Subject: Re: String::Cat optimization
Posted by mirek on Thu, 01 Dec 2011 16:52:45 GMT
View Forum Message <> Reply to Message

Tom1 wrote on Thu, 01 December 2011 08:50Hi,

I played around with Mirek's idea awhile and according to my simple '::GetTickCount()' benchmarking on MSC9/Win7x64 I managed to squeeze yet more performance out of it. The test covered all transfer lengths from 1 to 16 bytes.

The svo_memcpy() suffers a performance penalty at len==16, where secondary function call to memcpy steps in. The following macro approach helps dramatically to reduce that penalty. I also discovered that the memcpy() performance might not be reached systematically at transfer lengths above 11 bytes, so limiting the switch to <= 11 bytes should improve overall performance.

```
 inline void memcpy11i(char *t, const char *s, int len){
  switch(len) {
   case 11: t[10] = s[10];
   case 10: t[9] = s[9];
   case  9: t[8] = s[8];
   case  8: t[7] = s[7];
   case  7: t[6] = s[6];
   case  6: t[5] = s[5];
   case  5: t[4] = s[4];
   case  4: t[3] = s[3];
   case  3: t[2] = s[2];
   case  2: t[1] = s[1];
   case  1: t[0] = s[0];
  }
 }

#define memcpy11(t, s, len) (len)>11 ? memcpy(t, s, len) : memcpy11i(t, s, len)
```

How does this perform on your systems?

Best regards,

Tom

Well, that is actually even better, as MSC refuses to inline that function....

Mirek