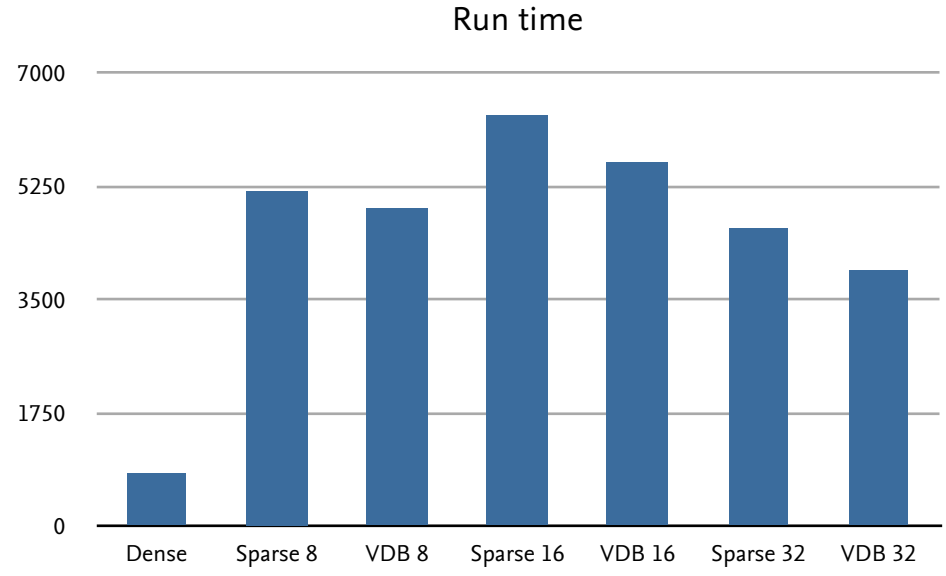
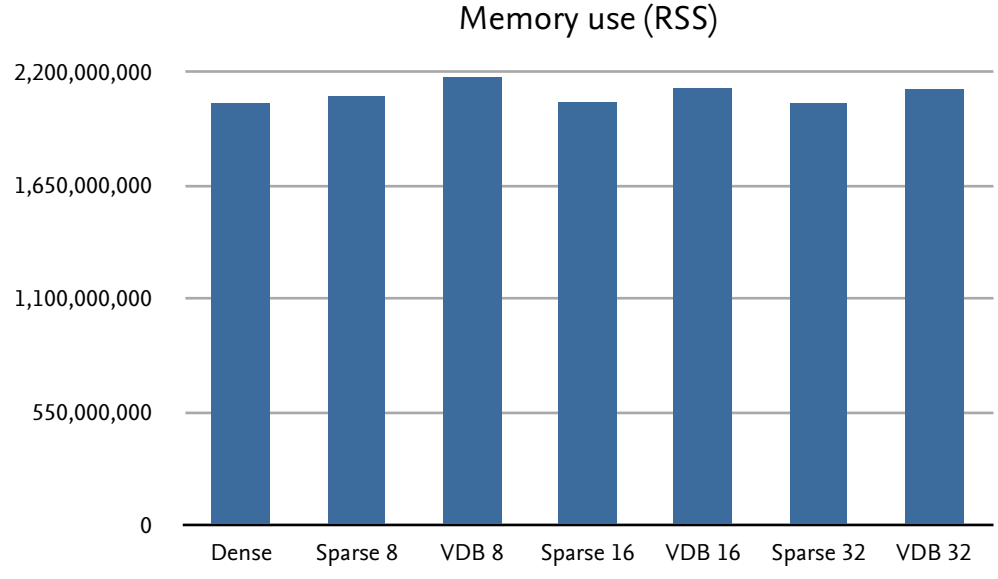
## Contiguous write access $800^3$

| Data structure | (ms) | (bytes) |
|---|---|---|
| Dense | 814 | 2,048,204,800 |
| Sparse 8 | 5177 | 2,080,235,520 |
| VDB 8 | 4904 | 2,172,227,584 |
| Sparse 16 | 6339 | 2,052,349,952 |
| VDB 16 | 5623 | 2,120,822,784 |
| Sparse 32 | 4594 | 2,048,884,736 |
| VDB 32 | 3945 | 2,113,699,840 |



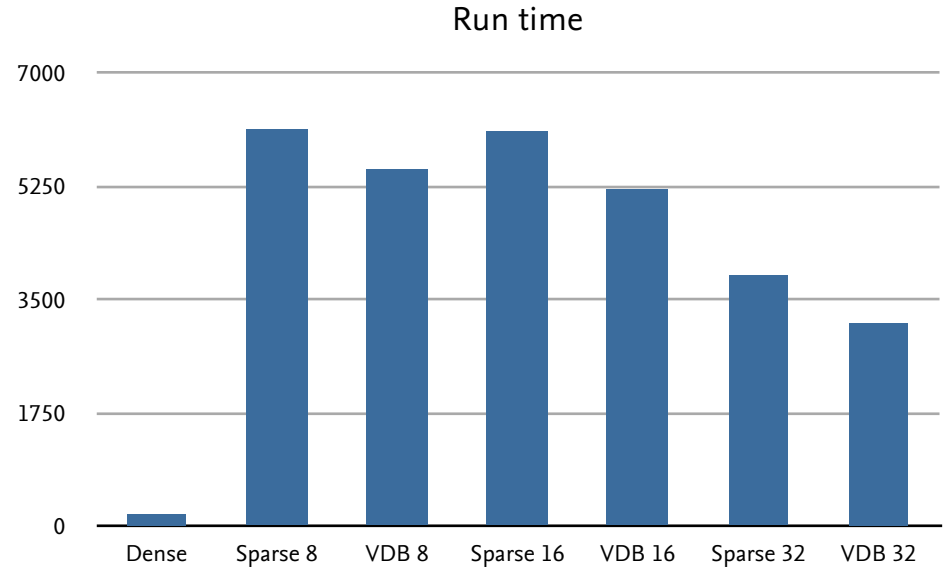

**About the test:**

This test writes to memory in a linear fashion, and does not make any attempts to access data in a cache-friendly way. It is representative of traversing the buffer naïvely.



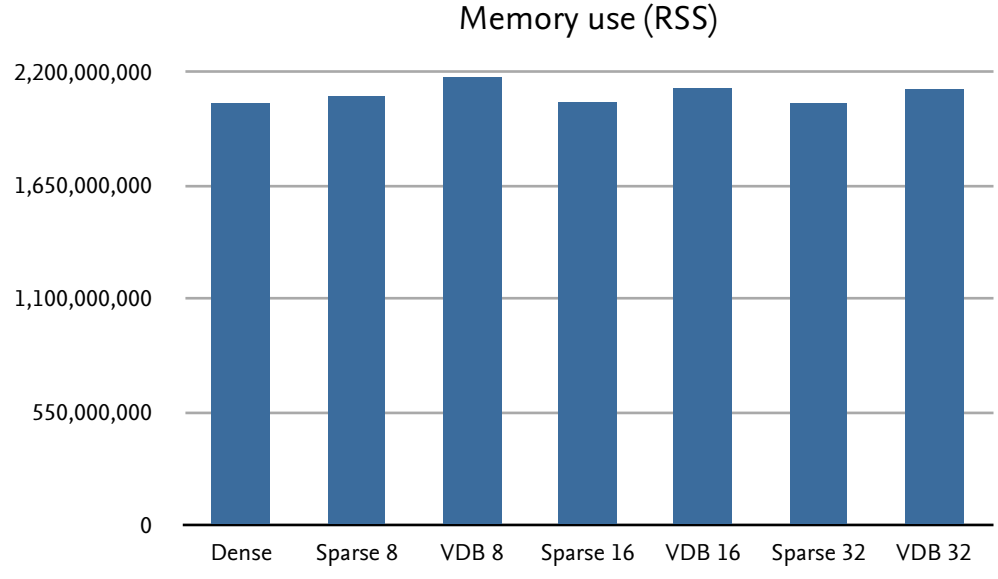
## Contiguous write access $800^3$ (Preallocated)

| Data structure | (ms) | (bytes) |
|---|---|---|
| Dense | 193 | 2,048,819,200 |
| Sparse 8 | 6127 | 2,080,415,744 |
| VDB 8 | 5506 | 2,172,313,600 |
| Sparse 16 | 6102 | 2,052,435,968 |
| VDB 16 | 5198 | 2,120,896,512 |
| Sparse 32 | 3875 | 2,048,958,464 |
| VDB 32 | 3141 | 2,113,769,472 |



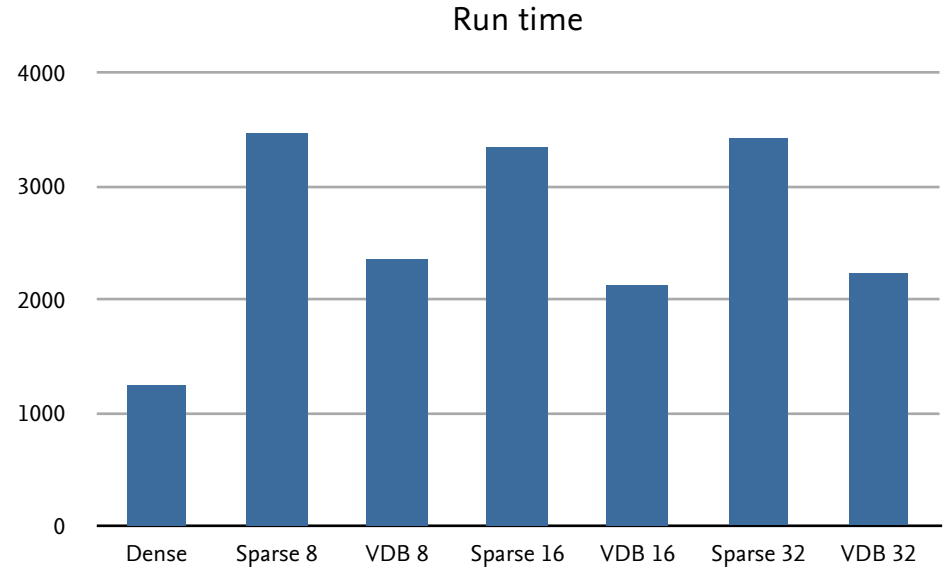

**About the test:**

Same as the previous test, but pre-allocates the memory used, so that only the traversal time is measured. It is representative of traversing the buffer naïvely.



