Windows Kernel Internals
User-mode Heap Manager

David B. Probert, Ph.D.
Windows Kernel Development
Microsoft Corporation
Topics

• Common problems with the NT heap
• LFH design
• Benchmarks data
• Heap analysis
Default NT Heap

• Unbounded fragmentation for the worst scenario:
  – External fragmentation
  – Virtual address fragmentation

• Poor performance for:
  – Large heaps
  – SMP
  – Large blocks
  – Fast growing scenarios
  – Fragmented heaps
Goals For LFH

- Bounded low fragmentation
- Low risk (minimal impact)
- Stable and high performance for:
  - Large heaps
  - Large blocks
  - SMP
  - Long running applications
LFH Design

• Bucket-oriented heap
• Better balance between internal and external fragmentation
• Improved data locality
• No locking for most common paths
Tradeoffs

- Performance / footprint
- Internal / external fragmentation
- Thread / processor data locality
- Using prefetch techniques
Block Size

0 1K 16K 512K

LFH

NT Heap

NT Memory Manager
## Allocation Granularity

<table>
<thead>
<tr>
<th>Block Size</th>
<th>Granularity</th>
<th>Buckets</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>512</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>1024</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>2048</td>
<td>64</td>
<td>16</td>
</tr>
<tr>
<td>4096</td>
<td>128</td>
<td>16</td>
</tr>
<tr>
<td>8196</td>
<td>256</td>
<td>16</td>
</tr>
<tr>
<td>16384</td>
<td>512</td>
<td>16</td>
</tr>
</tbody>
</table>
Alloc

Active segment

User data area

Descriptor

Segment queue

Unmanaged segments
Free

Active segment

Segment queue

Unmanaged segments
Free

Active segment

Unmanaged segments

Segment queue
Improving the SMP Scalability

- Thread locality
- Processor locality
Thread Data Locality

• Advantages
  – Easy to implement (TLS)
  – Can reduce the number of interlocked instructions

• Disadvantages
  – Significantly larger footprint for high number of threads
  – Common source of leaks (the cleanup is not guaranteed)
  – Larger footprint for scenarios involving cross thread operations
  – Performance issues at low memory (larger footprint can cause paging)
  – Increases the CPU cost per thread creation / deletion
Processor Locality

• Advantages
  – The memory footprint is bounded to the number of CPUs regardless of the number of threads
  – Expands the structures only if needed
  – No cleanup issues

• Disadvantages
  – The current CPU is not available in user mode
  – Not efficient for a large number of processors and few threads
MP Scalability

Affinity manager

Large segments cache

NT Heap

Descriptive cache

16 K
Better Than Lookaside

- Better data locality (likely in same page)
- Almost perfect SMP scalability (no false sharing)
- Covers a larger size range (up to 16k blocks)
- Works well regardless of the number of blocks
- Non-blocking operations even during growing and shrinking phases
Benchmarks

• Fragmentation
• Speed
• Scalability
• Memory efficiency
# Fragmentation test for 266 MB limit

<table>
<thead>
<tr>
<th></th>
<th>Default</th>
<th>LFH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncommited</td>
<td>235 MB</td>
<td>39 MB</td>
</tr>
<tr>
<td>Free</td>
<td>4 MB</td>
<td>7 MB</td>
</tr>
<tr>
<td>Busy</td>
<td>26 MB</td>
<td>224 MB</td>
</tr>
<tr>
<td>Fragmentation</td>
<td><strong>88%</strong></td>
<td><strong>14%</strong></td>
</tr>
</tbody>
</table>
Default NT Heap

- 88% Uncommitted
- 10% Busy
- 2% Free
Low Fragmentation Heap

- Uncommitted: 83%
- Free: 3%
- Busy: 14%
# External Fragmentation Test (70 MB)

<table>
<thead>
<tr>
<th></th>
<th>Default</th>
<th>LFH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncommited</td>
<td>25 MB</td>
<td>7 MB</td>
</tr>
<tr>
<td>Free</td>
<td>32 MB</td>
<td>8 MB</td>
</tr>
<tr>
<td>Busy</td>
<td>12 MB</td>
<td>46 MB</td>
</tr>
<tr>
<td>Fragmentation</td>
<td><strong>46% + 36%</strong></td>
<td><strong>14% + 12%</strong></td>
</tr>
</tbody>
</table>
NT Heap at 70 M usage
(8478 UCR, 10828 free blocks)

- Uncommitted: 18%
- Free: 46%
- Busy: 36%
Low Fragmentation Heap at 70 M
(417 UCR, 1666 free blocks)
Replacement test
0-1k, 10000 blocks (4P x 200MHz)

Graph showing the relationship between Threads and Alocs/sec for LFH and NT.
Replacement test
0-1k, 10000 blocks

Mem eff.

