Performance improvement of .NET 5 GC

Женя Рыжикова, STC Group

Зачем все эти улучшения?

“Сборка мусора — основной источник проблем с производительностью”

Federico Lois, DotNext 2017

Зачем все эти улучшения?

Посмотрим на измерения

Повторяющаяся сортировка массива в фоновом потоке

```
new Thread( start: () =>
{
    var a = new int[20];
    while (true) Array.Sort(a);
}){ IsBackground = true }.Start();
```

```
var sw = new Stopwatch();
while (true)
{
    sw.Restart();
    for (int i = 0; i < 10; i++)
    {
        GC.Collect();
        Thread.Sleep( millisecondsTimeout: 15);
    }
    Console.WriteLine(sw.Elapsed.TotalSeconds);
}
```

Источник: https://devblogs.microsoft.com/dotnet/performance-improvements-in-net-5/

Посмотрим на измерения

Повторяющаяся сортировка массива в фоновом потоке

Теперь примерно в 40 раз быстрее

.NET Core 3.0	.NET 5
6.6419048 seconds	0.159311 seconds
5.5663149 seconds	0.159453 seconds
5.7430339 seconds	0.1594669 seconds
6.032052 seconds	0.1593328 seconds
7.8892468 seconds	0.1586566 seconds

Источник: https://devblogs.microsoft.com/dotnet/performance-improvements-in-net-5/

Посмотрим на измерения

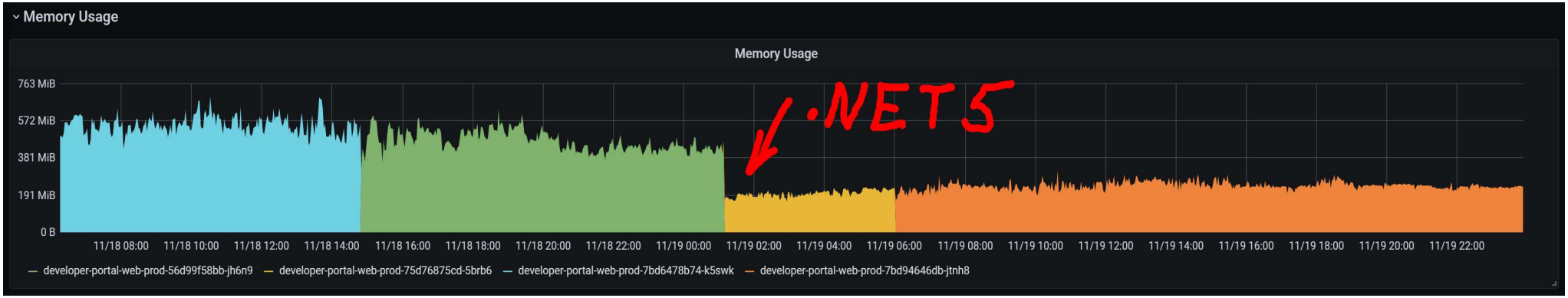
Некоторые тесты от TechEmpower

Plaintext	Json	Fortunes
11,712,405 rps	1,243,436 rps	417,818 rps
+ 38 %	+ 42 %	+ 20 %

Источник: https://github.com/TechEmpower/FrameworkBenchmarks

Посмотрим на измерения

Переезд небольшого приложения в компании ServiceTitan



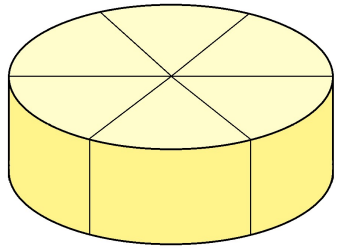
Источник: https://medium.com/swlh/astonishing-performance-of-net-5-more-data-5cdc8d821e8c

Немножко о запасе теории

1. Сборка мусора делится на несколько фаз: выбор поколения, пометка, планирование очистки, очистка и/или уплотнение
2. Режимы: server и workstation, concurrency и non-concurrency
3. GCStopTheWorld – приостановка всех потоков приложения в угоду потокам GC
4. Иногда приложение получает новые страницы памяти у ОС или возвращает их

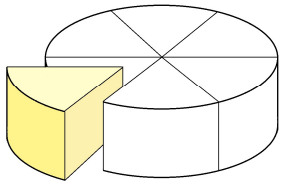
Как достигли таких результатов?