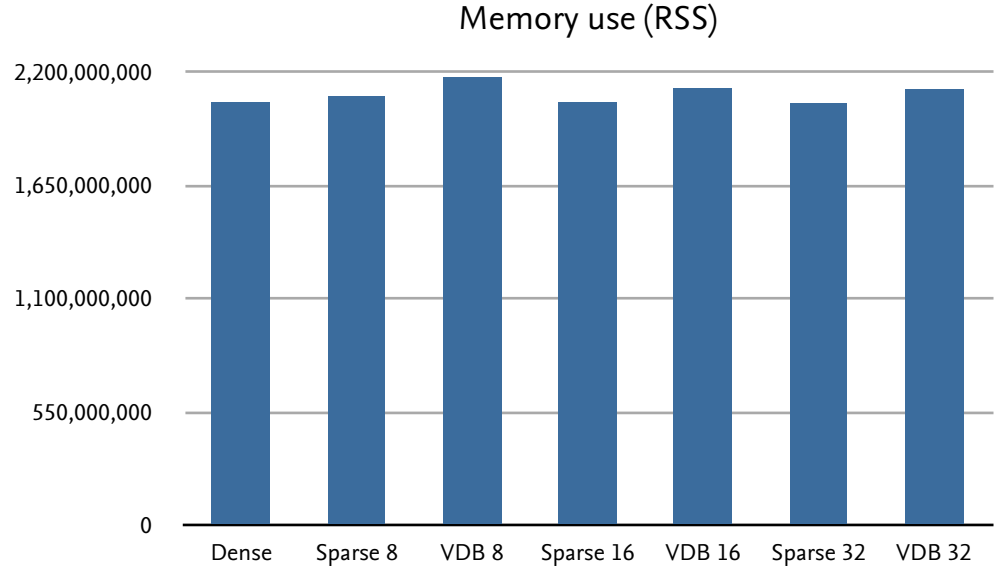
# Contiguous read access $800^3$

| Data structure | (ms) | (bytes) |
|---|---|---|
| Dense | 1247 | 2,049,114,112 |
| Sparse 8 | 3470 | 2,080,477,184 |
| VDB 8 | 2359 | 2,172,358,656 |
| Sparse 16 | 3347 | 2,052,481,024 |
| VDB 16 | 2129 | 2,120,921,088 |
| Sparse 32 | 3422 | 2,048,983,040 |
| VDB 32 | 2231 | 2,113,781,760 |



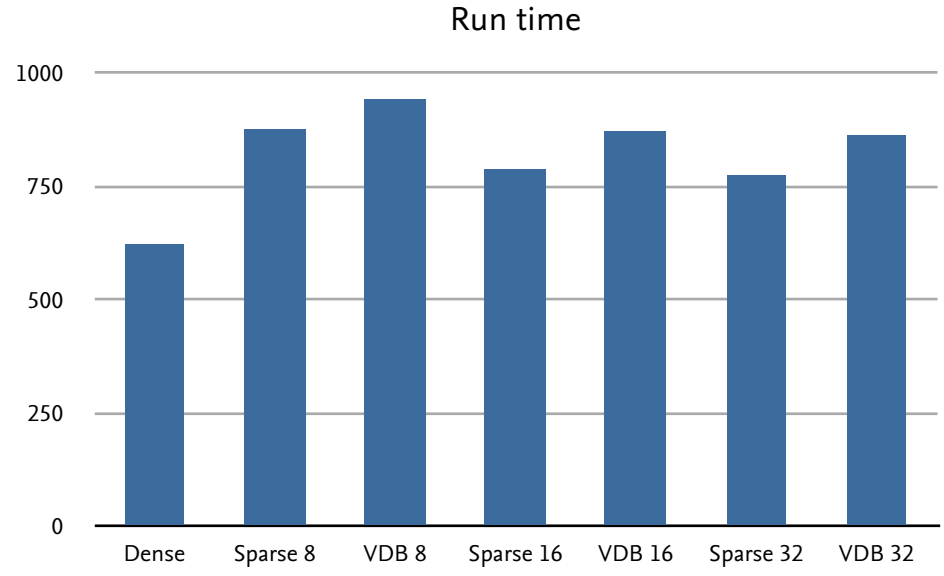

**About the test:**

This test reads from memory in a linear fashion, and does not make any attempts to access data in a cache-friendly way. It is representative of traversing the buffer naïvely.



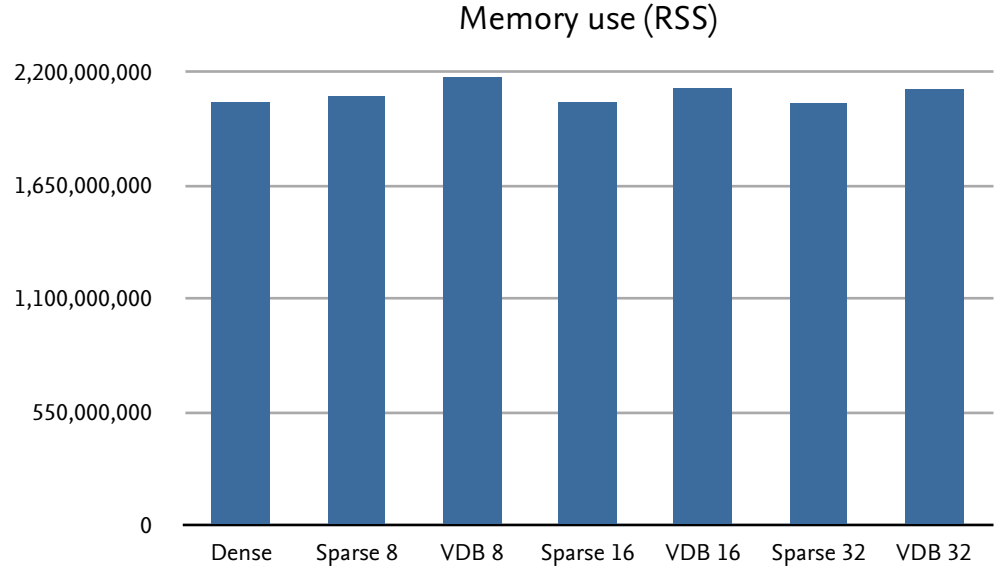
## Memory coherent write access $800^3$

| Data structure | (ms) | (bytes) |
|---|---|---|
| Dense | 621 | 2,049,122,304 |
| Sparse 8 | 876 | 2,080,497,664 |
| VDB 8 | 941 | 2,172,370,944 |
| Sparse 16 | 788 | 2,052,497,408 |
| VDB 16 | 871 | 2,120,937,472 |
| Sparse 32 | 774 | 2,049,003,520 |
| VDB 32 | 863 | 2,113,798,144 |



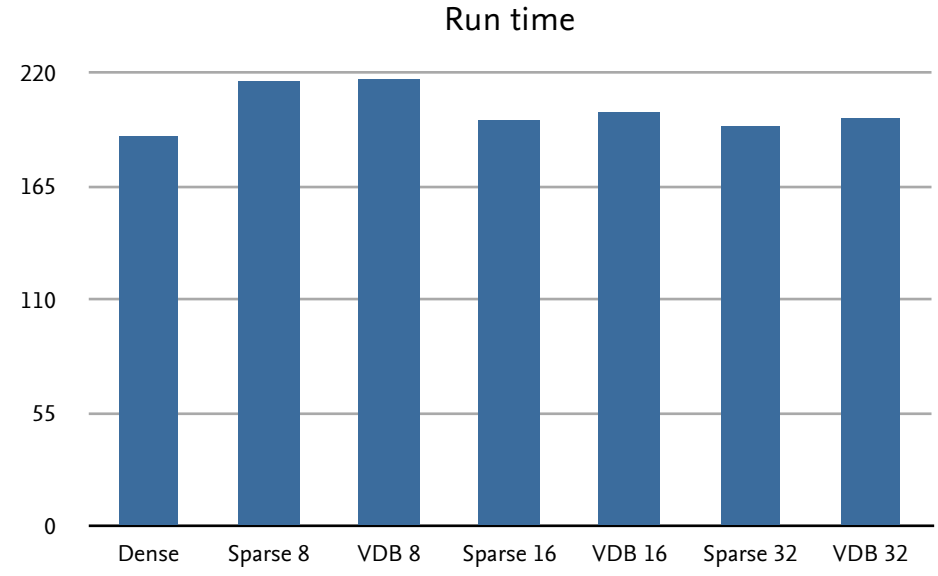

**About the test:**

This test measures the time it takes to write the value from each voxel in a freshly allocated buffer and includes both allocation and traversal time. It traverses the voxels in the order that is most efficient for each data structure. It is representative of the max speed that one could write the value of each voxel if a buffer needs to be allocated.



