

---

Subject: Re: Writing Bits object to disk  
Posted by [mirek](#) on Tue, 25 Apr 2017 08:31:35 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

crydev wrote on Tue, 25 April 2017 09:36A ran the following code in a performance test 100,000 times.

// Original function implementing Vector<bool>.

```
const int VectorBoolOrBitsetTestOriginal(bool* const buffer, const bool* const rand)
{
    int x = 0;
    for (int i = 0; i < 4096; ++i)
    {
        if (rand)
        {
            ++x;
        }
        buffer[i] = rand;
    }
    return x;
}
```

// Different function implementing Bits.

```
const int VectorBoolOrBitsetTestBitSet(Bits& buffer, const bool* const rand)
{
    int x = 0;
    for (int i = 0; i < 4096; ++i)
    {
        if (rand)
        {
            ++x;
        }
        buffer.Set(i, rand);
    }
    return x;
}
```

// Different function implementing std::bitset.

```
const int VectorBoolOrBitsetTestStdBitSet(std::bitset<4096>& buffer, const bool* const rand)
{
    int x = 0;
    for (int i = 0; i < 4096; ++i)
    {
        if (rand)
        {
            ++x;
        }
    }
}
```

```
    buffer.set(i, rand);  
}  
return x;  
}
```

The result is as follows, Bits being approximately a factor 10 slower. `std::bitset` already seems to be a twice as fast:

crydev

The first quick observation of Bits code (after all these years) reveals that there are pretty good opportunities to optimize this. Woohoo, optimization time!

BTW, could you please post me your benchamrk package zipped? Would save me a bit of time.

Also, I am still undecided about correct interface to expose raw data. Currently I am thinking along something like:

```
const byte *ReadRaw(int& count_of_bytes);  
byte *WriteRaw(int count_of_bytes);
```

WriteRaw would make sure that there is at least count bytes available for bits (without reallocating array down).

Mirek

---