1. Балансировка работы потоков в mark_phase
2. Сглаживание частоты возвратов памяти ОС
3. Уменьшение частоты конфликтов при сканировании статики
4. Оптимизации в reset_memory
5. Уменьшение времени GCStopTheWorld
6. Использование vxsort



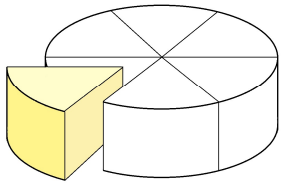
Улучшения в mark_phase

1. Балансировка работы потоков в mark_phase
2. Сглаживание частоты возвратов памяти ОС
3. Уменьшение частоты конфликтов при сканировании статики
4. Оптимизации в reset_memory
5. Уменьшение времени GCStopTheWorld
6. Использование vxsort



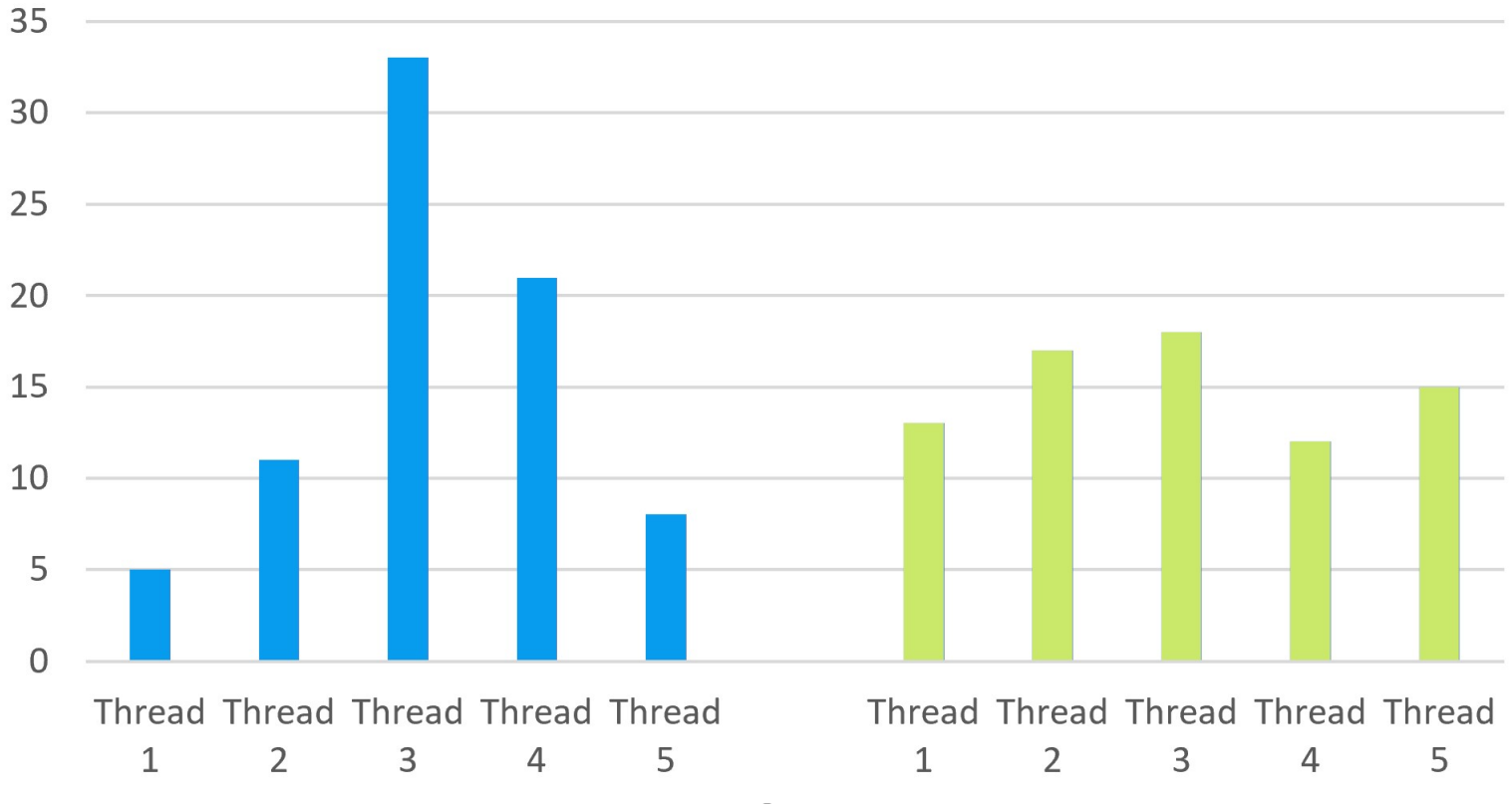
Улучшения в mark_phase

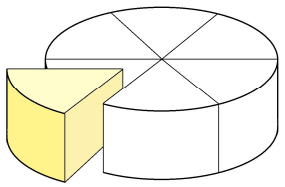
“Is .NET ServerGC setting a good thing in servers systems?”



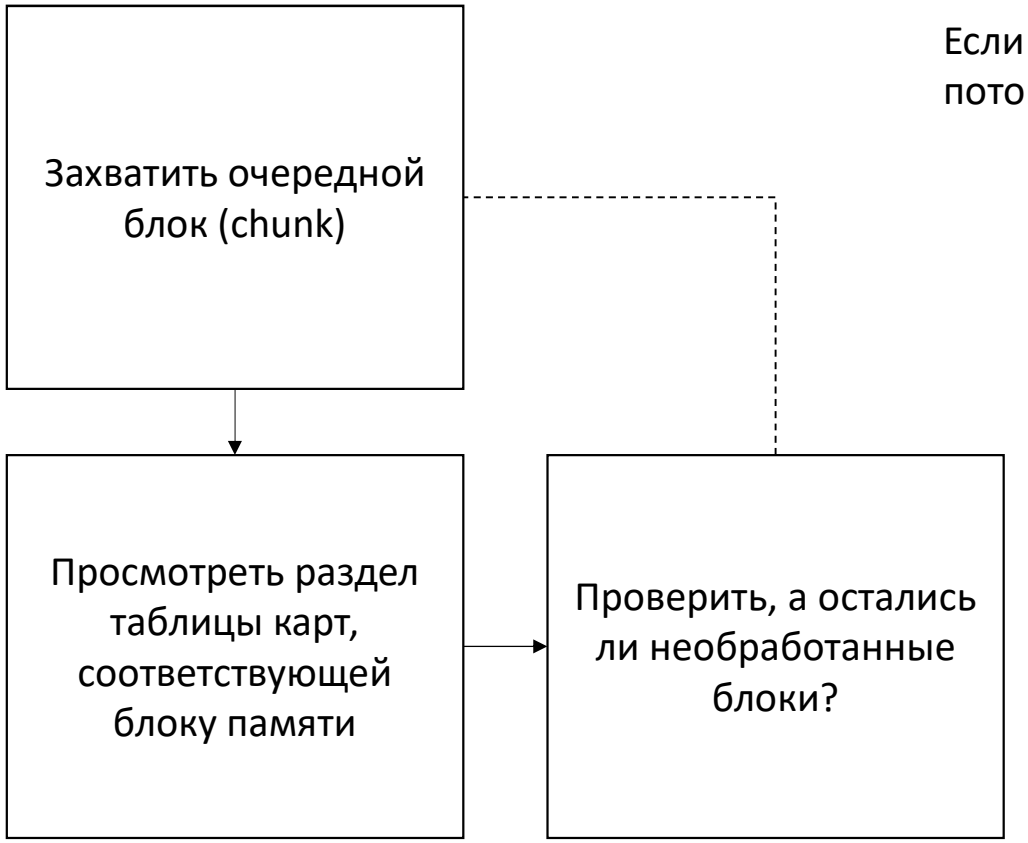
Источник: https://medium.com/@rakesh.upadhyaya (и многие другие)