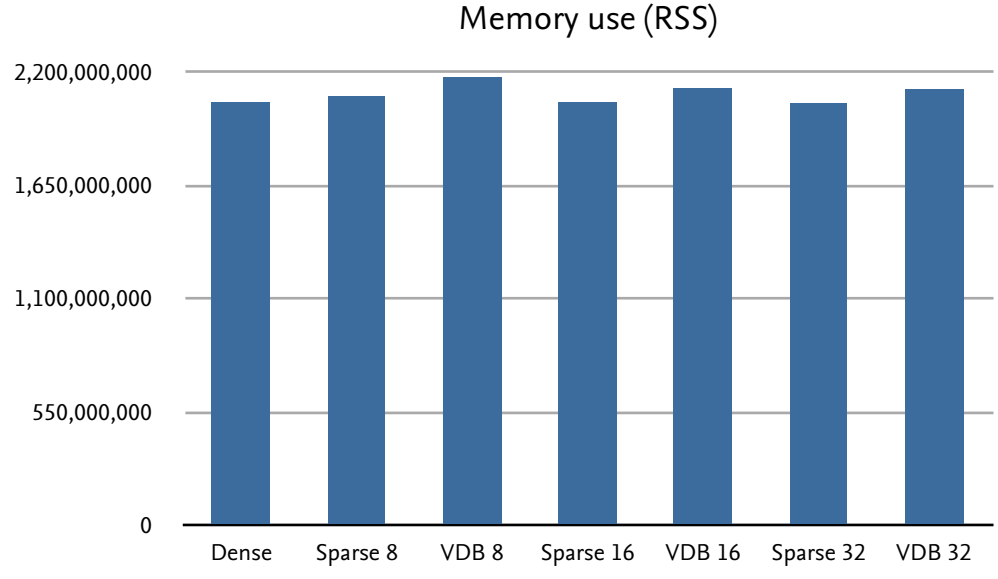
## Memory coherent write access $800^3$ (Preallocated)

| Data structure | (ms) | (bytes) |
|---|---|---|
| Dense | 189 | 2,049,142,784 |
| Sparse 8 | 216 | 2,080,505,856 |
| VDB 8 | 217 | 2,172,387,328 |
| Sparse 16 | 197 | 2,052,513,792 |
| VDB 16 | 201 | 2,120,957,952 |
| Sparse 32 | 194 | 2,049,024,000 |
| VDB 32 | 198 | 2,113,814,528 |



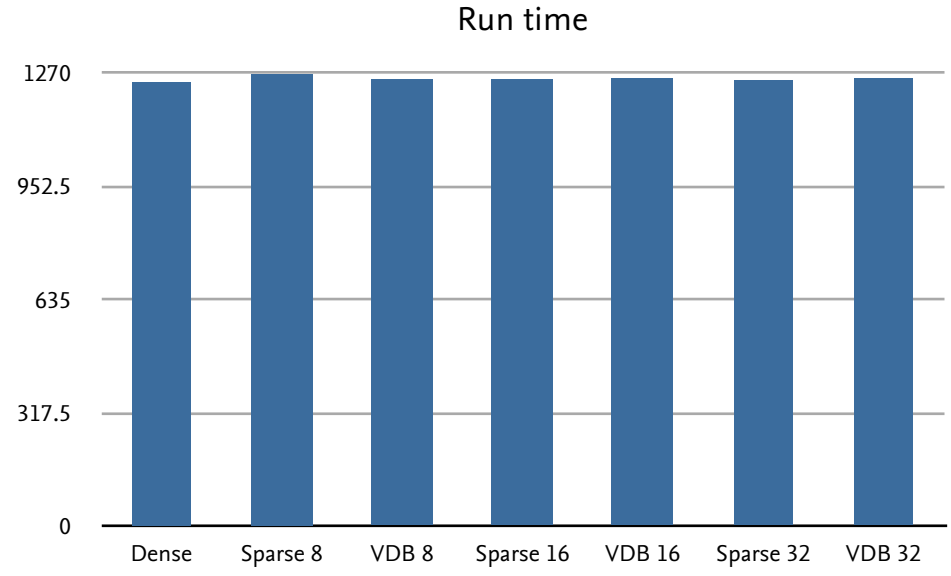

**About the test:**

This test measures the time it takes to write a value to each voxel in an existing buffer. It traverses the voxels in the order that is most efficient for each data structure. It is representative of the max speed that one could write the value of each voxel if a buffer already exists in memory.



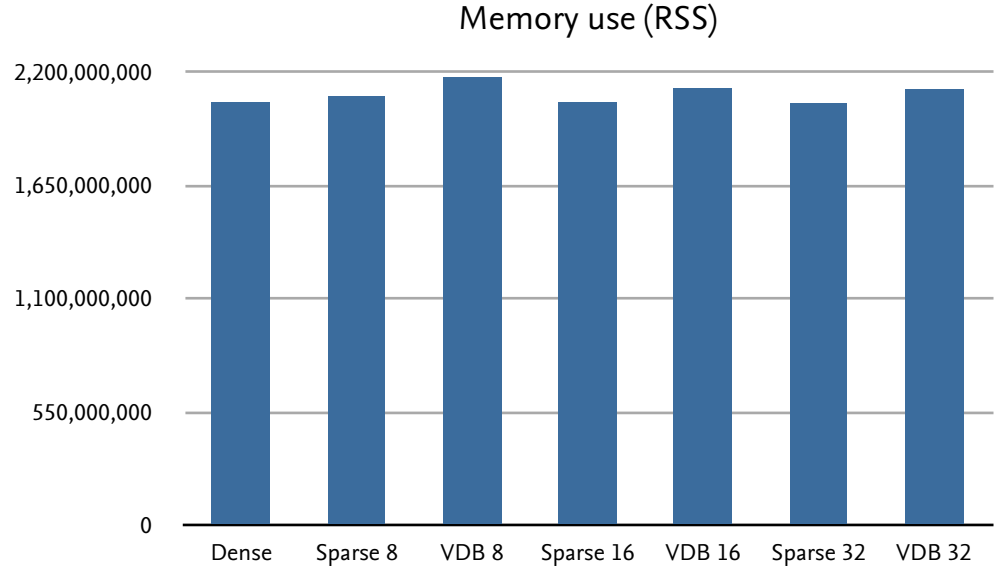
# Memory coherent read access $800^3$

| Data structure | (ms) | (bytes) |
|---|---|---|
| Dense | 1245 | 2,049,159,168 |
| Sparse 8 | 1267 | 2,080,522,240 |
| VDB 8 | 1252 | 2,172,395,520 |
| Sparse 16 | 1251 | 2,052,521,984 |
| VDB 16 | 1254 | 2,120,957,952 |
| Sparse 32 | 1248 | 2,049,024,000 |
| VDB 32 | 1255 | 2,113,818,624 |



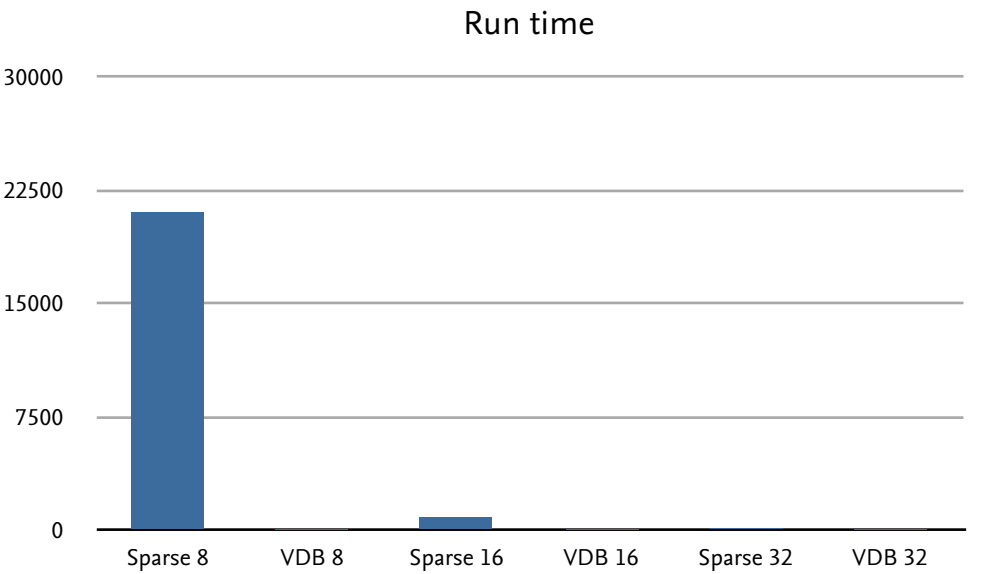

**About the test:**

This test measures the time it takes to read the value from each voxel. It traverses the voxels in the order that is most efficient for each data structure. It is representative of the max speed that one could read the value of each voxel.



