Subject: Re: BufferPainter::Clear() optimization
Posted by Tom1 on Sat, 16 May 2020 22:10:57 GMT
View Forum Message <> Reply to Message

Hi,

Interestingly, the FillRGBA() that can be found in the current BufferPainter Fillers, is a real performer. It wins below 1M dwords just about anything else. However, Mireks new MemSet is the winner thereafter. This applies on Windows 10 x64 to CLANG/CLANGx64/MSBT19 on my Core i7. Only MSBT19x64 has a different situation and the following code tries to optimize that, in addition to combining FillRGBA and MemSet for the other compilers:

```
#if defined(WIN64) && defined(COMPILER_MSC)

// for MSBT19x64 only:
inline void new_memsetd(void *b, dword data, int len){
 dword *t=(dword *)b;
 switch(len){
  case 6: t[5] = data;
  case 5: t[4] = data;
  case 4: t[3] = data;
  case 3: t[2] = data;
  case 2: t[1] = data;
  case 1: t[0] = data;
  case 0: return;

  default:{
   if(len&1) *t++=data;
   len>>=1;

   uint64 *w=(uint64*)t;
   uint64 q=*(dword*)&data;
   q |= (q << 32);

   switch(len) {
    default:{
     uint64 *lim = w + len - 32;
     while(w < lim) *w++ = q;
    }
    case 32: w[31] = q;
    case 31: w[30] = q;
    case 30: w[29] = q;
    case 29: w[28] = q;
    case 28: w[27] = q;
    case 27: w[26] = q;
    case 26: w[25] = q;
    case 25: w[24] = q;
    case 24: w[23] = q;
    case 23: w[22] = q;
```

```
    case 22: w[21] = q;
    case 21: w[20] = q;
    case 20: w[19] = q;
    case 19: w[18] = q;
    case 18: w[17] = q;
    case 17: w[16] = q;
    case 16: w[15] = q;
    case 15: w[14] = q;
    case 14: w[13] = q;
    case 13: w[12] = q;
    case 12: w[11] = q;
    case 11: w[10] = q;
    case 10: w[9] = q;
    case 9: w[8] = q;
    case 8: w[7] = q;
    case 7: w[6] = q;
    case 6: w[5] = q;
    case 5: w[4] = q;
    case 4: w[3] = q;
    case 3: w[2] = q;
    case 2: w[1] = q;
    case 1: w[0] = q;
   }
  }
 }
}

#else

inline void new_memsetd(void *b, dword data, int len){
 if(len<=1024*1024) FillRGBA((RGBA*)b,*(RGBA*)&data,len);
 else MemSet(b,data,len);
}

#endif
```

The benchmarking code for various fill sizes now looks like this:

```
 RGBA c = Red();

 int bsize=8*1024*1024;
 Buffer<RGBA> b(bsize,(RGBA)Blue());

 String result="\"N\",\"Fill()\",\"new_memsetd()\",\"MemSet()\",\"FillRGBA()\"\r\n";
 for(int len=1;len<=bsize;){
  int maximum=100000000/len;
  int64 t0=usecs();
  for(int i = 0; i < maximum; i++) Fill(~b, c, len);
  int64 t1=usecs();
```

```
  for(int i = 0; i < maximum; i++) new_memsetd(~b, *(dword*)&c, len);
  int64 t2=usecs();
  for(int i = 0; i < maximum; i++) MemSet(~b, c, len);
  int64 t3=usecs();
  for(int i = 0; i < maximum; i++) FillRGBA(~b, c, len);
  int64 t4=usecs();
  result.Cat(Format("%d,%f,%f,%f,%f\r\n",len,1000.0*(t1-t0)/maximum,1000.0*(t2-t1)/maximum,10
00.0*(t3-t2)/maximum,1000.0*(t4-t3)/maximum));
  if(len<64) len++;
  else len*=2;
 }

 SaveFile(GetHomeDirFile("Desktop/memset.csv"),result);
```

Best regards,

Tom