Улучшения в mark_phase

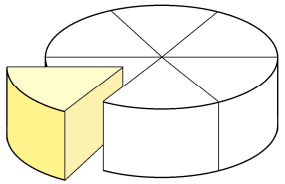




Улучшения в mark_phase

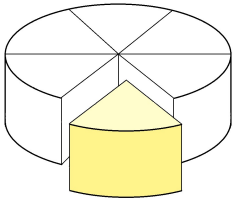


Если есть свободные блоки, можно отправить поток на их обработку

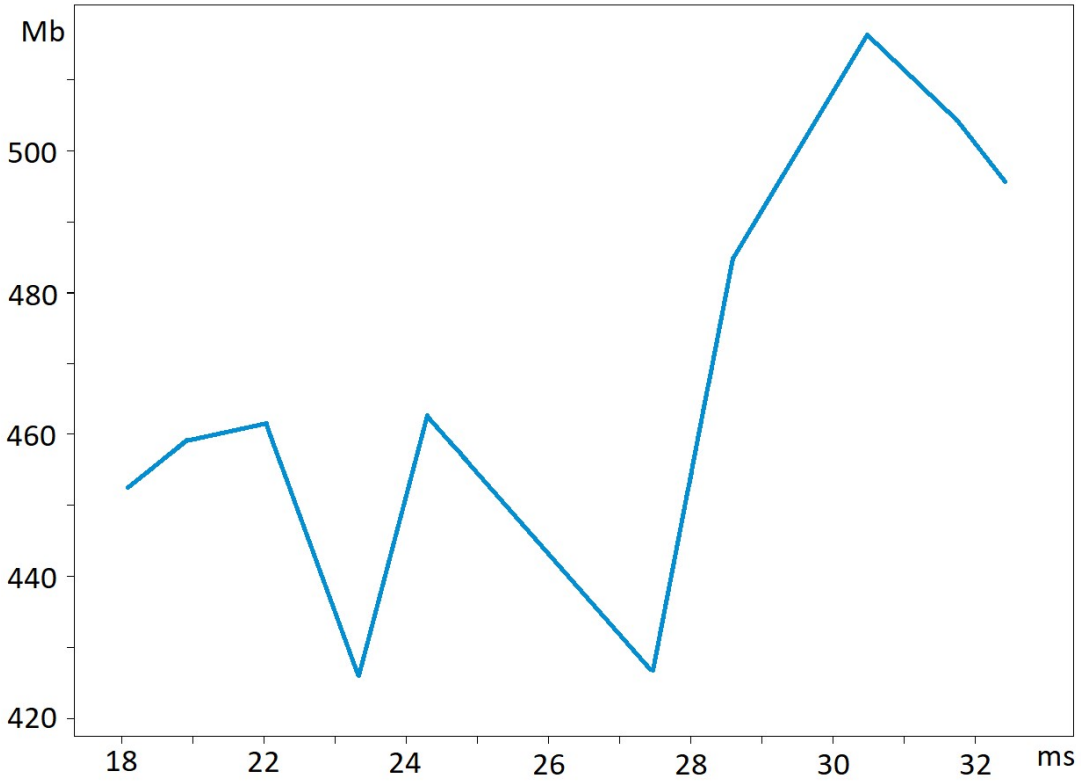


Оптимизация возвратов памяти

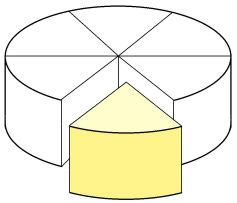
1. Балансировка работы потоков в mark_phase
2. Сглаживание частоты возвратов памяти ОС
3. Уменьшение частоты конфликтов при сканировании статики
4. Оптимизации в reset_memory
5. Уменьшение времени GCStopTheWorld
6. Использование vxsort



