What’s New in the .NET 5 GC?

Maoni Stephens

DotNext 2020 Moscow

Perf improvements with no user interaction

Improvements on fundamentals

Balancing work is crucial in Server GC

Perf improvements with no user interaction – cont.

- Ephemeral segment decommit
  - Completely done outside of the STW phase of a GC on a Server GC thread
  - Better decommit logic
- Card mark stealing
  - GC threads that finished marking cards on their heaps will steal from other GC threads
  - Benefits both the mark and the relocate phase

Perf improvements with no user interaction – cont.

- Vectorized mark list sorting
  - introsort (quicksort + heapsort + insertion sort) → Vectorized quicksort + vectorized bitonic sort (when AVX2+ is supported)
- Memory reset (MEM_RESET)
  - Only when memory load is high enough

You don't need to do anything!

User facing feature - POH

slides,
video

segments

POH API

```
T[] AllocateArray<T>(int length, bool pinned);
```

```
T[] AllocateUninitializedArray<T>(int length,
                                  bool pinned);
```

If pinned is true, the array cannot contain references or it'll throw an exception.

Related hardlimit configs

COMPlus_GCHeapHardLimitSOH

COMPlus_GCHeapHardLimitLOH

COMPlus_GCHeapHardLimitPOH

COMPlus_GCHeapHardLimitSOHPercent

COMPlus_GCHeapHardLimitLOHPercent

COMPlus_GCHeapHardLimitPOHPercent

Diagnostics - API

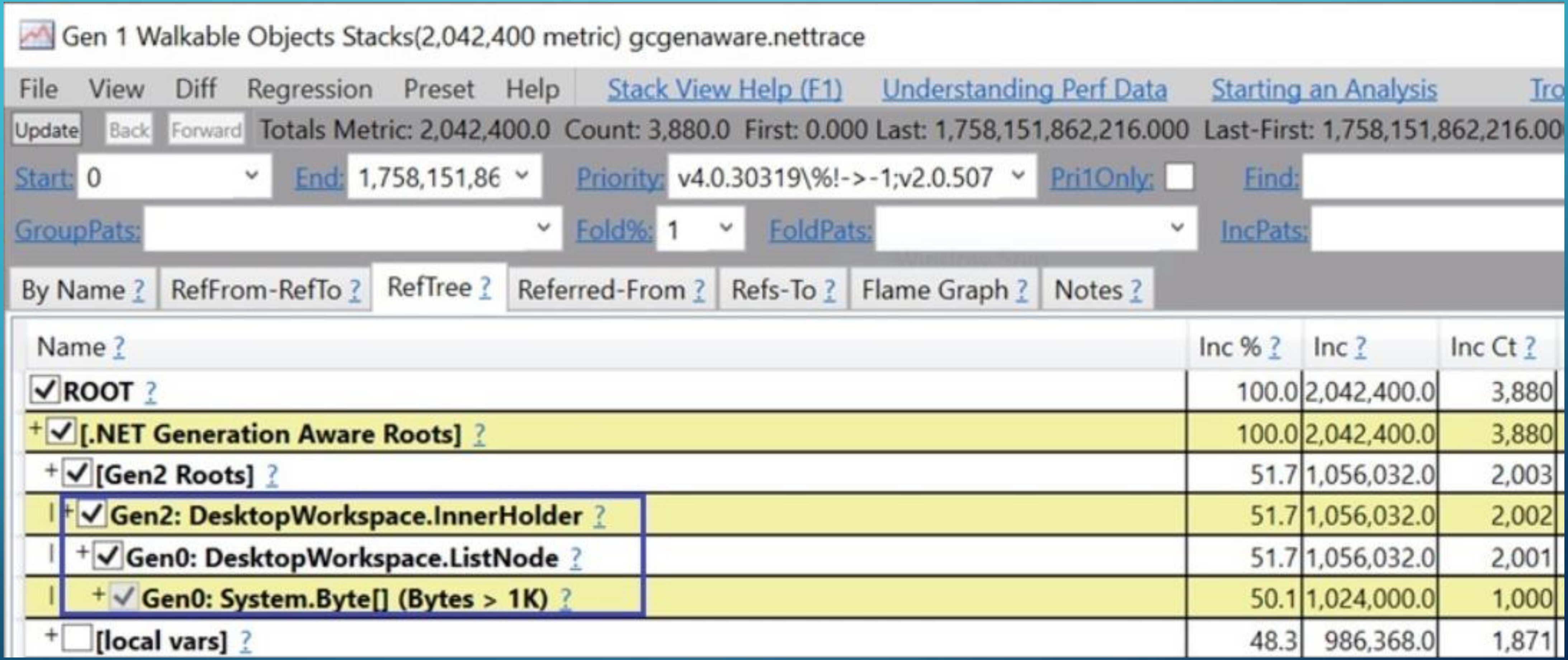
- New GC.GetGCMemoryInfo(), described in detail here
- In-proc can be a real convenience, eg, use it in container without a sidecar container to collect traces from.
- For both sampling and diagnostics

New properties

- Index of this GC
- Accumulative % pause time in GC
- Per generation data (before GC and after GC)
- Various GC attributes (compact/concurrent)
- Detailed pause durations
- Promoted, # of pinned objects, # objects promoted due to “ready for finalization”
- Total committed bytes

Diagnostics - generational aware tooling

- Why do we need generational aware?
  - Explained in detail in mem-doc
- Usage example
  - set COMPlus_GCGenAnalysisGen=1
  - set COMPlus_GCGenAnalysisBytes=100000
  - set COMPlus_GCGenAnalysisIndex=10
  - Optional: set COMPlus_EventPipeCircularMB=800



Gen 1 Walkable Objects Stacks(2,042,400 metric) gcgenaware.nettrace

Totals Metric: 2,042,400.0 Count: 3,880.0 First: 0.000 Last: 1,758,151,862,216.000 Last-First: 1,758,151,862,216.00

Start: 0 End: 1,758,151,86 Priority: v4.0.30319\%!->-1;v2.0.507 Pri1Only: □ Find:

GroupPats: Fold%: 1 FoldPats: IncPats:

Name ?	Inc % ?	Inc ?	Inc Ct ?
☑ ROOT ?	100.0	2,042,400.0	3,880
+ ☑ [.NET Generation Aware Roots] ?	100.0	2,042,400.0	3,880
+ ☑ [Gen2 Roots] ?	51.7	1,056,032.0	2,003
+ ☑ Gen2: DesktopWorkspace.InnerHolder ?	51.7	1,056,032.0	2,002
+ ☑ Gen0: DesktopWorkspace.ListNode ?	51.7	1,056,032.0	2,001
+ ☑ Gen0: System.Byte[] (Bytes > 1K) ?	50.1	1,024,000.0	1,000
+ □ [local vars] ?	48.3	986,368.0	1,871

Resources

- mem-doc
  - Please read this if you need to perform memory analysis on .NET!
  - Many of the concepts/terms in this talk are explained in detail in the doc
- Ask me questions at http://twitter.com/maoni0

(response time may be very inconsistent)

- File an issue on our repo: http://github.com/dotnet/runtime
- Pro .NET Memory Management book

I HAZ A QUESTION