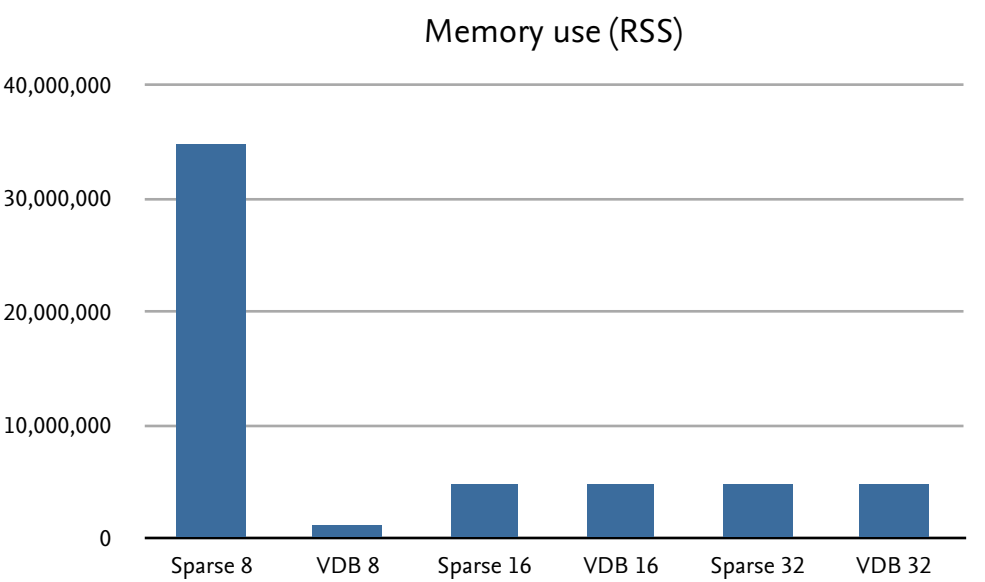
# Sparse fill $1000^3$

| Data structure | (µs) | (bytes) |
|---|---|---|
| Sparse 8 | 21056 | 34,721,792 |
| VDB 8 | 37 | 1,163,264 |
| Sparse 16 | 831 | 4,726,784 |
| VDB 16 | 33 | 4,726,784 |
| Sparse 32 | 102 | 4,726,784 |
| VDB 32 | 33 | 4,726,784 |



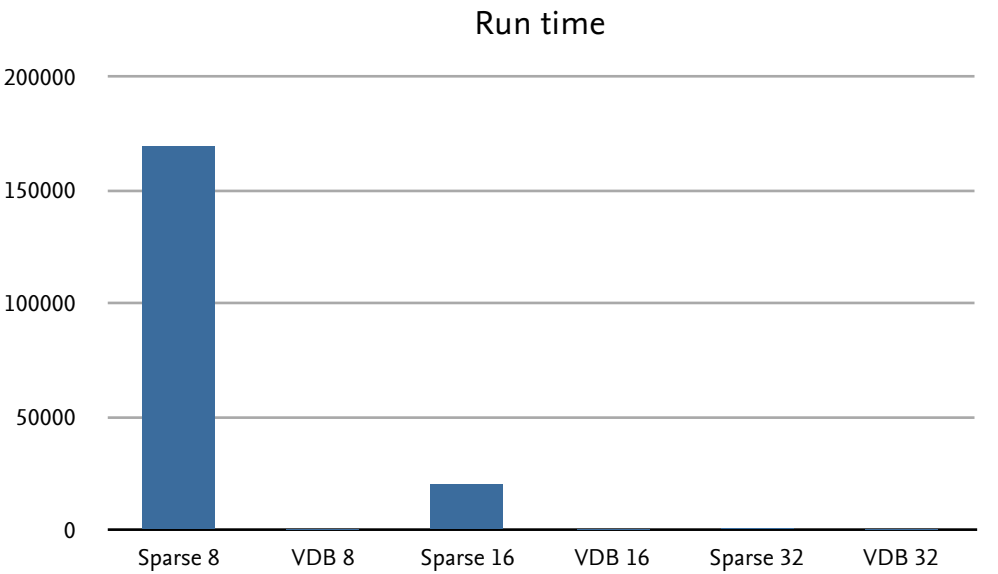

**About the test:**

This test measures the time it takes to fill an entire voxel buffer with “background values”. It is representative of setting the coarse “far” values of a narrow band levelset.



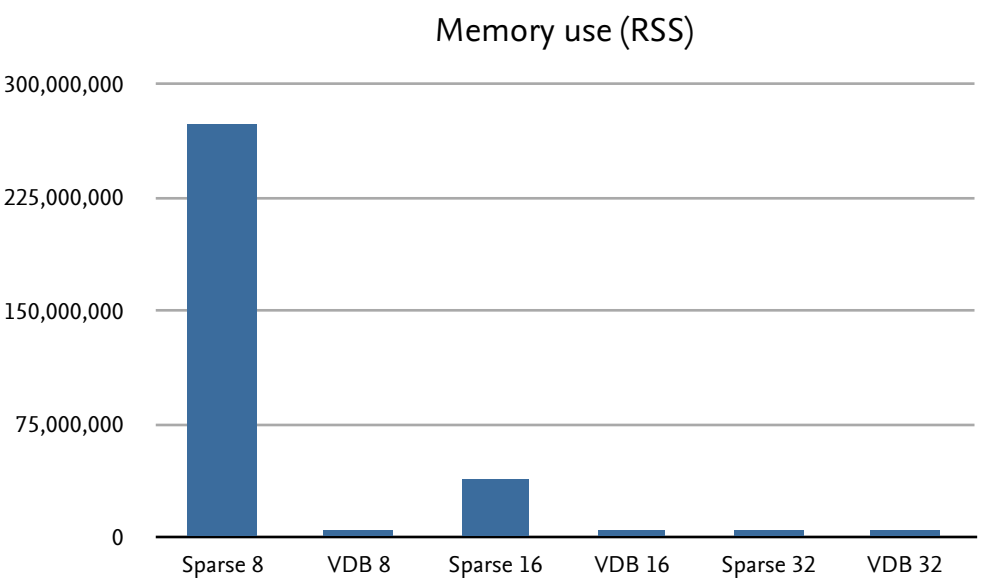
# Sparse fill $2000^3$

| Data structure | (µs) | (bytes) |
|---|---|---|
| Sparse 8 | 169206 | 273,166,336 |
| VDB 8 | 61 | 4,726,784 |
| Sparse 16 | 20121 | 38,285,312 |
| VDB 16 | 36 | 4,726,784 |
| Sparse 32 | 831 | 4,726,784 |
| VDB 32 | 33 | 4,726,784 |



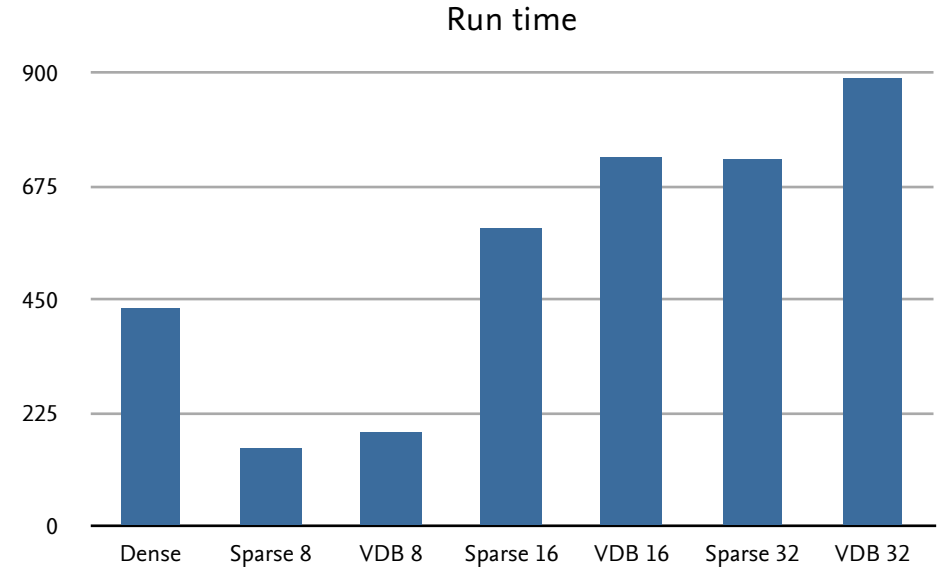

**About the test:**

This test measures the time it takes to fill an entire voxel buffer with “background values”. It is representative of setting the coarse “far” values of a narrow band levelset.



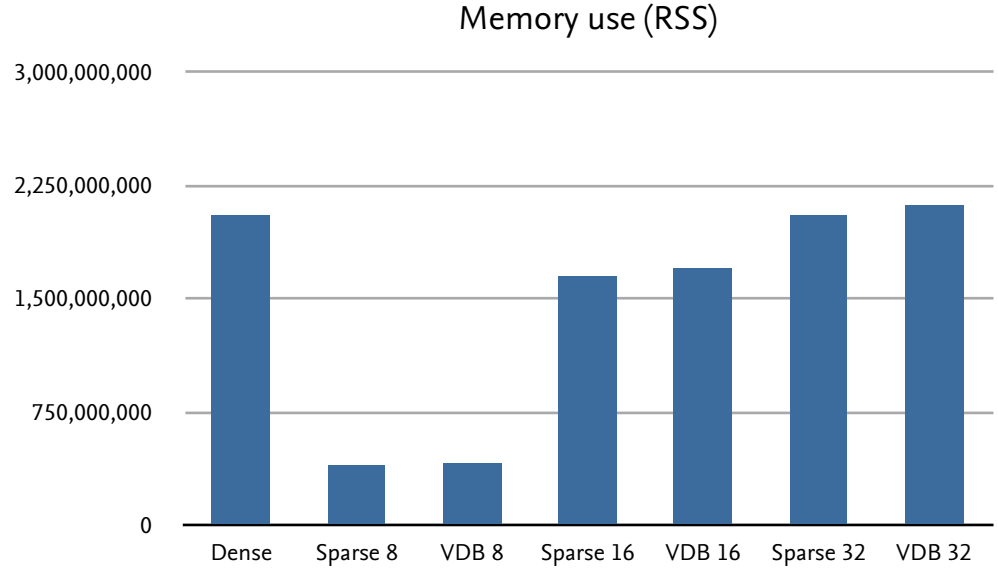
## Random incoherent write access: 200,000 points into $800^3$

| Data structure | (ms) | (bytes) |
|---|---|---|
| Dense | 432 | 2,052,759,552 |
| Sparse 8 | 155 | 393,469,952 |
| VDB 8 | 186 | 406,765,568 |
| Sparse 16 | 591 | 1,644,130,304 |
| VDB 16 | 732 | 1,698,942,976 |
| Sparse 32 | 729 | 2,051,473,408 |
| VDB 32 | 889 | 2,116,263,936 |