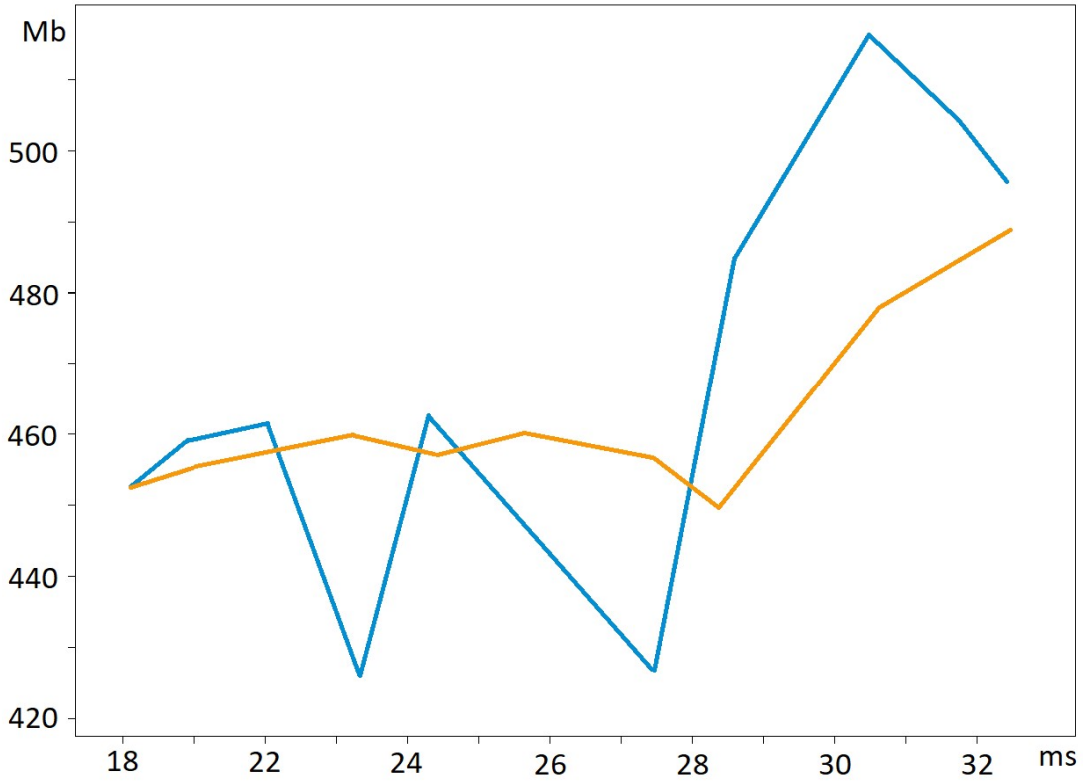
Оптимизация возвратов памяти



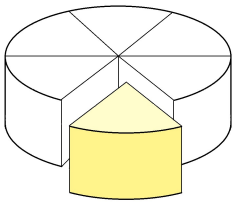
Объём захваченной памяти и его изменение с течением времени



Оптимизация возвратов памяти

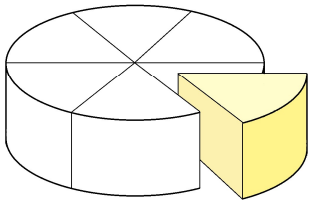


Объём захваченной памяти и его изменение с течением времени



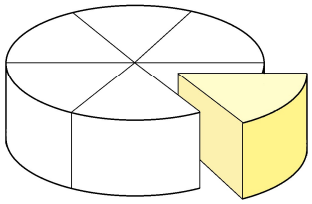
Сканирование статики

1. Балансировка работы потоков в mark_phase
2. Сглаживание частоты возвратов памяти ОС
3. Уменьшение частоты конфликтов при сканировании статики
4. Оптимизации в reset_memory
5. Уменьшение времени GCStopTheWorld
6. Использование vxsort



