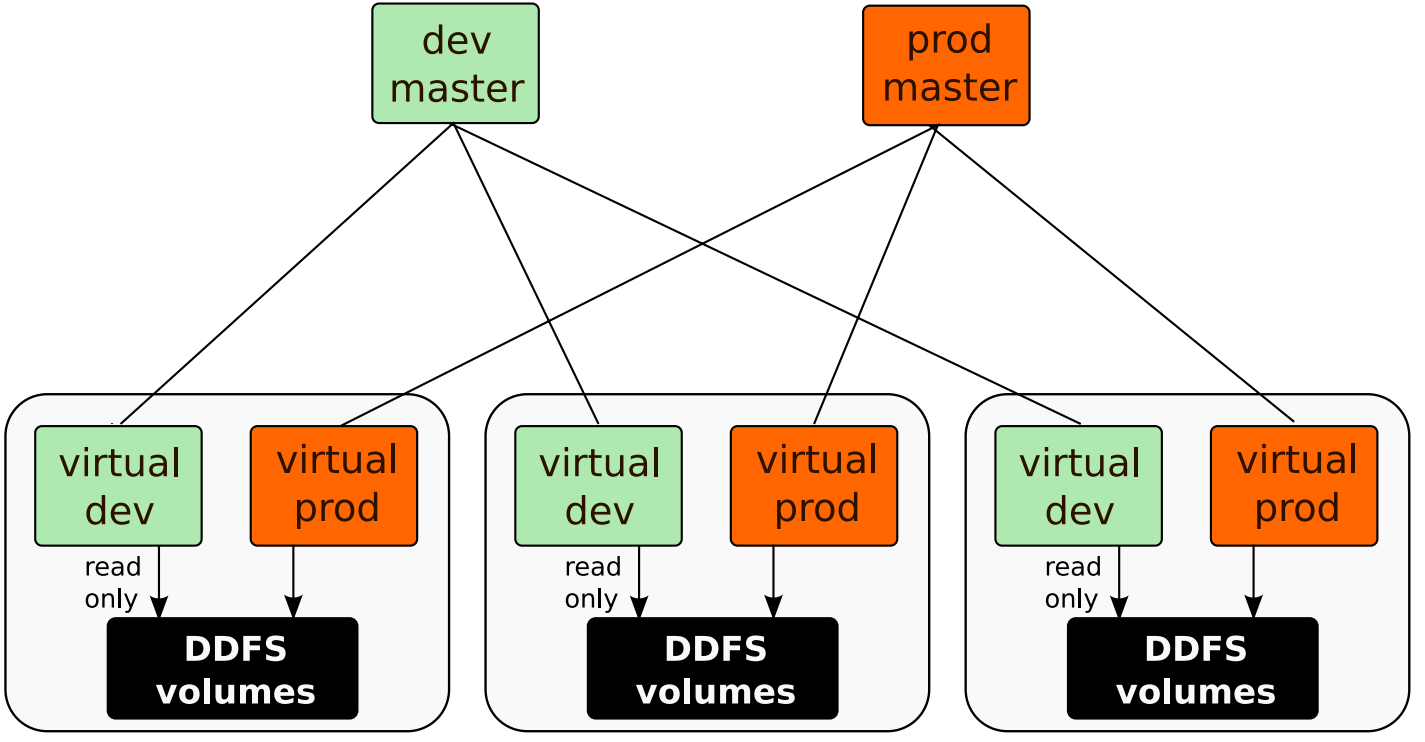
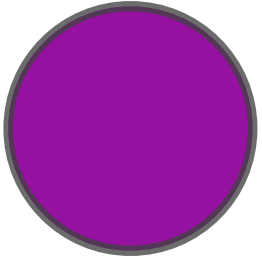
How Nokia uses **Disco** to mine **terabytes** of **logs** daily

Ville Tuulos
Nokia Research
Palo Alto

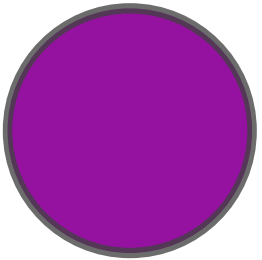
12pm


dev
master
prod
master
virtual
dev
virtual
prod
read
only
DDFS
volumes
virtual
dev
virtual
prod
read
only
DDFS
volumes
virtual
dev
virtual
prod
read
only
DDFS
volumes

