concordance=TRUE

Problem Set 5

Yue Hu

Oct 2017

1 problem 2

1= (-1)$^0 * 1.0 * 2^0, thatis, S = 0, d = 0, e = 1023$

2= (-1)$^0 * 1.0 * 2^1, thatis, S = 0, d = 0, e = 1024$

3= (-1)$^0 * 1.1 * 2^1, thatis, S = 0, d = 1, e = 1024$

d and e adds on for increasing integers.

to store $2^53 - 2, S = 0, dis$

$2^53 - 2 = (-1)^0 * 1.1111111111111111111111111111111111111111111111111110 * 2^52,$

that is, S = 0, d = 1111 11111111 11111111 11111111 11111111 11111111 11111110 (1 apperaring 51 times followed by 0), e = 1075

$2^53 - 1 = (-1)^0 * 1.1111111111111111111111111111111111111111111111111111 * 2^52,$

that is, S = 0, d = 1111 11111111 11111111 11111111 11111111 11111111 11111111(1 appearing 52 times), e = 1075

$2^53 = (-1)^0 * 1.0000000000000000000000000000000000000000000000000000 * 2^53,$

and when increrasing the lieast significant bit by 1, we get

$2^53 + 2 = (-1)^0 * 1.0000000000000000000000000000000000000000000000000001 * 2^53,$

So we can't represent $2^53 + 1$.

with the case of $2^54, e = 1023 + 54,$

so increasing the least significant bit which is the 52th bit of d, would result in an increas of $2^(-52+54) =$ 4.

we can see in R , bits of $2^53 - 1and2^53isdifferentbut2^53and2^53 + 1isthesame.$

```
library(pryr)
bits(2^53-1)

## [1] "01000011 00111111 11111111 11111111 11111111 11111111 11111111 11111111"

bits(2^53)

## [1] "01000011 01000000 00000000 00000000 00000000 00000000 00000000 00000000"

bits(2^53+1)

## [1] "01000011 01000000 00000000 00000000 00000000 00000000 00000000 00000000"
```

2 problem 3

2.1 a)

from the following code we can see it is faster to copy a large vector of integers than a numeric vector of the same length.

```
#integer case
x<- c(1:1e8)
object.size(x)

## 400000040 bytes

# let y point to x so when change one element of the vector a whole new copy is made.
y <- x
# note that if we use x[4] <- 50 the time is much longer
# since type of x changes from integer to double.
system.time(x[4] <- 50L)

##    user  system elapsed
##    0.08    0.05    0.13

# numeric case
x1<- rnorm(1e8)
y1 <- x1
object.size(x1)

## 800000040 bytes

system.time(x1[4] <- 50)

##    user  system elapsed
##    0.12    0.14    0.26
```

2.2 b)

the time to take a subset of size k=n/2 from an integer vector is a little bit faster.

```
x<- c(1:1e8)
x1<- rnorm(1e8)
system.time(sub <- x[1:5e7])

##    user  system elapsed
##    0.29    0.08    0.37

system.time(sub1 <- x1[1:5e7])

##    user  system elapsed
##    0.28    0.13    0.41
```

3 problem 4

3.1 a)

Because we should consider both the time for each computation and the the amount of communication that needs to happen.If we have very many tasks and each one takes little time, the communication overhead of starting and stopping the tasks will reduce efficiency, which will be the case if we do n individual column-wise computations.

3.2 b)

the amount of memory used:

A:each of the p workers will use n$^2 forX, n * mforYandn * mforresult, So(n^2 + 2nm)pintotal.$

B:each of the p workers will use n*m for X, n*m for Y and m*m for result, So (m$^2 + 2nm)pintotal.$

A uses more memory than B

the communication cost:

A:p task is needed and each task need to pass n*n for X, n*m for Y and n*m for result,

So C1 = (2mn+n$^2)pintotal.$

B:p*p task is needed and each task need to pass m*n for X, n*m for Y and m*m for result,

So C2 = (2mn+m$^2)p^2intotal.$

substract these two, and subsitute p=n/m:

C1-C1 = (2mn+n$^2) * n/m - (2mn + m^2) * (n/m)^2 = n^2(1 - n/m) < 0$

So A takes less Commication cost than B.

So B is better for minimizing memory use and A for minimizing communication