NPACI Blue Horizon
Using IBM SP Timing Routines

San Diego Supercomputer Center

Using IBM SP Timing Routines

Why ? - Time to solution is often best performance measure

Timer Requirements

Accurate - “adequate” resolution

Low access overhead - calling timer has no “significant” effect on program performance

Frequent timer reset to avoid “overflows”

Using IBM SP Timing Routines

Common Application Timers:

clock - supported by C
times - UNIX BSD 4.3, UNIX System V r4, POSIX
hetrusage - UNIX BSD 4.3, UNIX System V r4
ETIME,DTIME - Fortran language extensions - don’t use
gettimeofday - UNIX BSD 4.3, UNIX System V r4
system_clock - F90 language function - use with caution

Using IBM SP Timing Routines
rtc (real-time-counter) in Fortran

```
real*8 t1,t2,rtc
t1=rtc()
* code segment to be timed *
t2=rtc()
time=t2-t1
```

Using IBM SP Timing Routines

Portable Timing Routines in Fortran90

```
Part of F90 Language Standard

integer start, stop, rate

call system_clock(count_rate=rate)
call system_clock(count=start)

* code segment to be timed *

call system_clock(count=stop)
time = real(stop-start)/real(rate)
```

- count: machine-specific value is incremented by one for each clock count until the value count_max is reached; it is reset to 0 at the next count.
- count_rate: number of clock counts per second

Using IBM SP Timing Routines

Gettimeofday in C

```
#include <sys/time.h>
struct timeval timestr;
  double start,stop;
  void *Tzp=0;
      gettimeofday(&timestr, Tzp);
      start=(double)timestr.tv_sec+1.0E-06*(double)timestr.tv_usec;

    code segment to be timed *

      gettimeofday(&timestr, Tzp);
      stop=(double)timestr.tv_sec+1.0E-06*(double)timestr.tv_usec;
      printf("Time: %f %f %f\n",(stop-start),start,stop);
```

- tv_sec: seconds, tv_usec: microseconds

Using IBM SP Timing Routines Using gettimeofday in Fortran

- **C Code**

```
#include <sys/time.h>
double gettime();
 double gettime(void)
  {struct timeval timestr;
   void *Tzp=0;
  gettimeofday(&timestr, Tzp);
   return (double)timestr.tv_sec+1.0E-06*(double)timestr.tv_usec;}
```

- **Fortran Code**

```
double precision g1,g2
g1=gettime()
code segment to be timed *
time=gettime()-g1
```

Using IBM SP Timing Routines MPI_Wtime

```
include 'mpif.h’
double precision t1,t2
t1=MPI_Wtime()
code segment to be timed *
time=MPI_Wtime()-t1
```

Using IBM SP Timing Routines

References:

- Man pages
- IBM F90 Manual
- IBM MPI Manual