Ruby master - Bug #7500
Improve GC profiler timings on linux
12/03/2012 11:32 AM - tmm1 (Aman Gupta)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>authorNari (Narihiro Nakamura)</td>
</tr>
<tr>
<td>Target version:</td>
<td>2.0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On linux kernels, getrusage()'s precision depends on the value of HZ when the kernel was compiled. By default, HZ=250 provides a 4ms granularity. This patch uses clock_gettime() with CLOCK_PROCESS_CPUTIME_ID when available, which provides a 1ns precision on linux.</td>
</tr>
</tbody>
</table>

Associated revisions
Revision ab012bda - 12/05/2012 02:53 PM - nari
- gc.c (getrusage_time): uses clock_gettime() with CLOCK_PROCESS_CPUTIME_ID when available, which provides a 1ns precision on linux. [ruby-core:50495] [Bug #7500]
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38214 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 38214 - 12/05/2012 02:53 PM - nari
- gc.c (getrusage_time): uses clock_gettime() with CLOCK_PROCESS_CPUTIME_ID when available, which provides a 1ns precision on linux. [ruby-core:50495] [Bug #7500]

Revision 38214 - 12/05/2012 02:53 PM - nari
- gc.c (getrusage_time): uses clock_gettime() with CLOCK_PROCESS_CPUTIME_ID when available, which provides a 1ns precision on linux. [ruby-core:50495] [Bug #7500]

Revision 38214 - 12/05/2012 02:53 PM - nari
- gc.c (getrusage_time): uses clock_gettime() with CLOCK_PROCESS_CPUTIME_ID when available, which provides a 1ns precision on linux. [ruby-core:50495] [Bug #7500]

Revision 38214 - 12/05/2012 02:53 PM - nari
- gc.c (getrusage_time): uses clock_gettime() with CLOCK_PROCESS_CPUTIME_ID when available, which provides a 1ns precision on linux. [ruby-core:50495] [Bug #7500]

Revision 38214 - 12/05/2012 02:53 PM - nari
- gc.c (getrusage_time): uses clock_gettime() with CLOCK_PROCESS_CPUTIME_ID when available, which provides a 1ns precision on linux. [ruby-core:50495] [Bug #7500]

History
#1 - 12/03/2012 11:46 AM - authorNari (Narihiro Nakamura)
- Category set to core
- Assignee set to authorNari (Narihiro Nakamura)

08/07/2021
#2 - 12/03/2012 12:16 PM - kosaki (Motohiro KOSAKI)

AFAIK, clock_gettime(CLOCK_PROCESS_CPUTIME_ID, &ts) return time included kernel running. So, clock_gettime() and ru_utime of getrusage() aren't equivalent.

#3 - 12/03/2012 12:29 PM - kosaki (Motohiro KOSAKI)

And, if I understand the kernel source correctly, getrusage() and get_time(CLOCK_PROCESS_CPUTIME_ID) use the same clock source. So, I doubt this patch improve time accuracy.

Do anyone have a test result?

#4 - 12/04/2012 10:04 AM - tmm1 (Aman Gupta)

```c
#include
#include
#include
#include
#include
#include

double getrusage_time() {
    struct rusage usage;
    struct timeval time;
    getrusage(RUSAGE_SELF, &usage);
    time = usage.ru_utime;
    return time.tv_sec + time.tv_usec * 1e-6;
}

double clock_time() {
    struct timespec ts;
    if (clock_gettime(CLOCK_PROCESS_CPUTIME_ID, &ts) == 0) {
        return ts.tv_sec + ts.tv_nsec * 1e-9;
    }
    return 0.0;
}

int main() {
    int n;
    printf("getrusage() before: \%f\n", getrusage_time());
    for (n=0; n<10000; n++) pow(2, 2048);
    printf("getrusage() after: \%f\n", getrusage_time());
    printf("clock_gettime() before: \%f\n", clock_time());
    for (n=0; n<10000; n++) pow(2, 2048);
    printf("clock_gettime() after: \%f\n", clock_time());
}
```

#5 - 12/04/2012 12:24 PM - tmm1 (Aman Gupta)

Increasing the number of iterations shows the 4ms granularity.

```c
int loops = 1000000;
for (n=0; n<loops; n++) pow(2, 2048);
for (n=0; n<loops; n++) pow(2, 2048);

#6 - 12/04/2012 12:42 PM - tmm1 (Aman Gupta)

Increasing the number of iterations shows the 4ms granularity.

```c
int loops = 1000000000;
for (n=0; n<loops; n++) pow(2, 2048);
```
This issue was solved with changeset r38214.
Aman, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- gc.c (getrusage_time): uses clock_gettime() with CLOCK_PROCESS_CPUTIME_ID when available, which provides a 1ns precision on linux. [ruby-core:50495] [Bug #7500]