2)PV(K)$$

$$d_1 = \frac{1}{\sigma\sqrt{T-t}} \left[ \ln\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t) \right]$$

$$d_2 = d_1 - \sigma\sqrt{T-t}$$

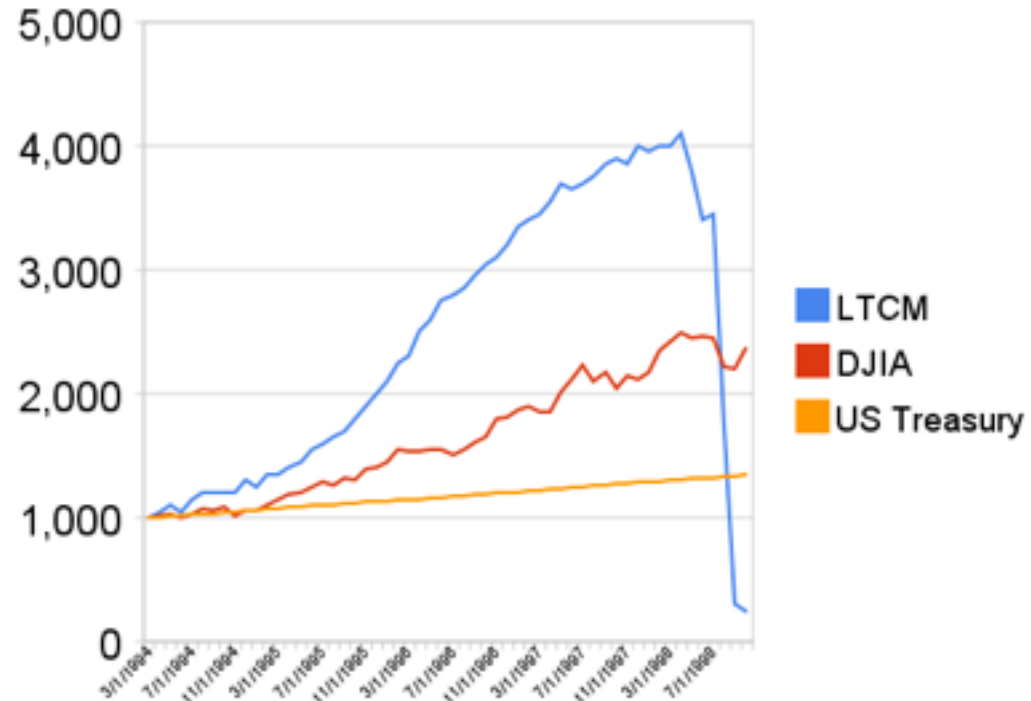
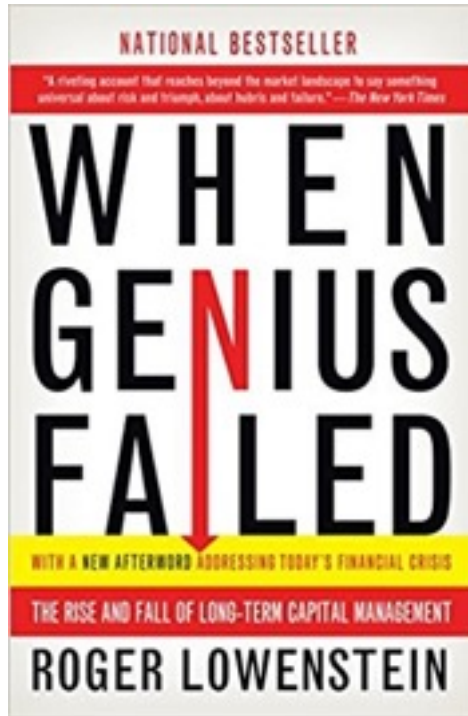
$$PV(K) = Ke^{-r(T-t)}$$

\* [Black Scholes model](#)

где

- $N(x)$  — кумулятивная функция распределения стандартного нормального распределения
- $\sigma$  - волатильность доходности базисного актива
- и ряд других параметров ...

# А расчет рисков еще более сложный и важный



# Задачи



1. Читать правильно
2. Читать быстро
3. Хранить компактно
4. Релизить без ошибок
5. Быстро разбираться в инцидентах



И на чем же  
мы строим системы  
для таких расчетов?