Сканирование статики

1. Ссылки, хранящиеся в стеке (локальные переменные)
2. Статические ссылки
3. Pinning handles
4. Finalizer references



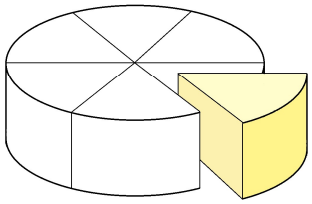
Сканирование статики

Можно просматривать статические ссылки не по всей сборке…

pDomainAssembly->EnumStaticGCRefs(fn, sc);

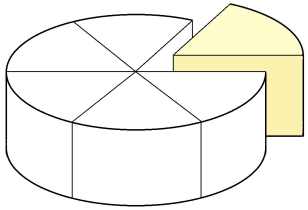
А по бакетам в таблице хендлеров кучи

m_pLargeHeapHandleTable->EnumStaticGCRefs(fn, sc);



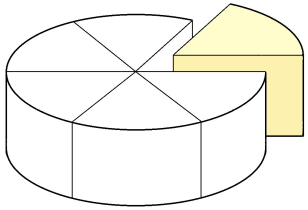
Оптимизация reset_memory

1. Балансировка работы потоков в mark_phase
2. Сглаживание частоты возвратов памяти ОС
3. Уменьшение частоты конфликтов при сканировании статики
4. Оптимизации в reset_memory
5. Уменьшение времени GCStopTheWorld
6. Использование vxsort

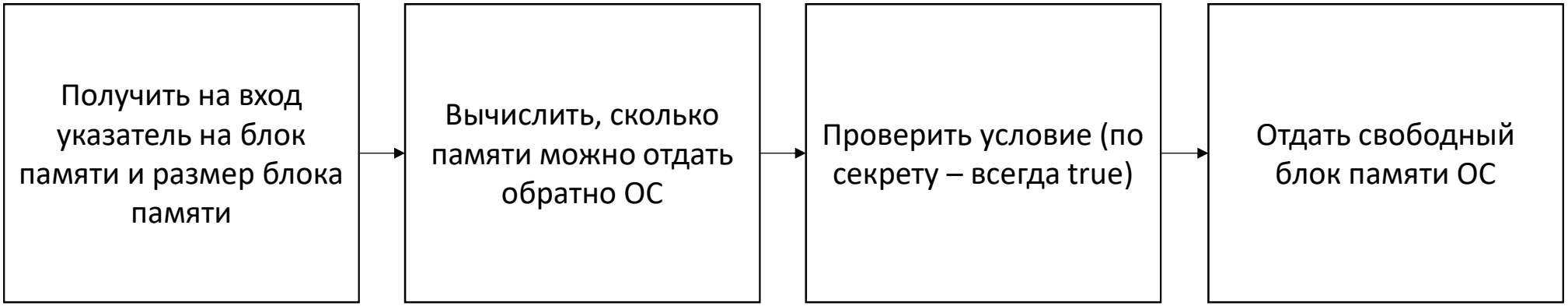


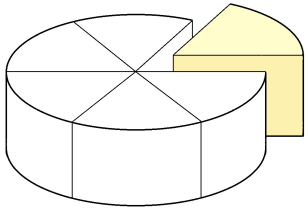
Оптимизация reset_memory

“Возвраты памяти ОС весьма дорогостоящи”