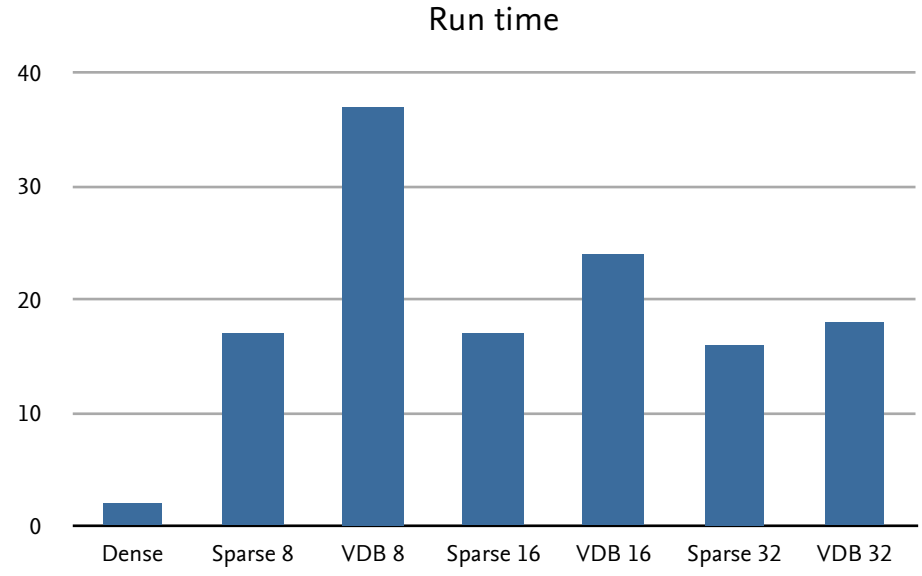

**About the test:**

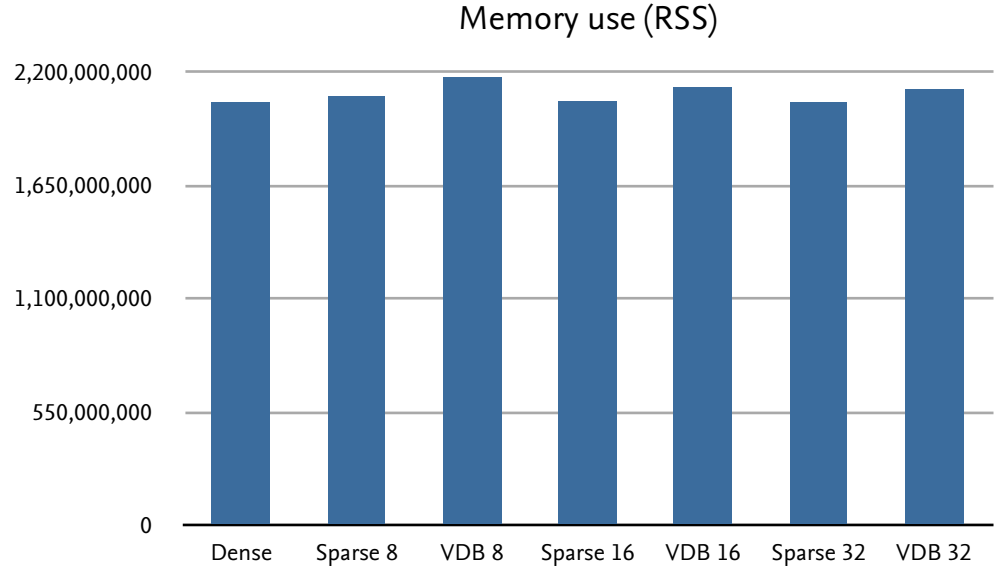
This test measures the time it takes to write 200,000 voxel values without interpolation from randomly selected positions. There is no coherency to the sampling. It is representative of splatting a point cloud to newly created buffer using a box filter.



## Random incoherent write access: 200,000 points into $800^3$ (Preallocated)

| Data structure | (ms) | (bytes) |
|---|---|---|
| Dense | 2 | 2,053,402,624 |
| Sparse 8 | 17 | 2,082,971,648 |
| VDB 8 | 37 | 2,174,844,928 |
| Sparse 16 | 17 | 2,054,971,392 |
| VDB 16 | 24 | 2,123,407,360 |
| Sparse 32 | 16 | 2,051,473,408 |
| VDB 32 | 18 | 2,116,263,936 |



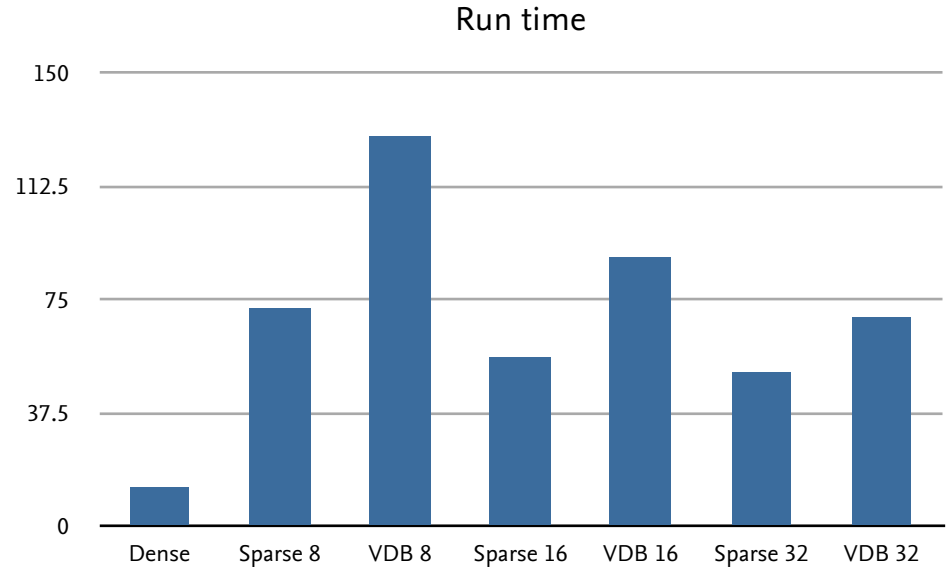




**About the test:**

This test measures the time it takes to write 200,000 voxel values without interpolation from randomly selected positions. There is no coherency to the sampling. It is representative of splatting a point cloud to an existing buffer using a box filter.

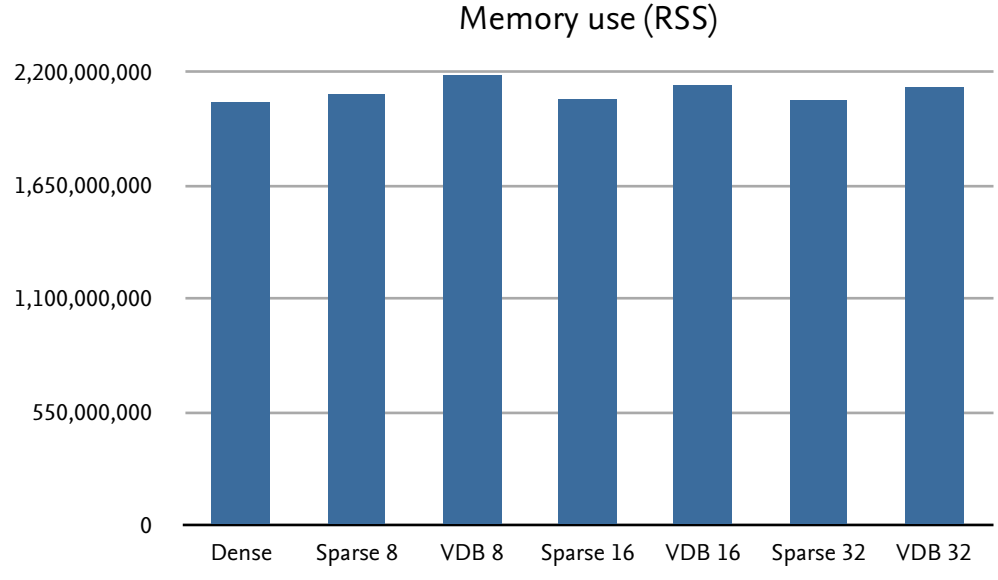
## Random incoherent read access: 200,000 points from $800^3$

| Data structure | (ms) | (bytes) |
|---|---|---|
| Dense | 13 | 2,052,907,008 |
| Sparse 8 | 72 | 2,092,580,864 |
| VDB 8 | 129 | 2,184,454,144 |
| Sparse 16 | 56 | 2,064,580,608 |
| VDB 16 | 89 | 2,133,016,576 |
| Sparse 32 | 51 | 2,061,082,624 |
| VDB 32 | 69 | 2,125,873,152 |