# Технологии под капотом



ClickHouse



mongoDB®



protobuf  
Protocol Buffers



React



spring®

# Основные точки для оптимизаций



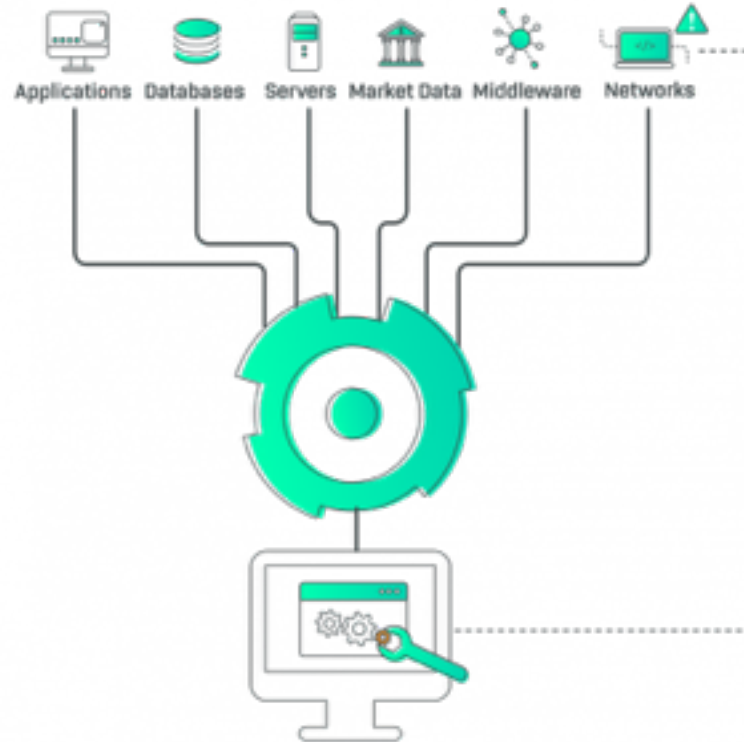
1. Supportability
2. Тестирование
3. CI/CD
4. Контроль памяти и GC
5. Объемы данных
6. Алгоритмы
7. Загрузка грида



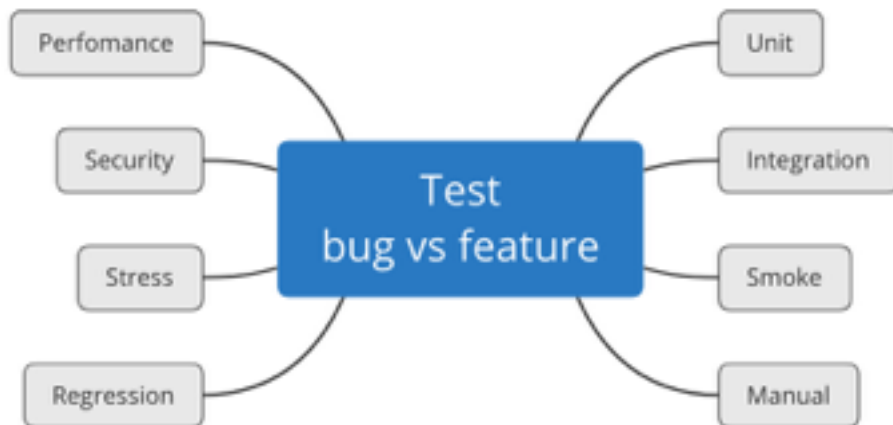
# Supportability - GENEOS



- jmx
- OS
- metrics
- hardware
- network
- logs
- ...

