

## Re: map.insert(key,val) vs. map[key]=val ?

**Source:** [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.cpp/2004-10/2695.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.cpp/2004-10/2695.html)

---

**From:** Victor Bazarov (v.Abazarov\_at\_comAcast.net)

**Date:** 10/19/04

Date: Tue, 19 Oct 2004 13:14:10 -0400

Patrick Guio wrote:

> *On Tue, 19 Oct 2004, Victor Bazarov wrote:*

>

>>> *I have benchmarked 2 different ways to add an element in a map  
>>> container.*

>>>

>>> *1 map.insert(key,val)*

>>> *2 map[key]=val*

>>>

>>> *I used gcc and it comes out that the second way is about twice as  
>>> fast as the first one.*

>>> *Is there a reason/case why I should use the insert() method rather  
>>> the operator[]?*

>>> *Are the 2 different completely equivalent?*

>>

>>

>> *No, they are not. According to the Standard, the second one creates  
>> a default 'value' associated with 'key' in the container, and then uses  
>> the assignment op to get the contents of 'val' into the newly created  
>> value. The operator[] is said to be implemented in terms of 'insert'.  
>> So, all things being standard, I'd expect 'insert' to work faster in  
>> general.*

>

>

> *Strange, here is my very simple (maybe too simple?) benchmark*

>

> *#include <map>*

> *#include <string>*

> *#include <iostream>*

> *#include <ctime>*

I had to add

```
#define pair mypair
```

here

## comp.lang.c++. Re: map.insert(key,val) vs. map[key]=val ?

```
>
> typedef std::map<int, std::string> lut;
> typedef lut::value_type pair;
>
> #define BENCH(BLOCK,NAME,N) \
> { \
>     lut a; \
>     clock_t tic=clock(); \
>     for (int i=0; i<N; i++) { \
>         BLOCK \
>     } \
>     double elapsed = double(clock()-tic)/N/CLOCKS_PER_SEC; \
>     std::cout << "Time elapsed = " << 1e9*elapsed << " ns" << std::endl; \
> } \
>
> int const N=100000;
>
> int main()
> {
>     BENCH(a.insert(pair(i%1000,"foo1")), "map::insert()",N)
>     BENCH(a[i%1000]="foo1";, "map::operator[]",N)
>
>     BENCH(lut a; a.insert(pair(i%1000,"foo1")), "map::insert()",N)
>     BENCH(lut a; a[i%1000]="foo1";, "map::operator[]",N)
> }
>
> And the output on a linux box running g++ 3.2.3 compiling without any
> optimisation
>
> Time elapsed = 900 ns
> Time elapsed = 500 ns
> Time elapsed = 1100 ns
> Time elapsed = 1200 ns
>
> How do you explain this?
```

I am not going to even try. In my run (gcc v.2.96) I get

```
Time elapsed = 1100 ns
Time elapsed = 1000 ns
Time elapsed = 1100 ns
Time elapsed = 1300 ns
```

Which is close enough to not to be concerned. BTW, what do you think the \*real\* resolution of 'clock' on your system is?

I went ahead and compiled and ran it on another system several times in a row (with less than a second between the runs, as fast as it took me to press the up arrow and 'enter'). Here is what I got:

comp.lang.c++. Re: map.insert(key,val) vs. map[key]=val ?

```
/home/vbazarov/temp% ./test
Time elapsed = 700 ns
Time elapsed = 600 ns
Time elapsed = 900 ns
Time elapsed = 900 ns
/home/vbazarov/temp% ./test
Time elapsed = 700 ns
Time elapsed = 700 ns
Time elapsed = 800 ns
Time elapsed = 1000 ns
/home/vbazarov/temp% ./test
Time elapsed = 600 ns
Time elapsed = 600 ns
Time elapsed = 800 ns
Time elapsed = 1200 ns
/home/vbazarov/temp% ./test
Time elapsed = 700 ns
Time elapsed = 600 ns
Time elapsed = 700 ns
Time elapsed = 1000 ns
/home/vbazarov/temp% ./test
Time elapsed = 700 ns
Time elapsed = 600 ns
Time elapsed = 900 ns
Time elapsed = 1100 ns
/home/vbazarov/temp% ./test
Time elapsed = 700 ns
Time elapsed = 600 ns
Time elapsed = 800 ns
Time elapsed = 1000 ns
```

After increasing the number of runs ten times, I get a bit more precise readings, but still not really significantly different:

```
Time elapsed = 760 ns
Time elapsed = 630 ns
Time elapsed = 790 ns
Time elapsed = 1030 ns
```

Is that something indicative of anything? I am not sure. Possibly the ability of a compiler to optimise certain things and inability to optimise other things.

BTW, all previous results presented here are G++-compiled. If I do the same test on VC++ 7.1, I get

```
Time elapsed = 262 ns
Time elapsed = 140 ns
Time elapsed = 1020 ns
Time elapsed = 959 ns
```

comp.lang.c++. Re: map.insert(key,val) vs. map[key]=val ?

I am not going to provide any explanation here either.

Victor