Subject: cpu times
Posted by Gianluigi Boca on Fri, 11 May 2012 16:36:41 GMT
View Forum Message <> Reply to Message

dear collaborators,
we all know at least from oral tradition, that the Standard Template Library containers are somewhat slower compared to the 'traditional' C code style arrays.

But how much slower are they actually ?

I checked the difference in cpu consumption when using a conventional C array or a Standard Template Library <vector> instead, using a very simple program.

I measured the cputime consumption
of 10,000,000,000 assignment operations [avoiding
though a calculation that can be optimized heavily by the compiler].

I wrote two almost identical simple loop programs :

1) Conventional C array program :

```
int main ()
{
 int v[10],b ;
 itmp = 500000;
for(int j=0;j<10000;j++){
 for(int i=0;i<itmp;i++){
 b=i+j;
 v[3]=j+i;
 b=v[3];
 }
}
```

2) Template <vector> program :

```
#include <vector>
int main ()
{
 vector <int> v(10,0) ;
 int b;
 int itmp = 500000;

for(int j=0;j<10000;j++){
 for(int i=0;i<itmp;i++){
 b=i+j;
 v.at(3)=i+j;
 b=v.at(3);
 }
}
  return 0;
};
```

I measured the cpu consumption of the two programs.
I also measured (and subtracted) the cputime consumption of the NON RELEVANT part of the code, namely :

```
int main ()
{
 int b;
 int itmp = 500000;

for(int j=0;j<10000;j++){
 for(int i=0;i<itmp;i++){
 b=i+j;
 }
}
  return 0;
};
```

THE FOLLOWING IS THE CpuTime CONSUMPTION OF THE TRADITIONAL C STYLE

```
 v[3]=j+i;
 b=v[3];
```

STATEMENTS : 9.352 sec

while THE FOLLOWING IS THE CpuTime CONSUMPTION OF THE Standard Template Library

```
 v.at(3)= i+j;
 b=v.at(3);
```

STATEMENTS : 166 sec

In other words, a factor almost 18 worse of the Template <vector>.

As you very well know, the Template <vector> gives the advantage
of the array boundary check (only when you use the at() function
though, NOT when you use the [] form!) but I am wondering if we can afford a factor of speed 18 slower in acces time for a code such as the PANDA code that is supposed to digest billion and billion of events in the future.

Please comment, thanks

Gianluigi