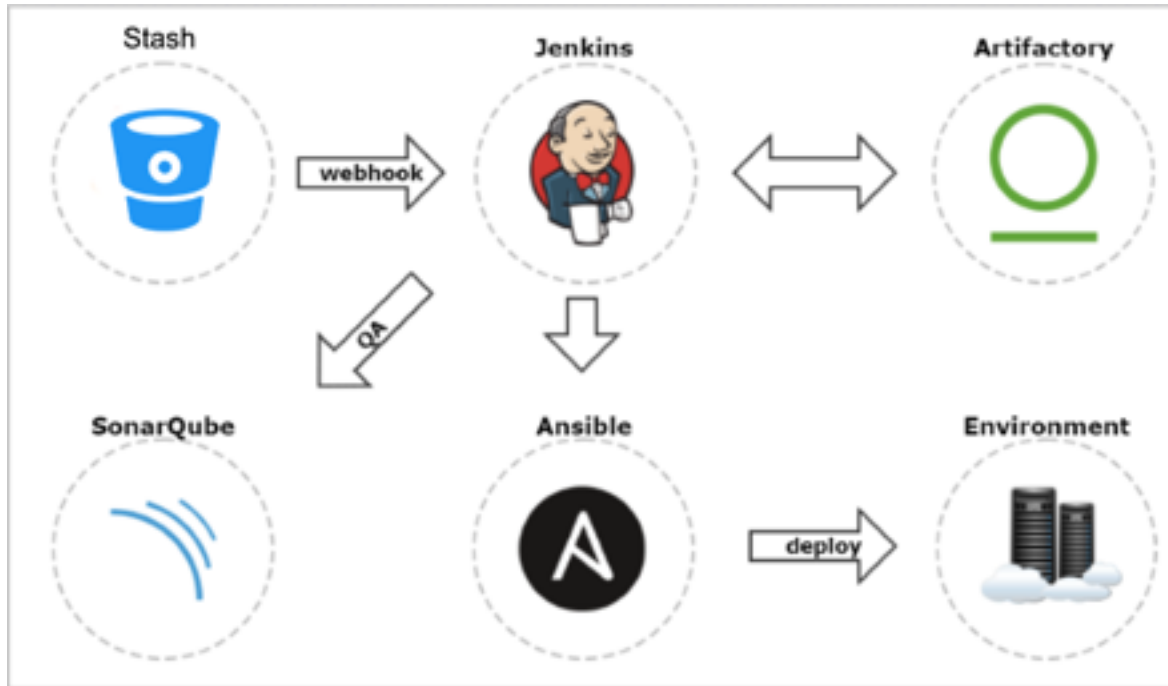


# Автотестирование - test harness



- сравнение корректности расчетов
- проверка UI
- сравнение перформанса
- анализ сборок мусора
- анализ потребляемой памяти
- выявление тормозящих сервисов

# CI/CD pipeline



# Работа с памятью и GC

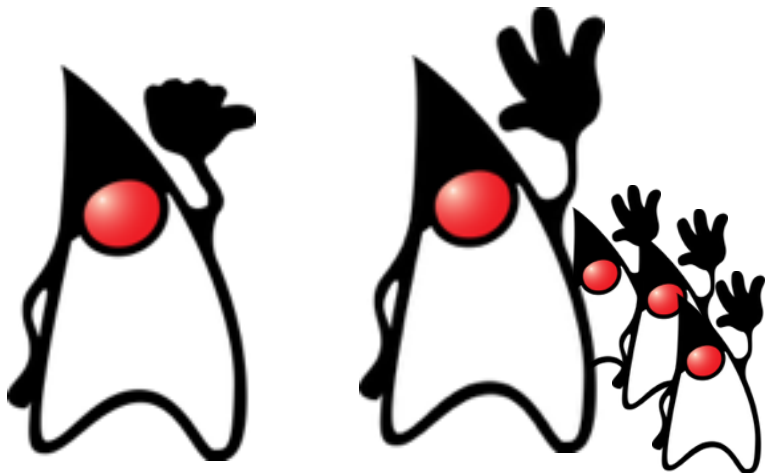


Serial GC



**-XX:+UseSerialGC**

# Работа с памятью и GC



**-XX:UseParallelGC**  
**-XX:+UseParNewGC**  
**-XX:+UseParallelOldGC**

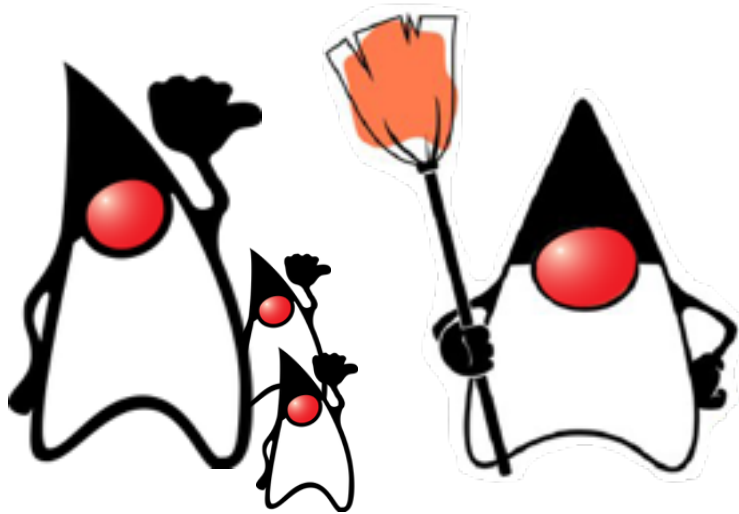
# Работа с памятью и GC



~~Serial GC~~

~~Parallel GC~~

CMS GC



**-XX:+UseConcMarkSweepGC**

# Работа с памятью и GC



~~Serial GC~~

~~Parallel GC~~

~~CMS GC~~

Garbage first  
(G1)