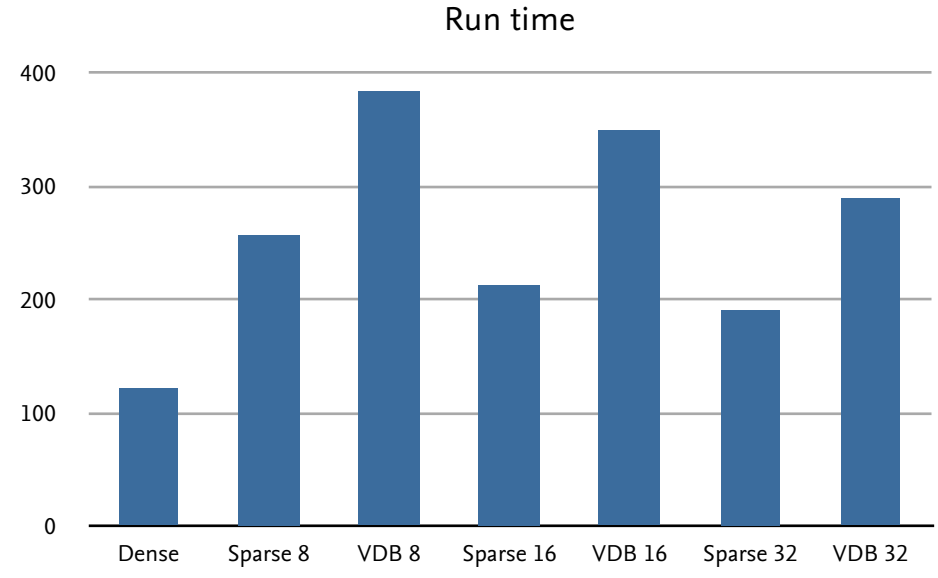

**About the test:**

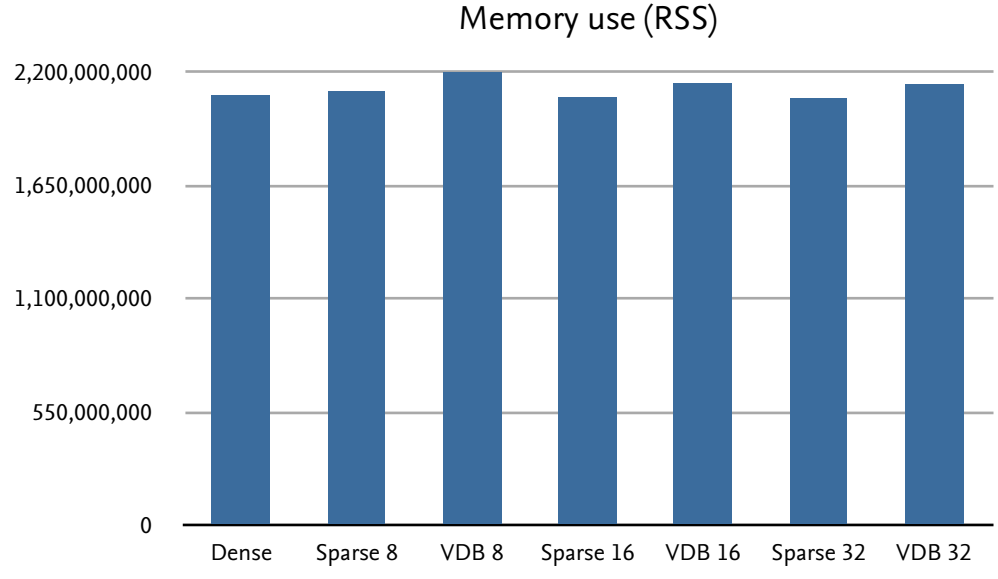
This test measures the time it takes to read 200,000 voxel values from randomly selected positions. There is no coherency to the sampling. It is representative of looking up positions from a point cloud in the voxel buffer, using nearest neighbor interpolation.



## Random incoherent interpolation: 200,000 points from $800^3$

| Data structure | (ms) | (bytes) |
|---|---|---|
| Dense | 122 | 2,085,240,832 |
| Sparse 8 | 257 | 2,104,606,720 |
| VDB 8 | 384 | 2,196,484,096 |
| Sparse 16 | 213 | 2,076,610,560 |
| VDB 16 | 349 | 2,145,050,624 |
| Sparse 32 | 191 | 2,073,112,576 |
| VDB 32 | 289 | 2,137,915,392 |



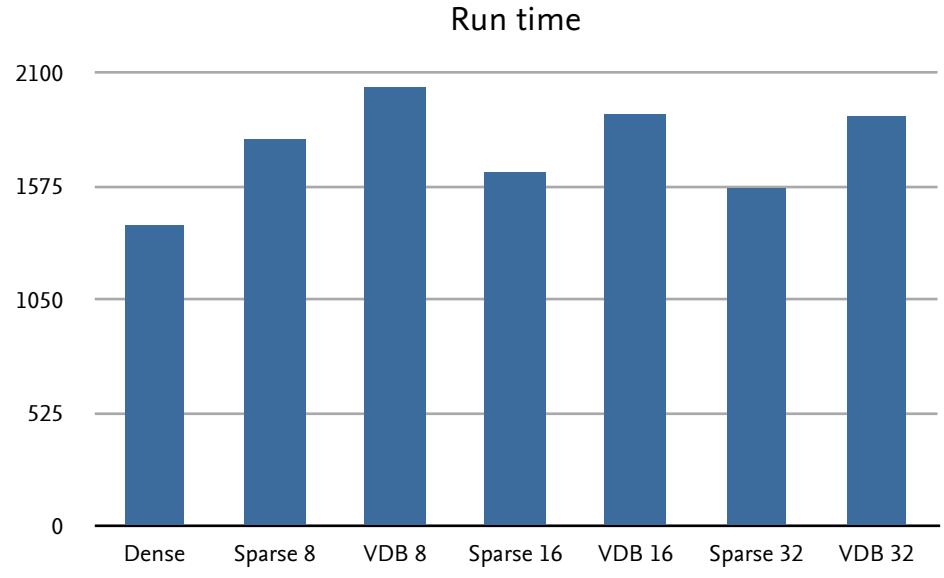




**About the test:**

This test measures the time it takes to interpolate 200,000 values from randomly selected positions. There is no coherency to the sampling. It is representative of looking up positions from a point cloud in the voxel buffer, using linear interpolation.

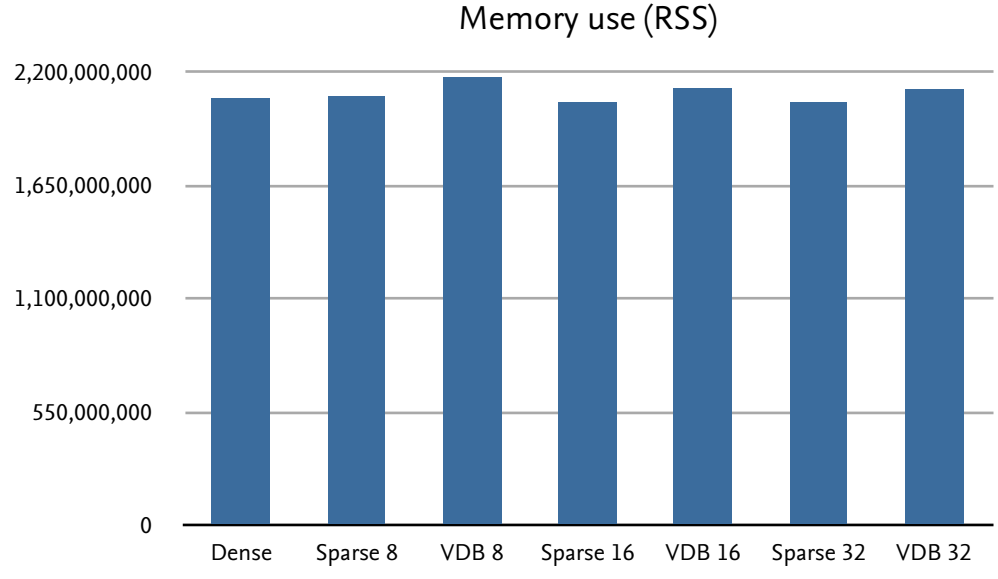
## Uniform raymarching $800^3$, 10,000 rays

| Data structure | (ms) | (bytes) |
|---|---|---|
| Dense | 1392 | 2,073,300,992 |
| Sparse 8 | 1792 | 2,080,661,504 |
| VDB 8 | 2032 | 2,172,538,880 |
| Sparse 16 | 1638 | 2,052,661,248 |
| VDB 16 | 1906 | 2,121,101,312 |
| Sparse 32 | 1566 | 2,049,163,264 |
| VDB 32 | 1899 | 2,113,961,984 |



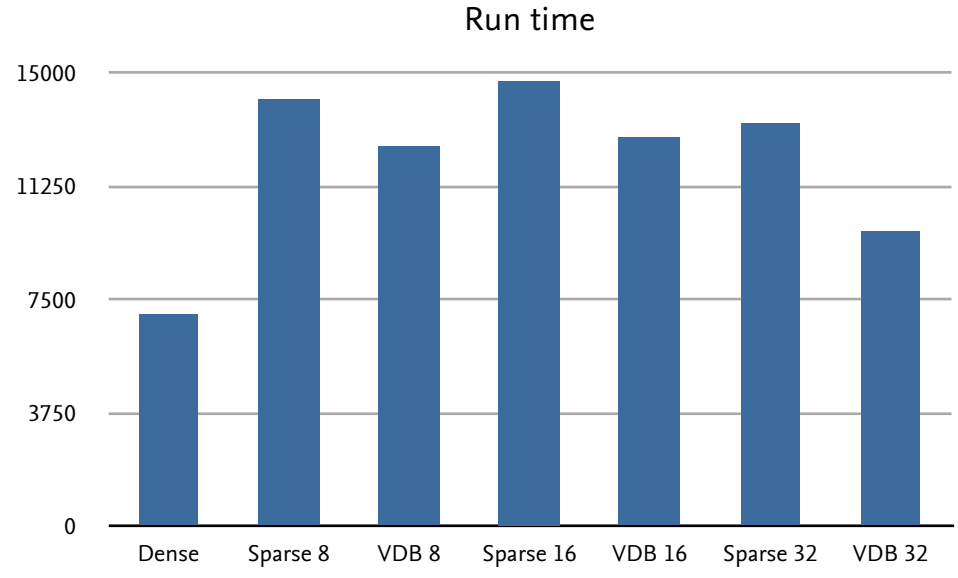

**About the test:**

This test measures the time it takes to interpolate values along a number of rays shot through the voxel buffer. It allows for each library to take advantage of potential cache coherency as the sampling pattern is somewhat ordered.