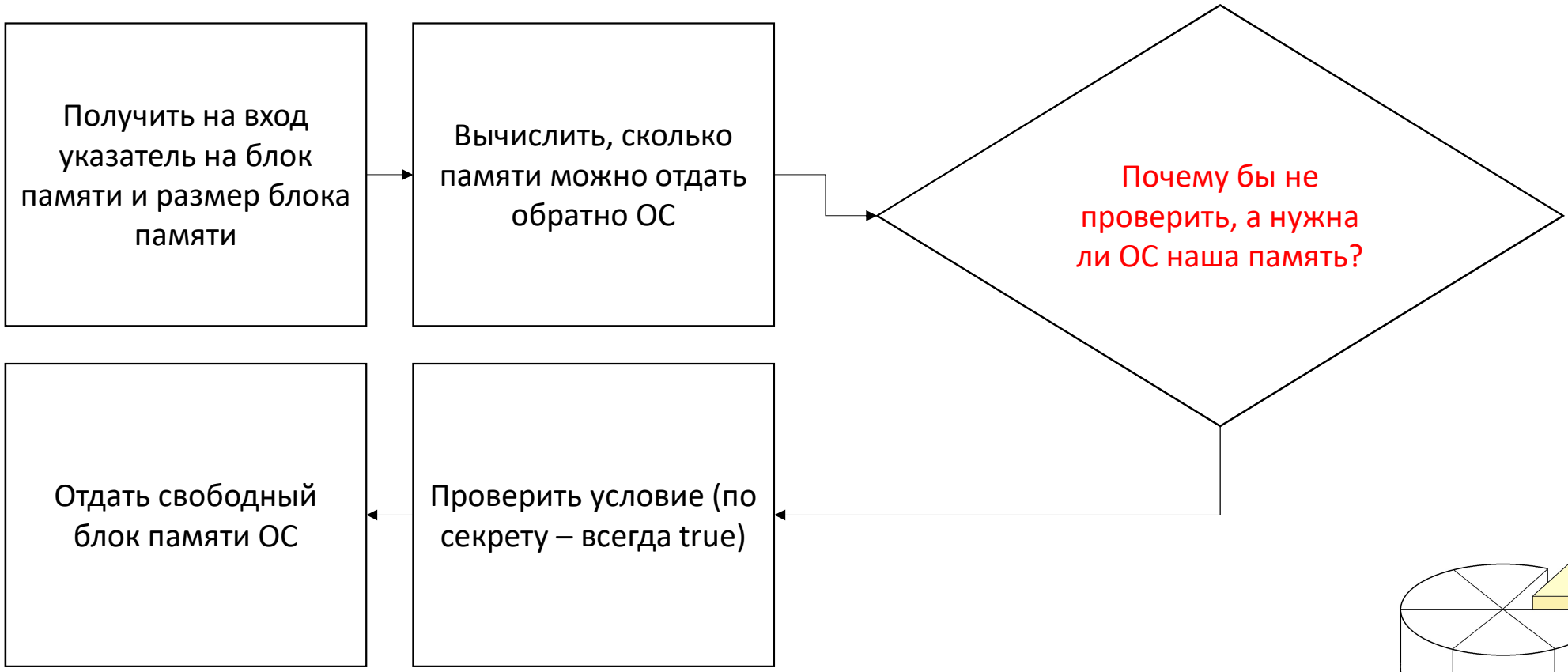


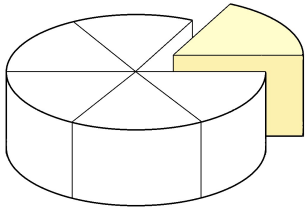
Оптимизация reset_memory





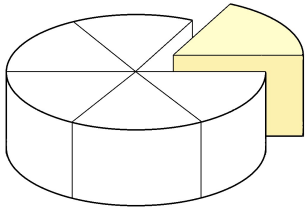
Оптимизация reset_memory





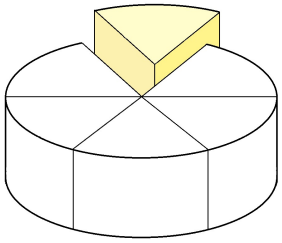
Оптимизация reset_memory

```
-           if (reset_mm_p)
+           if (reset_mm_p && gc_heap::g_low_memory_status)
```



