Subject: Re: BufferPainter::Clear() optimization
Posted by Tom1 on Mon, 18 May 2020 19:40:50 GMT
View Forum Message <> Reply to Message

Hi Mirek,

Something like this, maybe... I'm not quite sure as this method reports 16M cache for me -- although this works quite well for me:

```
static int cachesize=999;
INITBLOCK{
#ifdef COMPILER_MSC
 int cpuInfo[4];
 Zero(cpuInfo);
 __cpuid(cpuInfo, 0x80000006);
#else
 unsigned int cpuInfo[4];
 Zero(cpuInfo);
 __get_cpuid(0x80000006, &cpuInfo[0], &cpuInfo[1], &cpuInfo[2], &cpuInfo[3]);
#endif
 cachesize=1024*(cpuInfo[2]>>16)*(cpuInfo[2]&0xff);
};


void inline Fill3T(void *b, dword data, int len){
 switch(len){
  case 3: ((dword *)b)[2] = data;
  case 2: ((dword *)b)[1] = data;
  case 1: ((dword *)b)[0] = data;
  case 0: return;
 }
 __m128i q = _mm_set1_epi32(*(int*)&data);
 __m128i *w = (__m128i*)b;

 if(len >= 32) {
  __m128i *e = (__m128i*)b + (len>>2) - 8;
  if(len >= (cachesize>>2) && ((uintptr_t)w & 3) == 0) { // for really huge data, bypass the cache
   _mm_storeu_si128(w, q); // Head align
   int s=(-((int)((uintptr_t)b)>>2))&0x3;
   w = (__m128i*) ((dword*)b) + s;
   do {
    _mm_stream_si128(w++, q);
    _mm_stream_si128(w++, q);
    _mm_stream_si128(w++, q);
    _mm_stream_si128(w++, q);
    _mm_stream_si128(w++, q);
    _mm_stream_si128(w++, q);
    _mm_stream_si128(w++, q);
```

```
    _mm_stream_si128(w++, q);
   }while(w<=e);
   _mm_sfence();
  }
  else
   do {
    _mm_storeu_si128(w++, q);
    _mm_storeu_si128(w++, q);
    _mm_storeu_si128(w++, q);
    _mm_storeu_si128(w++, q);
    _mm_storeu_si128(w++, q);
    _mm_storeu_si128(w++, q);
    _mm_storeu_si128(w++, q);
    _mm_storeu_si128(w++, q);
   }while(w<=e);
 }

 if(len & 16) {
  _mm_storeu_si128(w++, q);
  _mm_storeu_si128(w++, q);
  _mm_storeu_si128(w++, q);
  _mm_storeu_si128(w++, q);
 }
 if(len & 8) {
  _mm_storeu_si128(w++, q);
  _mm_storeu_si128(w++, q);
 }
 if(len & 4) {
  _mm_storeu_si128(w, q);
 }
 _mm_storeu_si128((__m128i*) (((dword*)b) + len - 4), q); // Tail align
}
```

Best regards,

Tom