$$\frac{1}{\log^c N} = (x+1)(x+k+1)$$

5 Wednesday

$$k\log^c N = x^2 + xk + x + x + k + 1$$
$$= x^2 + 2x + 1 + (x+1)k$$

~~$k\left(\log^c N - x - 1\right) = x^2 + 2x + 1$~~

~~$k = \frac{(x+1)^2}{(\log^c N - x - 1)}$~~

$$\frac{\log^c N - x^2 \mp 2x \mp 1}{(x+1)} = k$$

6 Thursday

$$\boxed{k = \frac{\log^c N - x^2 - 2x - 1}{(x+1)} = \frac{\log^c N - (x+1)^2}{(x+1)}}$$

Each rectangle has $\frac{\log^c N - (x+1)^2}{(x+1)}$ sequences to be searched and each search is $\log N$

MAY 2013
M T W T F S S M T W T F S S
1 2 3 4 5
6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29 30 31

Thus per rectangle time is

Friday 7

$O(\log^c N$ ~~[illegible]~~

Ka Shrinivaasan
23/8/2013

30/8/2013

Krishna iResearch Open Source design notes

(1) AstroInfer design for next version & future versions committed to SVN (initial version) - AstroInfer 11 and onwards.

(2) VIRGO - Virgo 1.0 has almost similar or better cloud execution capabilities (including kernel space execution) than SunRPC and other cloud libraries - More needs to be done:

Saturday 8

2.1) Memory pooling in cloud:
If an application wants to allocate 5GB RAM virgo malloc() should be able to do it failsafe. NUMA malloc implementations are available but using it in VIRGO could be incompatible (but needs to be explored).

Sunday 9

Every application code uses clone() and malloc() for spawning new thread or allocating memory. Virgo clone() has been implemented and works well in VIRGO 1.0 in all 3 modes (2 user, 1 kernel)

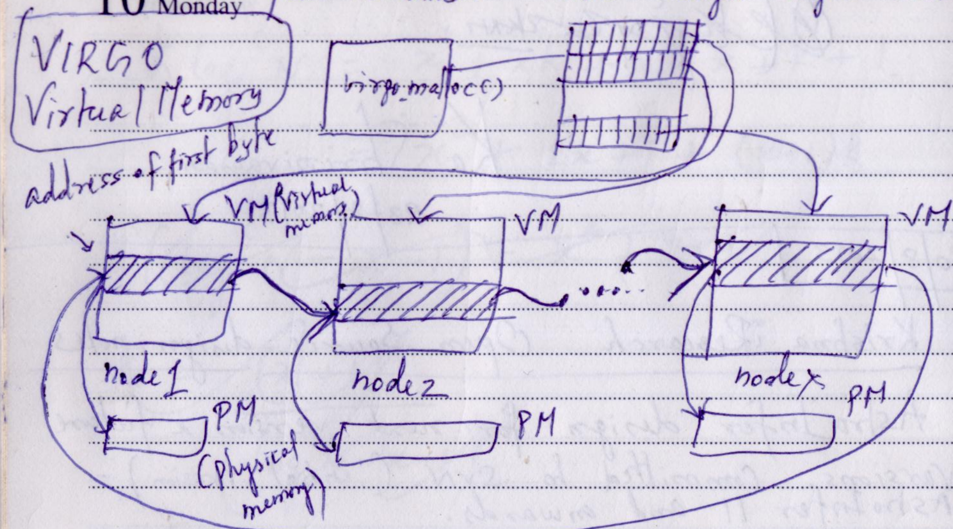
virgo_malloc():

JULY 2013
M T W T F S S M T W T F S S
1 2 3 4 5 6 7
8 9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31

Design of virgo_malloc() is quite

time consuming and could take

10 Monday few months. Crude design might look like



But is it worthwhile to implement virgo_malloc()?

References:

11 Tuesday

i) DOUG LEA's malloc article:
http://gee.cs.oswego.edu/dl/html/malloc.htm)

Option 1: Implementation of virgo_malloc as a wrapper for existing malloc()

Prototype of virgo_malloc() is as below:

p = virgo_malloc(sizeof() * number_of_objects)

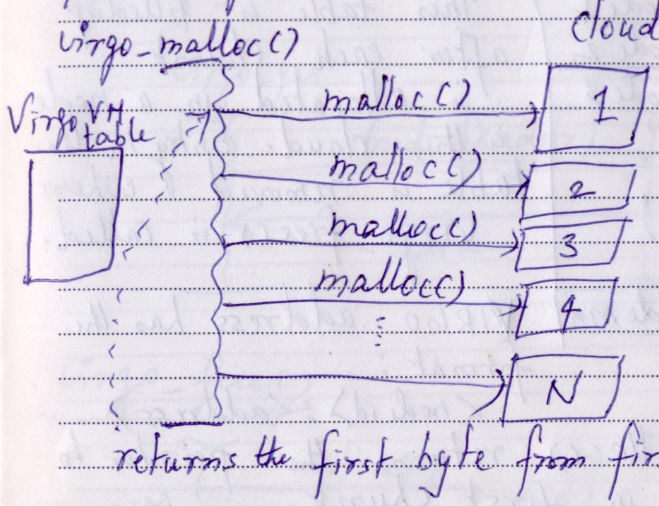
Parameter is the number of bytes to be allocated. This number of bytes is divided by number of nodes in cloud to get the number of bytes allocated per node. Thus the memory is scattered

MAY 2013	M	T	W	T	F	S	S	M	T	W	T	F	S	S
										1	2	3	4	5
	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	20	21	22	23	24	25	26	27	28	29	30	31		

and gathered from all nodes thereby pooling the memory. virgo_malloc()

12 Wednesday

internally sends commands to each cloud node sequentially as drawn below:



returns the first byte from first node.