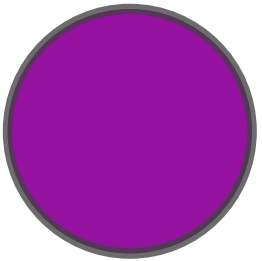
Data



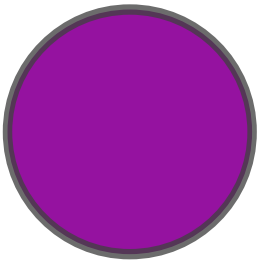
daily-log-2009-12-03_gz



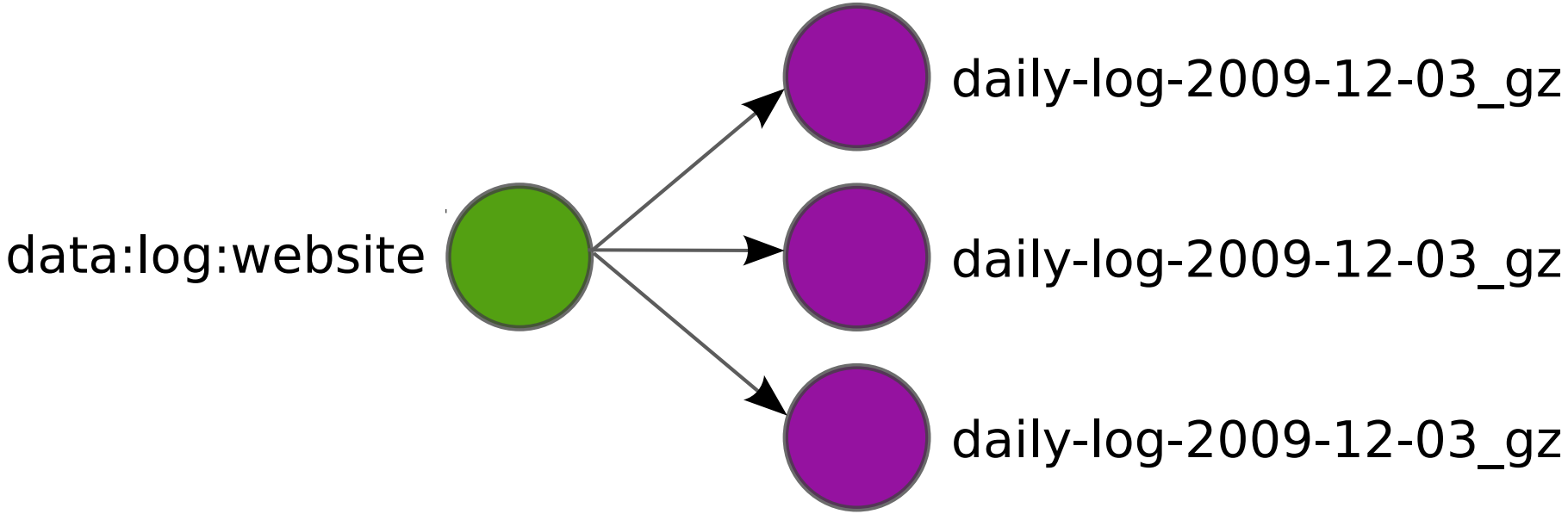
daily-log-2009-12-03_gz