Threads

LFH
NT
Replacement test
1-2k, 10000 blocks

Alocs/sec

Threads

LFH

NT
Replacement test
1-2k, 10000 blocks

![Graph showing the relationship between Mem eff. and Threads, with Mem eff. ranging from 0 to 1.8 and Threads ranging from 1 to 32. The graph compares LFH and NT.]
Replacement test on a 32P machine
0-1k, 100000 blocks

![Graph showing performance comparison between different thread counts and operations per second (Ops/sec) with log scale for both axes. The graph includes lines for LFH, NT, and Ideal.]
Replacement test on 32P machine
0-1k, 100000 blocks
Replacement test on 32P machine
22 bytes, 100000 blocks

<table>
<thead>
<tr>
<th>Threads (log)</th>
<th>Ops/sec (log)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000000</td>
</tr>
<tr>
<td>2</td>
<td>10000000</td>
</tr>
<tr>
<td>4</td>
<td>100000000</td>
</tr>
<tr>
<td>8</td>
<td>1000000000</td>
</tr>
<tr>
<td>16</td>
<td>10000000000</td>
</tr>
<tr>
<td>32</td>
<td>100000000000</td>
</tr>
<tr>
<td>64</td>
<td>1000000000000</td>
</tr>
<tr>
<td>128</td>
<td>10000000000000</td>
</tr>
<tr>
<td>256</td>
<td>100000000000000</td>
</tr>
<tr>
<td>512</td>
<td>1000000000000000</td>
</tr>
</tbody>
</table>

Threads (log)

Ops/sec (log)

LFH
NT
Ideal
Replacement test on 32P machine
1k-2k, 100000 blocks
Replacement test on 32P machine
1k-2k, 100000 blocks

![Graph showing memory efficiency (Mem. Eff.) vs. number of threads (Threads (log)). The graph compares LFH and NT methods. The memory efficiency increases with the number of threads, especially for LFH, while NT remains relatively constant.](image-url)
Larson MT test on 32P machine
0 - 1k, 3000 blocks/thread

Ops/sec vs Threads

LFH
NT
Ideal
Larson MT test on 32P machine
0 - 1k, 3000 blocks/thread

Graph showing the performance of different operations per second (Ops/sec) for various threads, with categories LFH, NT, and Ideal.
Larson MT test on 32P machine
1k -2k, 100000 blocks

![Graph showing performance metrics](image-url)
Larson MT test on 32P machine
1k -2k, 100000 blocks

![Graph showing performance of different thread configurations.](image-url)

**Y-axis**: Ops/sec (log)

**Legend**:
- LFH
- NT
- Ideal
Aggressive alloc test on 32P machine
50 Mbytes allocs in blocks of 32 bytes

![Graph showing time vs threads for LFH and NT.](Image)
When is the Default Heap Preferred

- ~95% of applications
- The heap operations are rare
- Low memory usage
Where LFH is Recommended

- High memory usage and:
  - High external fragmentation (> 10-15%)
  - High virtual address fragmentation (>10-15%)
- Performance degradation on long run
- High heap lock contention
- Aggressive usage of large blocks (> 1K)
Activating LFH

- **HeapSetInformation**
  - Can be called any time after the heap creation
  - Restriction for some flags (HEAP_NO_SERIALIZE, debug flags)
  - Can be destroyed only with the entire heap

- **HeapQueryInformation**
  - Retrieve the current front end heap type
    - 0 – none
    - 1 – lookaside
    - 2 – LFH
Heap Analysis

• !heap to collect statistics and validate the heap
  – !heap –s
  – !heap –s heap_addr –b8
  – !heap –s heap_addr –d40

• Perfmon
### Overall Heap Stats

```plaintext
0:001> !heap -s

<table>
<thead>
<tr>
<th>Heap</th>
<th>Flags</th>
<th>Reserv (k)</th>
<th>Commit (k)</th>
<th>Virt (k)</th>
<th>Free (k)</th>
<th>List length</th>
<th>UCR</th>
<th>Virt blocks</th>
<th>Lock</th>
<th>Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>00080000 00000002</td>
<td>1024</td>
<td>28</td>
<td>28</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>L</td>
<td></td>
</tr>
<tr>
<td>00180000 00000000</td>
<td>64</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00250000 00000000</td>
<td>64</td>
<td>24</td>
<td>24</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>L</td>
<td></td>
</tr>
<tr>
<td>00270000 00000000</td>
<td>130304</td>
<td>58244</td>
<td>96888</td>
<td>36722</td>
<td>10828</td>
<td>8478</td>
<td>0</td>
<td>0</td>
<td>L</td>
<td></td>
</tr>
</tbody>
</table>

External fragmentation 63 % (10828 free blocks)
Virtual address fragmentation 39 % (8478 uncommitted ranges)
```
### Overall Heap Stats

```
0:000> !heap -s

<table>
<thead>
<tr>
<th>Heap</th>
<th>Flags</th>
<th>Reserv (k)</th>
<th>Commit (k)</th>
<th>Virt (k)</th>
<th>Free (k)</th>
<th>List length</th>
<th>UCR</th>
<th>Virt blocks</th>
<th>Lock</th>
<th>Fast</th>
</tr>
</thead>
<tbody>
<tr>
<td>00080000</td>
<td>00000002</td>
<td>1024</td>
<td>28</td>
<td>28</td>
<td>16</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>00180000</td>
<td>00008000</td>
<td>64</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>00250000</td>
<td>00001002</td>
<td>64</td>
<td>24</td>
<td>24</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>00270000</td>
<td>00001002</td>
<td>256</td>
<td>116</td>
<td>116</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>002b0000</td>
<td>00001002</td>
<td>130304</td>
<td>122972</td>
<td>122972</td>
<td>1936</td>
<td>67</td>
<td>1</td>
<td>0</td>
<td>14d5b8</td>
<td></td>
</tr>
</tbody>
</table>

Lock contention 1365432
```