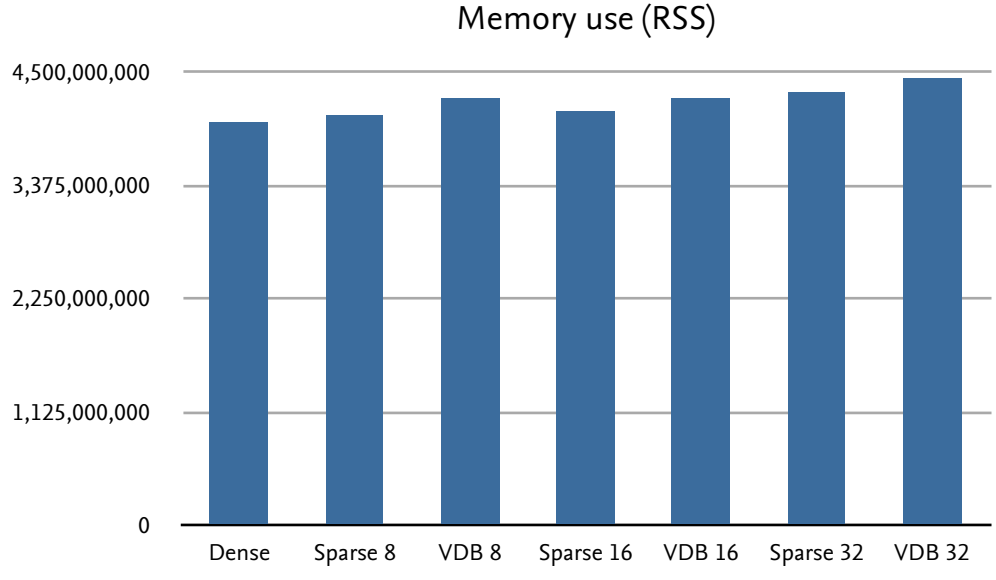
# Dense level set sphere $1000^3$

| Data structure | (ms) | (bytes) |
|---|---|---|
| Dense | 7007 | 4,001,300,480 |
| Sparse 8 | 14131 | 4,063,166,464 |
| VDB 8 | 12580 | 4,237,012,992 |
| Sparse 16 | 14735 | 4,105,441,280 |
| VDB 16 | 12854 | 4,239,904,768 |
| Sparse 32 | 13322 | 4,296,687,616 |
| VDB 32 | 9759 | 4,431,974,400 |



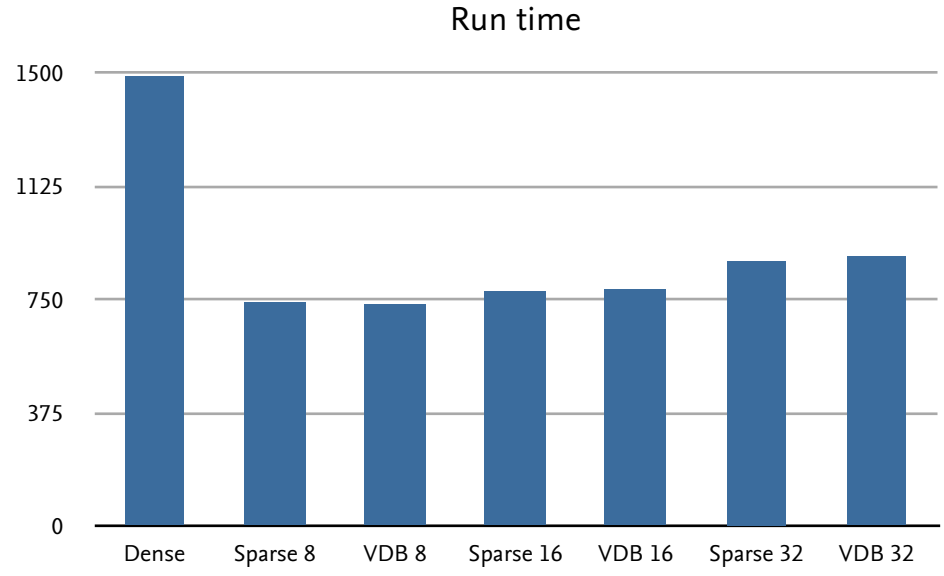

**About the test:**

This test measures the time it takes and the memory used when creating a full (i.e. every voxel populated) level set of a sphere.



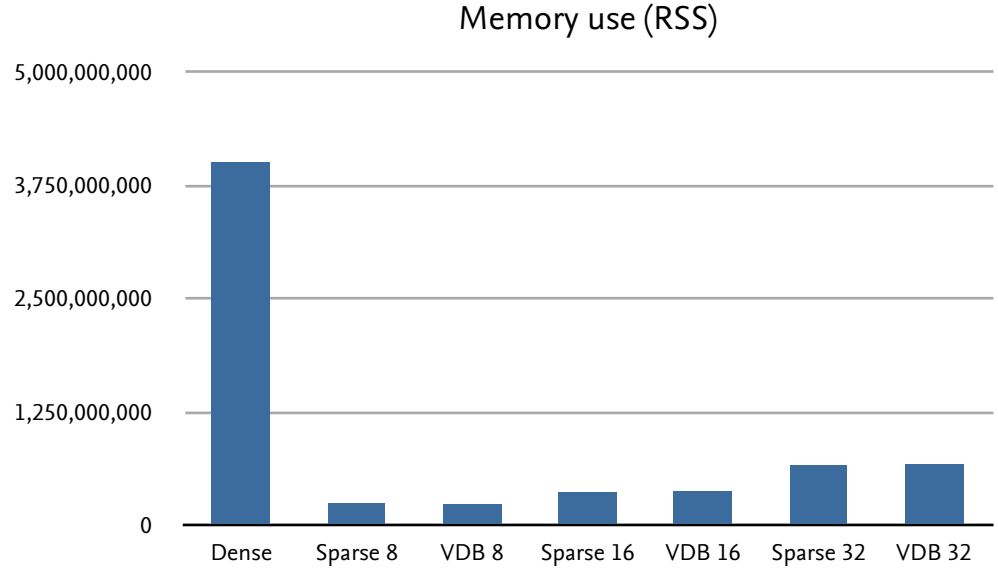
# Narrow band levelset $1000^3$

| Data structure | (ms) | (bytes) |
|---|---|---|
| Dense | 1488 | 4,001,075,200 |
| Sparse 8 | 740 | 237,608,960 |
| VDB 8 | 734 | 225,603,584 |
| Sparse 16 | 778 | 358,555,648 |
| VDB 16 | 785 | 368,463,872 |
| Sparse 32 | 878 | 654,282,752 |
| VDB 32 | 893 | 674,873,344 |



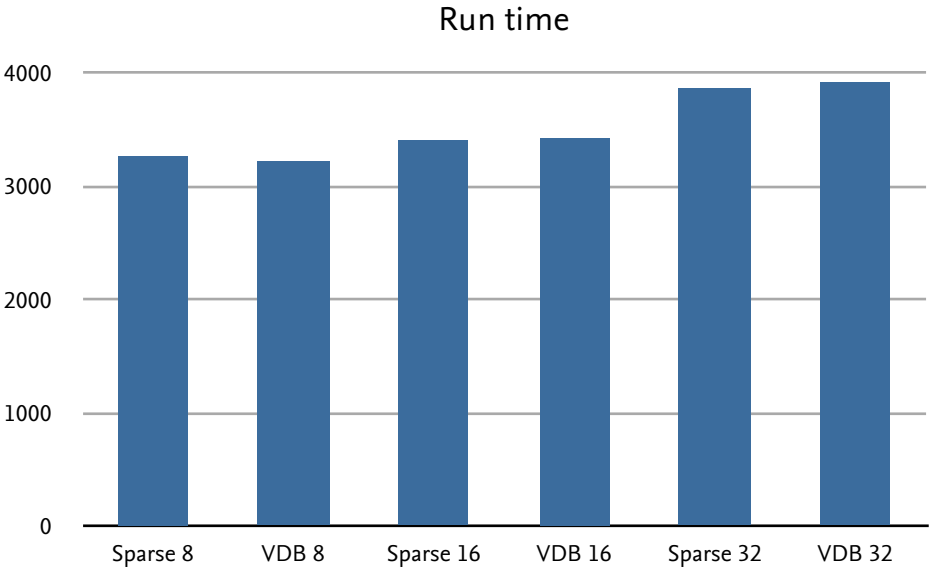

**About the test:**

This test measures the time it takes and the memory used when creating a simple narrow band levelset of a sphere.