To get and set a byte two functions are added:

13 Thursday

virgo_get(address)
virgo_set(address, data)

also virgo_free() is needed.

VIRGO Address Space:

There is a strong necessity for adding one more layer of virtual addressing that transcends all nodes in the cloud. For example if each node has 10GB ~~RAM~~ Virtual RAM and there are 5000 nodes in the cloud, 50 Terabytes of VIRGO addresses are possible including VRAM from all machines. Thus virgo address is from 0x0 to 0x5tera byte.

JULY 2013	M	T	W	T	F	S	S	M	T	W	T	F	S	S
								1	2	3	4	5	6	7
	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	22	23	24	25	26	27	28	29	30	31				

VIRGO VM Diagram is depicted in previous pages.

14 Friday

Virgo_malloc() also has a page translation table as below:

addr1-addr2	node 1
addr3-addr4	node 2
addr5-addr6	node 3
.	.
.	.
.	.
addrm-addrm+1	node m+1

This table is filledup after each chunk is allocated in a node within cloud. Entry in the table is removed when virgo_free() is called.

VIRGO address has the format:

<nodeid>:<address>

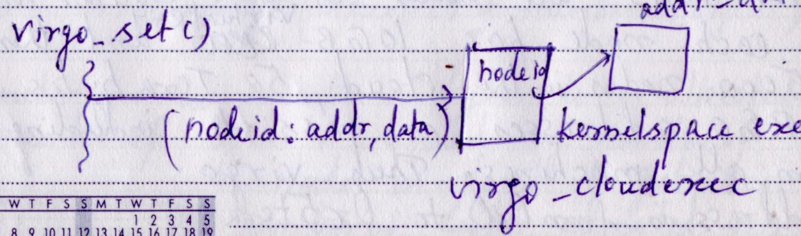
Thus virgo_malloc() returns the pointer to first byte in first chunk.

15 Saturday

virgo_set (address, data):

virgo_set() works by sending the command to the remote node by parsing the address and the data. The server side virgo_cloudexec() executing in kernel space which just calls the module that sets the byte with data.

16 Sunday

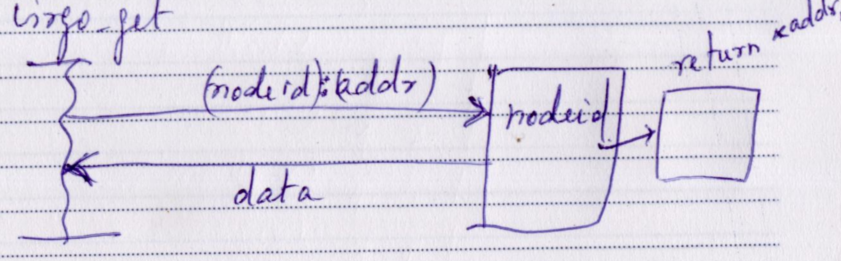


MAY 2013	M	T	W	T	F	S	S	M	T	W	T	F	S	S
										1	2	3	4	5
	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	20	21	22	23	24	25	26	27	28	29	30	31		

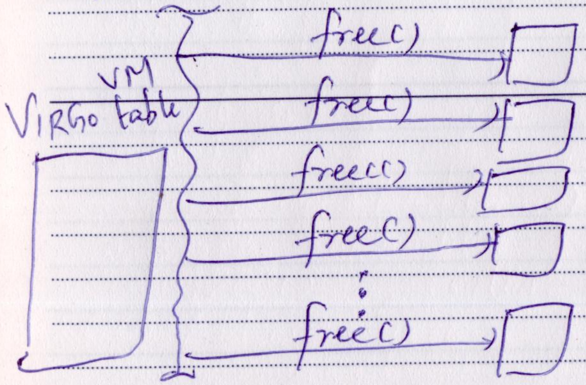
virgo_get (address):

17 Monday

virgo_get is symmetric that sends the addr to nodeid which is kernelspace exec-ed by virgo_cloudexec and returns data



virgo_free():



18 Tuesday

Thus virgo_malloc() is an application that runs on virgo cloud platform. Also VIRGO platform helps developing kernel level applications which would be otherwise difficult in other cloud implementations like Hadoop (Hbase, etc.,).

Ka Shrinivaasan

JULY 2013	M	T	W	T	F	S	S	M	T	W	T	F	S	S
								1	2	3	4	5	6	7
	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	22	23	24	25	26	27	28	29	30	31				

1/9/2013