RTX64

Local Memory Performance Comparison

This document provides a performance comparison between local memory allocation and Windows memory allocation.

RTX64 4.0 implemented a completely new local memory architecture. There is no longer one global local memory pool but multiple separate internal and external memory allocation spaces (MSpaces). With this new architecture there has been a dramatic improvement in the determinism of memory allocations and frees. The data below compares Windows memory performance with RTX64 4.x and 3.x local memory performance.

Testing Environment

System	Intel Core i9-7900X CPU @3.30GHz	
RTX64 Control Panel customizations	**RTX64 4.x**	**RTX64 3.x**
	Turn off *Zero memory on allocation*	Select *Use local memory*
RtssRun parameters	**RTX64 4.x**	**RTX64 3.x**
	RtssRun /i 4194292 /e 0 test.rtss	RtssRun test.rtss
	/i initial_size The initial amount of local memory (allocated at process startup), in kilobytes, within the process MSpace.	
	/e expand_size The amount of local memory, in kilobytes, by which to expand the process MSpace. A value of 0 disables automatic expansion of the process MSpace.	

Added in RTX64 4.1

/r start, firstalloc

The time of allocation of the process external MSpace.
This switch requires one of these valid arguments:

- start - allocates the MSpace at process start
- firstalloc - allocates the MSpace at the first memory request

Source

The source file used for this comparison, RTX64_Local_Memory_Performance.cpp, is available for download from the Support site.

Time to Allocate

Below is a comparison of local memory versus Windows memory of the time to allocate memory of different sizes, in nanoseconds, using C Runtime function malloc.

Size (bytes)	Windows memory	RTX64 3.x local memory	RTX64 4.x local memory
4194304	9.738229e+03 (ns)	3.754353e+05 (ns)	1.151443e+03 (ns)
2097152	5.387196e+03 (ns)	1.867993e+05 (ns)	7.734124e+02 (ns)
1048576	3.726730e+03 (ns)	9.431061e+04 (ns)	6.273231e+02 (ns)
524288	3.403309e+03 (ns)	4.840634e+04 (ns)	5.675217e+02 (ns)

Size (bytes)	Windows memory	RTX64 3.x local memory	RTX64 4.x local memory
262144	2.527113e+03 (ns)	2.385355e+04 (ns)	5.886057e+02 (ns)
131072	2.267923e+03 (ns)	1.204788e+04 (ns)	4.882693e+02 (ns)
65536	1.786178e+03 (ns)	6.288335e+03 (ns)	4.132421e+02 (ns)
32768	1.109033e+03 (ns)	3.603063e+03 (ns)	3.878605e+02 (ns)
16384	1.019030e+03 (ns)	2.401609e+03 (ns)	1.775531e+02 (ns)
8192	9.850308e+02 (ns)	1.843240e+03 (ns)	9.158333e+01 (ns)
4096	8.968472e+02 (ns)	1.505717e+03 (ns)	6.433333e+01 (ns)
2048	4.295598e+02 (ns)	1.097725e+03 (ns)	6.451812e+01 (ns)
1024	2.375393e+02 (ns)	9.312101e+02 (ns)	8.139553e+01 (ns)
512	1.434559e+02 (ns)	7.859831e+02 (ns)	4.982428e+01 (ns)
256	1.197506e+02 (ns)	7.491413e+02 (ns)	9.031643e+01 (ns)
128	8.539010e+01 (ns)	7.061655e+02 (ns)	2.575604e+01 (ns)
64	4.021377e+01 (ns)	6.997458e+02 (ns)	2.315399e+01 (ns)
32	4.290278e+01 (ns)	6.978684e+02 (ns)	2.301449e+01 (ns)
16	3.519143e+01 (ns)	6.960556e+02 (ns)	2.338768e+01 (ns)

Time to Free

Below is a comparison of local memory versus Windows memory of the time to free memory of different sizes, in nanoseconds, using C Runtime function malloc.

Size (bytes)	Windows memory	RTX64 3.x local memory	RTX64 4.x local memory
4194304	4.112605e+04 (ns)	1.889739e+03 (ns)	9.937319e+02 (ns)
2097152	2.152816e+04 (ns)	1.444077e+03 (ns)	8.180930e+02 (ns)
1048576	1.177847e+04 (ns)	1.154293e+03 (ns)	5.811775e+02 (ns)
524288	2.815152e+03 (ns)	1.147034e+03 (ns)	4.139511e+02 (ns)
262144	2.023338e+03 (ns)	8.473418e+02 (ns)	2.757748e+02 (ns)
131072	1.345579e+03 (ns)	8.631250e+02 (ns)	2.047120e+02 (ns)
65536	9.775628e+02 (ns)	8.024106e+02 (ns)	2.243593e+02 (ns)
32768	8.577470e+02 (ns)	7.860803e+02 (ns)	2.035175e+02 (ns)
16384	7.175637e+02 (ns)	8.557935e+02 (ns)	1.166129e+02 (ns)
8192	4.400477e+02 (ns)	7.699921e+02 (ns)	1.034777e+02 (ns)
4096	2.671546e+02 (ns)	7.465531e+02 (ns)	5.117633e+01 (ns)

Size (bytes)	Windows memory	RTX64 3.x local memory	RTX64 4.x local memory
2048	1.470109e+02 (ns)	7.472687e+02 (ns)	3.894867e+01 (ns)
1024	8.696014e+01 (ns)	7.049976e+02 (ns)	2.668418e+01 (ns)
512	5.783696e+01 (ns)	6.854638e+02 (ns)	2.600966e+01 (ns)
256	2.446981e+01 (ns)	6.671135e+02 (ns)	2.672886e+01 (ns)
128	2.153019e+01 (ns)	6.701359e+02 (ns)	2.595652e+01 (ns)
64	2.058575e+01 (ns)	6.826021e+02 (ns)	2.664674e+01 (ns)
32	2.080495e+01 (ns)	7.039994e+02 (ns)	2.591908e+01 (ns)
16	2.379408e+01 (ns)	7.048333e+02 (ns)	2.680978e+01 (ns)