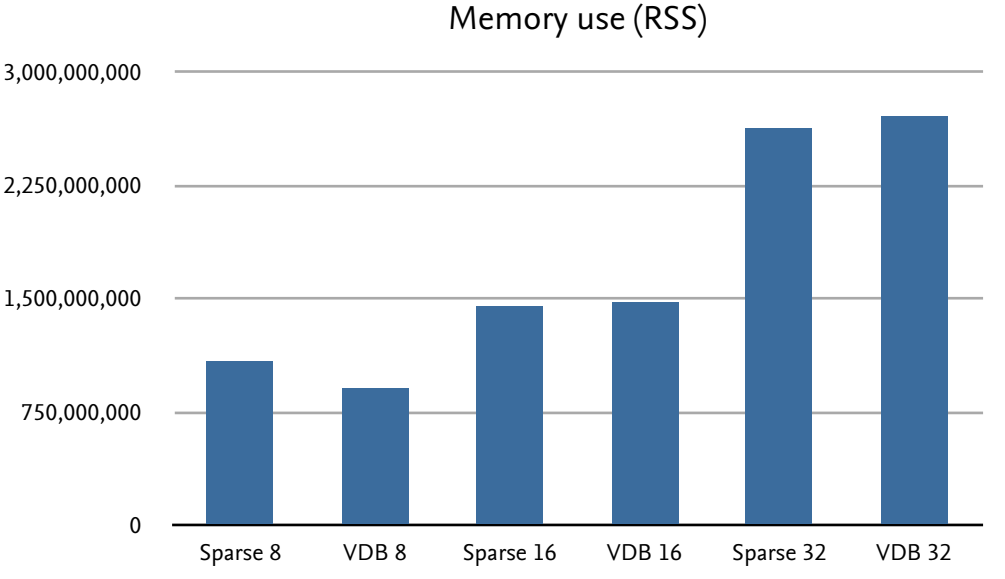
# Narrow band levelset $2000^3$

| Data structure | (ms) | (bytes) |
|---|---|---|
| Sparse 8 | 3267 | 1,080,627,200 |
| VDB 8 | 3219 | 908,431,360 |
| Sparse 16 | 3403 | 1,450,844,160 |
| VDB 16 | 3426 | 1,476,337,664 |
| Sparse 32 | 3865 | 2,625,261,568 |
| VDB 32 | 3916 | 2,705,952,768 |



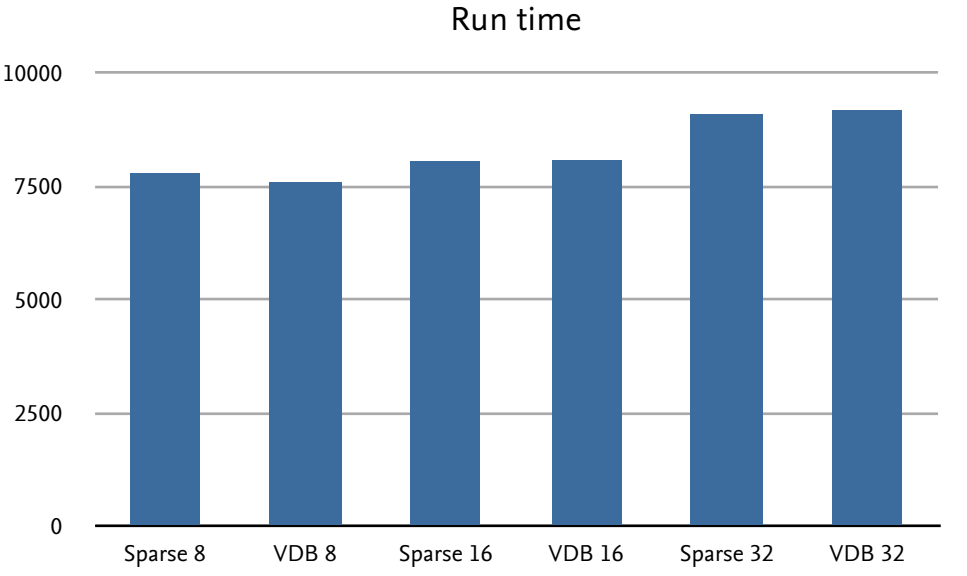

**About the test:**

This test measures the time it takes and the memory used when creating a simple narrow band levelset of a sphere.



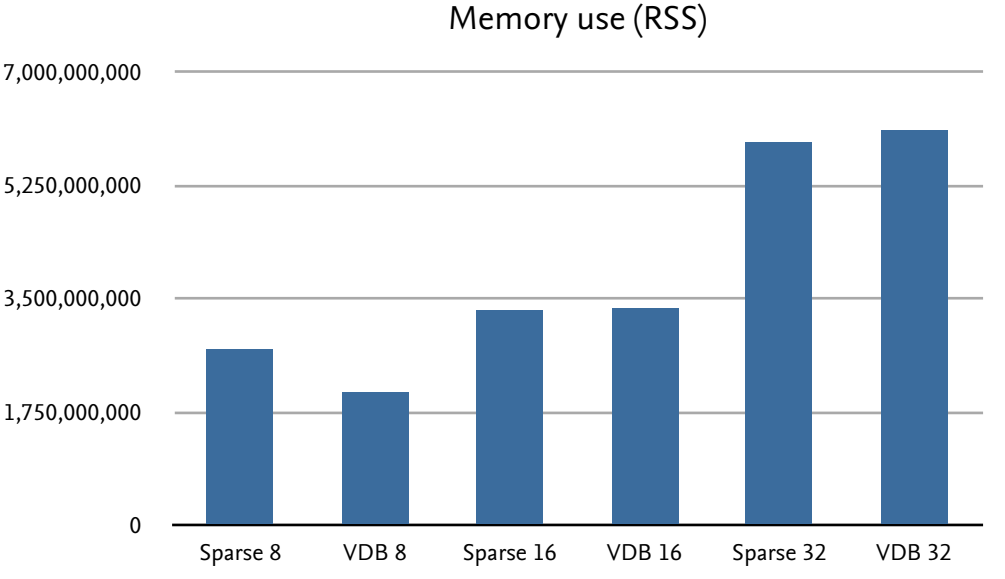
# Narrow band levelset $3000^3$

| Data structure | (ms) | (bytes) |
|---|---|---|
| Sparse 8 | 7776 | 2,714,550,272 |
| VDB 8 | 7591 | 2,046,099,456 |
| Sparse 16 | 8042 | 3,317,821,440 |
| VDB 16 | 8067 | 3,339,866,112 |
| Sparse 32 | 9088 | 5,914,374,144 |
| VDB 32 | 9174 | 6,092,304,384 |



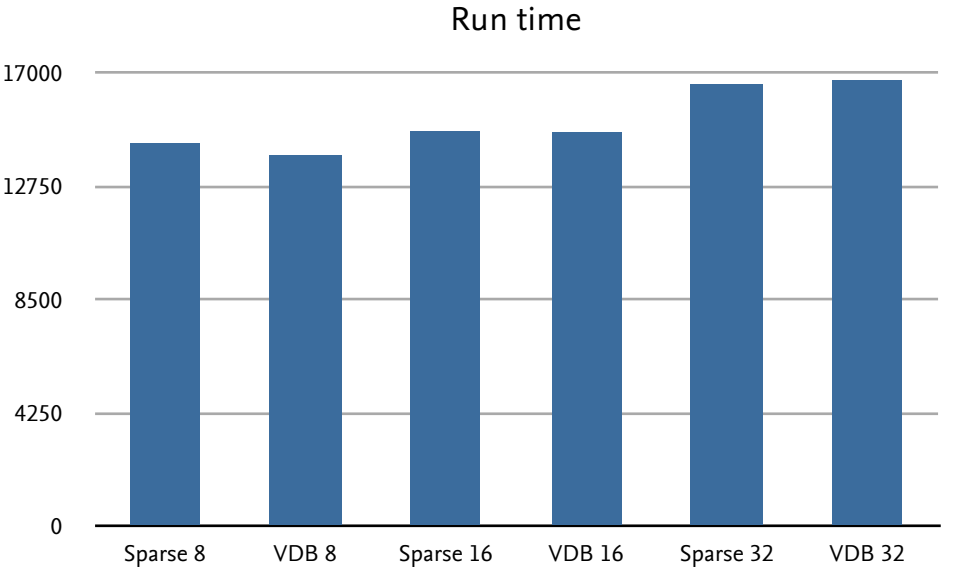

**About the test:**

This test measures the time it takes and the memory used when creating a simple narrow band levelset of a sphere.



# Narrow band levelset $4000^3$

| Data structure | (ms) | (bytes) |
|---|---|---|
| Sparse 8 | 14355 | 5,330,575,360 |
| VDB 8 | 13894 | 3,643,482,112 |
| Sparse 16 | 14812 | 5,953,753,088 |
| VDB 16 | 14775 | 5,931,925,504 |
| Sparse 32 | 16573 | 10,479,644,672 |
| VDB 32 | 16726 | 10,786,873,344 |




**About the test:**

This test measures the time it takes and the memory used when creating a simple narrow band levelset of a sphere.