---

The table shows the overall heap stats with various columns indicating different memory allocations and statuses. The `Lock contention` value indicates the lock contention count, which is 1365432 in this case.
Overall Heap Stats

0:006> !heap -s

The process has the following heap extended settings 00000008:
- Low Fragmentation Heap activated for all heaps

Affinity manager status:
- Virtual affinity limit 8
- Current entries in use 4
- Statistics: Swaps=18, Resets=0, Allocs=18

<table>
<thead>
<tr>
<th>Heap</th>
<th>Flags</th>
<th>Reserv</th>
<th>Commit</th>
<th>Virt</th>
<th>Free</th>
<th>List</th>
<th>UCR</th>
<th>Virt</th>
<th>Lock</th>
<th>Cont.</th>
<th>Fast</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00080000</td>
<td>00000002</td>
<td>1024</td>
<td>432</td>
<td>432</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td>0</td>
<td>LFH</td>
</tr>
<tr>
<td>00180000</td>
<td>00008000</td>
<td>64</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>00250000</td>
<td>00001002</td>
<td>1088</td>
<td>364</td>
<td>364</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td>0</td>
<td>LFH</td>
</tr>
<tr>
<td>00370000</td>
<td>00001002</td>
<td>256</td>
<td>212</td>
<td>212</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td>0</td>
<td>LFH</td>
</tr>
<tr>
<td>003b0000</td>
<td>00001002</td>
<td>7424</td>
<td>5720</td>
<td>6240</td>
<td>43</td>
<td>3</td>
<td>26</td>
<td>0</td>
<td>f</td>
<td></td>
<td></td>
<td>LFH</td>
</tr>
</tbody>
</table>
Default NT Heap Side

0:006> !heap -s 003b0000

Walking the heap 003b0000 ....

0:  Heap 003b0000
  Flags       00001002 - HEAP_GROWABLE
  Reserved    7424 (k)
  Commited    5720 (k)
  Virtual bytes 6240 (k)
  Free space  43 (k)
  External fragmentation          0% (3 free blocks)
  Virtual address fragmentation   8% (26 uncommited ranges)
  Virtual blocks 0
  Lock contention 15
  Segments 4
  2432 hash table for the free list
    Commits 0
    Decommmits 0

... Page 1/3
LFH Heap Side

Low fragmentation heap  003b0688
Lock contention        4
Metadata usage     76800
Statistics:
  Segments created       2236
  Segments deleted       733
  Segments reused         0
  Conversions             0
  ConvertedSpace         0

Block cache:
  Free blocks             0
  Sequence                 0
  Cache blocks 0 0 14 37 70 74 19
  Available 0 0 79 252 517 795 74

... Page 2/3
Default NT Heap Side

0:006> !heap -s 003b0000

Walking the heap 003b0000 ....

0: Heap 003b0000
   Flags          00001002 - HEAP_GROWABLE
   Reserved       7424 (k)
   Committed      5720 (k)
   Virtual bytes  6240 (k)
   Free space     43 (k)
   External fragmentation          0% (3 free blocks)
   Virtual address fragmentation   8% (26 uncommitted ranges)
   Virtual blocks  0
   Lock contention 15
   Segments        4
   2432 hash table for the free list
      Commits 0
      Decommitts 0
## Blocks Distribution

<table>
<thead>
<tr>
<th>Range (bytes)</th>
<th>Default heap</th>
<th>Front heap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Busy</td>
<td>Free</td>
</tr>
<tr>
<td>0 - 1024</td>
<td>18</td>
<td>83</td>
</tr>
<tr>
<td>1024 - 2048</td>
<td>113</td>
<td>0</td>
</tr>
<tr>
<td>2048 - 3072</td>
<td>70</td>
<td>1</td>
</tr>
<tr>
<td>4096 - 5120</td>
<td>74</td>
<td>0</td>
</tr>
<tr>
<td>8192 - 9216</td>
<td>19</td>
<td>2</td>
</tr>
<tr>
<td>16384 - 17408</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>32768 - 33792</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>104448 - 105472</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>312</td>
<td>86</td>
</tr>
</tbody>
</table>
Discussion