И снова GCStopTheWorld

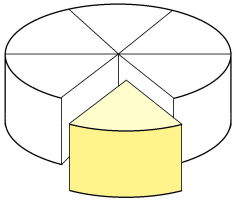
1. Балансировка работы потоков в mark_phase
2. Сглаживание частоты возвратов памяти ОС
3. Уменьшение частоты конфликтов при сканировании статики
4. Оптимизации в reset_memory
5. Уменьшение времени GCStopTheWorld
6. Использование vxsort



Оптимизация возвратов памяти

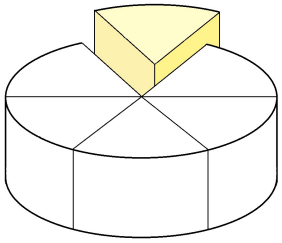
Два месяца назад

Два дня назад



И снова GCStopTheWorld

// give foreground GC a chance to run

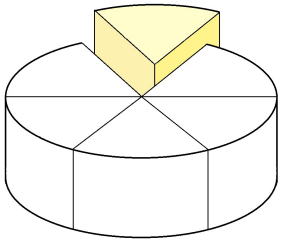


И снова GCStopTheWorld

В процессе разметки можно вызвать foreground GC:

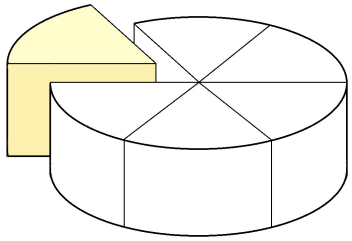
1. Когда посещаем страницу в поисках ссылок во второй раз

2. Когда обработали X байт памяти (X – вычисляется через эвристики, линейно зависит от количества обработанных ссылок)



Vxsort

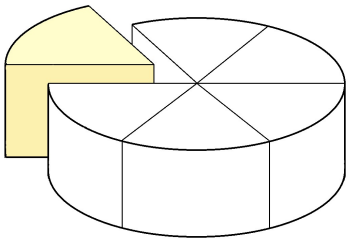
1. Балансировка работы потоков в mark_phase
2. Сглаживание частоты возвратов памяти ОС
3. Уменьшение частоты конфликтов при сканировании статики
4. Оптимизации в reset_memory
5. Уменьшение времени GCStopTheWorld
6. Использование vxsort



Vxsort

Introsort = (quicksort + heapsort + insertion sort)

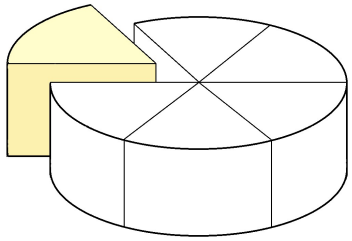
Vxsort = Vectorized quicksort + vectorized bitonic sort
(when AVX2 + is supported)



Vxsort

По сравнению с Introspective Sort

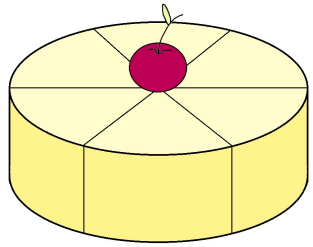
1M элементов int64	128K элементов int64
55 ns -> 11,5 ns	45 ns -> 10,5 ns
На каждый элемент	На каждый элемент
< в 4.8 раз	< в 4.5 раз



https://github.com/damageboy/vxsort-cpp/

Ещё что-нибудь?

А ещё в Microsoft продолжили и
даже ускорили переписывание кода на c#



Источник: https://devblogs.microsoft.com/dotnet/performance-improvements-in-net-5/

Поговорите со мной

Спасибо!

tg: @KinderGir

habr: janeryzhikova

jane.ryzhikova@gmail.com