Stats243 Problem Set 5

Mengling Liu

Oct 2017

Collaboration: I disscussed with Rui Chen and Fan Dong for this homework set.
2.
$1 = 0\ 01111111111\ 0000000000000000000000000000000000000000000000000000$
$(-1)^0 \times 2^{2^{10}-1-1023} \times 1 = 1$
$2 = 0\ 10000000000\ 0000000000000000000000000000000000000000000000000000$
$(-1)^0 \times 2^{2^{10}-1023} \times 1 = 2$
$3 = 0\ 10000000000\ 1000000000000000000000000000000000000000000000000000$
$(-1)^0 \times 2^{2^{10}-1023} \times (1 + 2^{-1}) = 3$
$2^{53} - 1 = 0100001100111111111111111111111111111111111111111111111111111111$
$(-1)^0 \times 2^{1075-1023} \times (1.11...1)_2 = 2^{53} - 1$
$2^{53} = 0100001101000000000000000000000000000000000000000000000000000000$
$(-1)^0 \times 2^{1076-1023} \times (0.00...0)_2 = 2^{53}$
$2^{53} + 1 = 0100001101000000000000000000000000000000000000000000000000000000$
Same as the binary representation of 2^{53}
$2^{53} + 2 = 0100001101000000000000000000000000000000000000000000000000000001$
Spacing: Machine epsilon tell us also about the relative spacing of numbers. The absolute spacing is x*epsilon. The spacing of 2^{53} is $2^{53} \times 2^{-52} = 2$ The spacing of 2^{54} is $2^{54} \times 2^{-52} = 4$

```
library(pryr)

## Warning:  package 'pryr' was built under R version 3.4.2

bits(1)

## [1] "00111111 11110000 00000000 00000000 00000000 00000000 00000000 00000000"

bits(2)

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

bits(3)

## [1] "01000000 00001000 00000000 00000000 00000000 00000000 00000000 00000000"

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"

bits(2^53+2)

## [1] "01000011 01000000 00000000 00000000 00000000 00000000 00000000 00000001"
```

```r
library(data.table)
library(microbenchmark)
v_numeric <- rnorm(10^8)
v_integer <- 1:10^8

# v_numeric2 <- copy(v_numeric)
# v_integer2 <- copy(v_integer)

microbenchmark(copy(v_numeric),copy(v_integer), times=10L)

## Making a copy of integer vector is faster than making a copy of numeric vector

# Unit: milliseconds
#             expr      min       lq     mean   median       uq      max neval
#  copy(v_numeric) 324.2798 381.8769 476.5286 463.9413 583.2158 710.0574    10
#  copy(v_integer) 153.3201 158.7076 235.1404 161.0268 352.2862 388.1163    10

k <- length(v_numeric)/2

microbenchmark(sample(v_numeric,k), sample(v_integer,k), times=10L)

## It is faster to take a subset of size k from an integer vector of size n than from a numeric vector of

# Unit: seconds
#                  expr      min       lq     mean   median       uq      max neval
#  sample(v_numeric, k) 20.04916 20.64664 21.64265 21.85248 22.71800 22.83213    10
#  sample(v_integer, k) 18.07413 19.02444 20.00454 19.93404 21.28569 21.68283    10
```

4a) The communication cost will be higher if you break up Y into n individual column-wise computations than if breaking up Y into p blocks. The more work we divide, the higher than the comunication costs. If you have many tasks and each one takes little time, the communication overhead of starting and stopping the tasks will reduce efficiency.

b) Calculating amount of memory use and communication cost for approach A and approach B: Memory used: A: The memory required for X is n^2, and the jth subcolumn is nm, so the memory required for each workder is $n^2+nm+nm = n^2+2nm$. Thus the memory required for p workers is $(n^2+2nm)\times p = (n^2+2n(n/p)) = n^2(p+2)$

B: The memory required for X is nm and jth subcolumn of Y is $nm+nm$, and the memory required for each result is m^2. Thus the memory required for p workers at any given time is $(2nm+m^2)\times p = n^2(2+(1/p))$

Then B has a lower number of memory required than A.

Communication cost: A:The communication cost from master to workers and from master back is $(n^2+nm+nm)\times p = n^2(p+2)$ B: The communication cost from master to workers and from master back is $(2mn+m^2)\times p = n^2(2p+1)$

If $p>1$, then $2p+1>p+2$, then A has a lower number of communication costs than B.