**-XX:+UseG1GC**



# JVM-ключи. Общие



-server

-XX:+UseG1GC

-XX:+UnlockExperimentalVMOptions

-XX:+UseCompressedOops

-XX:-OmitStackTraceInFastThrow

-XX:+DisableExplicitGC



# JDK 8 JVM-ключи. GC логи



-XX:+PrintGC

-XX:+PrintGCDetails

-XX:+PrintGCTimeStamps

-XX:+PrintGCDateStamps

-XX:+PrintAdaptiveSizePolicy

-XX:-PrintCompilation

-Xloggc:logs/gc.log

# JDK 8 JVM-ключи. G1 тюнинг



-XX:MaxGCPauseMillis=3000

-XX:G1MixedGCLiveThresholdPercent=95

-XX:G1HeapWastePercent=5

-XX:G1OldCSetRegionThresholdPercent=25

-XX:G1MixedGCCountTarget=12

# JDK 8 JVM-ключи. Память.



-XX:MetaspaceSize=200M

-XX:InitialBootClassLoaderMetaspaceSize=100M

-XX:MinMetaspaceFreeRatio=50

-XX:MaxMetaspaceFreeRatio=80

-XX:MinMetaspaceExpansion=8M

-XX:MaxMetaspaceExpansion=32M

# JDK 11 JVM-ключи



```
-XX:MaxGCPauseMillis=3000  
-Xlog:gc*,gc+ergo*=trace,classhisto*=trace:file=logs/gc.log:time  
-XX:+PrintFlagsFinal  
--add-opens jdk.management/com.sun.management.internal=ALL-  
UNNAMED
```

## Пример 1. Streams



```
long[] getDurations(List<Bond> bonds) {  
    return bonds  
        .stream()  
        .mapToLong(Bond::getDuration)  
        .distinct()  
        .toArray();  
}
```

# Пример 1. Streams



```
long[] get Durations (List<Bond> bonds) {  
    return bonds  
        .stream()  
        .mapToLong (Bond::getDuration)  
        .distinct()  
        .toArray();  
}
```

# Пример 1. Аллокации



```
LongStream distinct() {  
    // While functional and quick to implement,  
    // this approach is not very efficient.  
    // An efficient version requires  
    // a long-specific map/set implementation.  
return boxed()  
    .distinct()  
    .mapToLong(i -> (long) i);  
}
```

# Пример 1. Аллокации



```
LongStream distinct() {  
    // While functional and quick to implement,  
    // this approach is not very efficient.  
    // An efficient version requires  
    // a long-specific map/set implementation.  
    return boxed()  
        .distinct()  
        .mapToLong(i -> (long) i);  
}
```



# Пример 1. Решение



```
import gnu.trove.set.hash.TLongHashSet;  
  
long[] getDurations(List<Bond> bonds) {  
    TLongHashSet durations =  
        new TLongHashSet(bonds.size());  
    for (Bond bond : bonds)  
        durations.add(bond.getDuration());  
    return bonds;  
}
```

# Пример 1. Решение



```
import gnu.trove.set.hash.TLongHashSet;  
  
long[] getDurations(List<Bond> bonds) {  
    TLongHashSet durations =  
        new TLongHashSet(bonds.size());  
    for (Bond bond : bonds)  
        durations.add(bond.getDuration());  
    return bonds;  
}
```

# Уплотнение данных



## Исходные данные:

- ~2000 опционов и фьючерсов \* ~ 7 серий по инструменту  
~ = 15 000 инструментов
- ~ 100 000 в день обновлений в день
- ~ 30 риск-показателей
- ~ 200 дополнительных пересчетов для шокирования  
каждого инструмента

Итого более  $9 \cdot 10^{12}$  пересчетов в день

# Уплотнение данных



- Перевод с csv/json в бинарное представление
- Использования словарей для еще большего уплотнения
- Перевод строкового на колоночное представление
- Выкидывание пустых нулевых значений

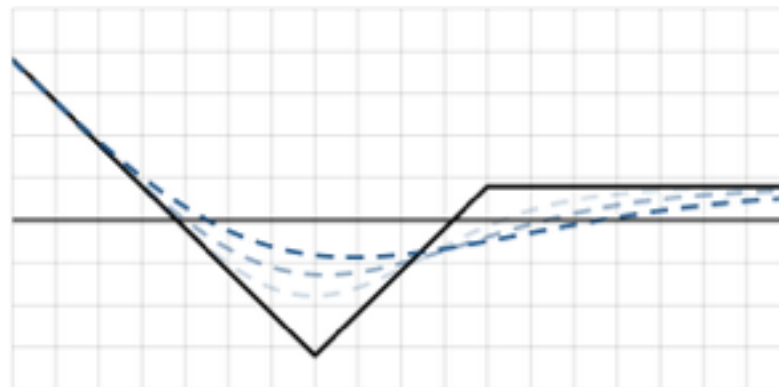
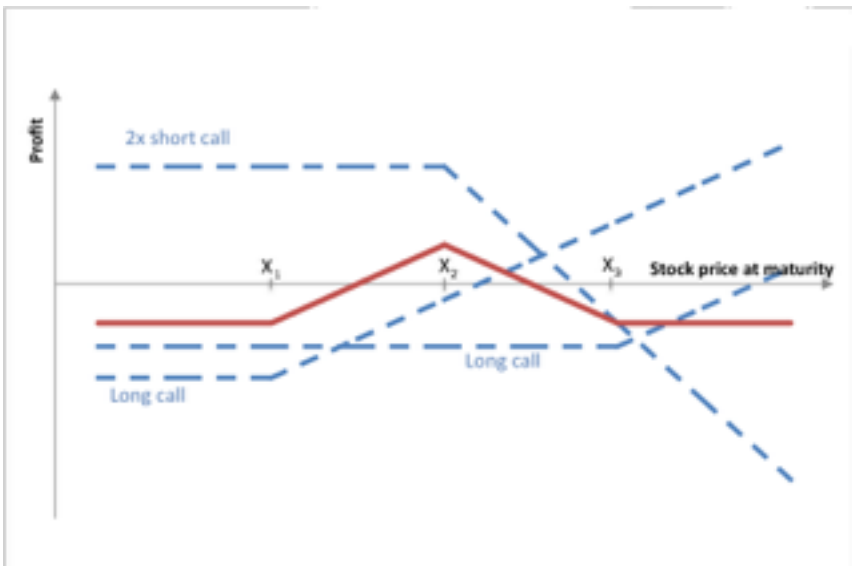
# Уплотнение данных



## Результаты

- перевод одной системы с rest+csv на protobuf дал выигрыш в 10 раз (40 -> 4 Gb данных)
- перевод строчного на колоночное представление данных позволил выкинуть дальние незначащие нули в конце сетки данных дал минус ~45% в объеме
- уплотнение статических данных с использованием словарей ~15%

# Интерполяция и аппроксимация



# Работа с гридом



Как поднять когда мы досчитаем и как максимально загрузить грид с более чем 15k CPU

- предсказание сколько времени потребуется на расчет конкретной задачи
- как максимально плотно уложить задачи на грид без простоя
- анализ загрузки утилизации грида

# Дисклеймер



Данный материал не является предложением или предоставлением какой-либо услуги. Данный материал предназначен исключительно для информационных и иллюстративных целей и не предназначен для распространения в рекламных целях. Любой анализ третьих сторон не предполагает какого-либо одобрения или рекомендации. Мнения, выраженные в данном материале, являются актуальными на текущий момент, появляются только в этом материале и могут быть изменены без предварительного уведомления. Эта информация предоставляется с пониманием того, что в отношении материала, предоставленного здесь, вы будете принимать самостоятельное решение в отношении любых действий в связи с настоящим материалом, и это решение является основанным на вашем собственном суждении, и что вы способны понять и оценить последствия этих действий. ООО "Дойче Банк Техцентр" не несёт никакой ответственности за любые убытки любого рода, относящиеся к этому материалу.





# Спасибо за внимание

#PositiveImpact

Антон Батяев  
[anton.batiaev@db.com](mailto:anton.batiaev@db.com)