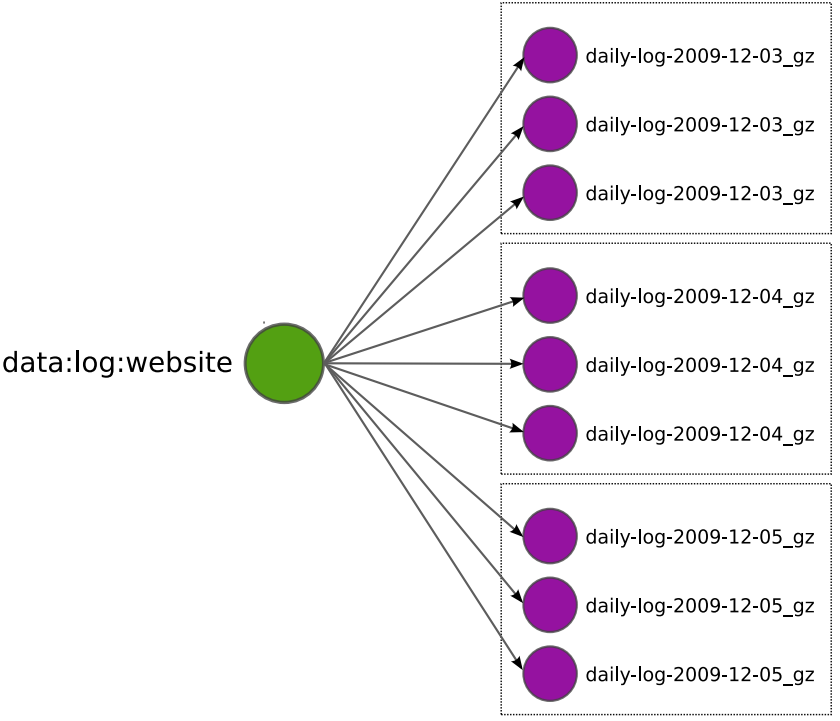


daily-log-2009-12-03_gz

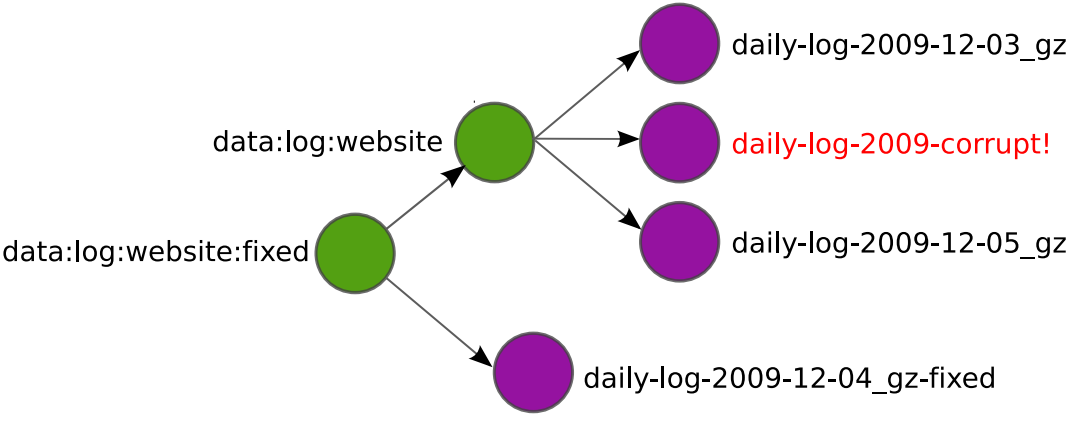


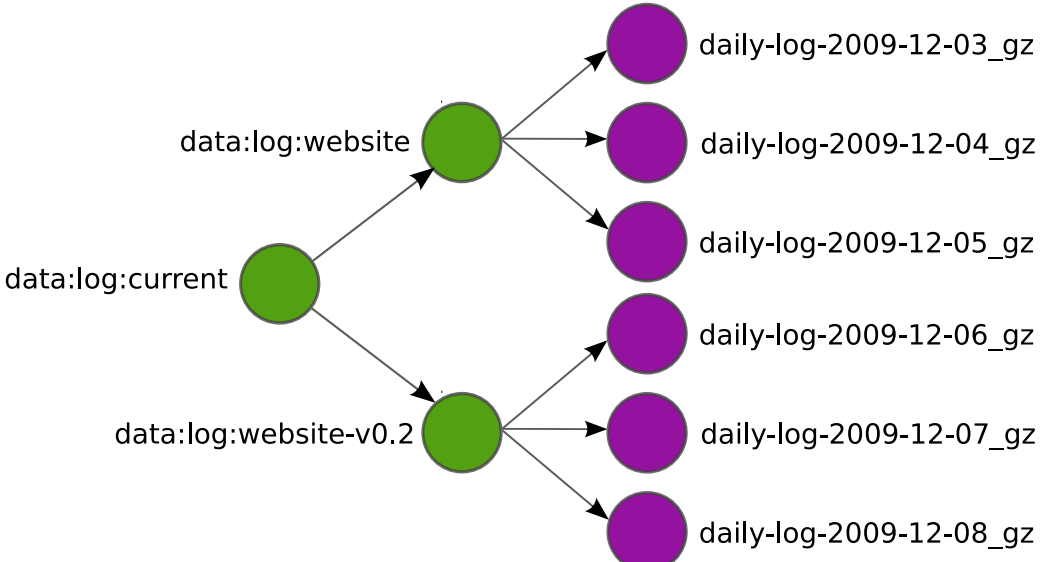
daily-log-2009-12-03_gz

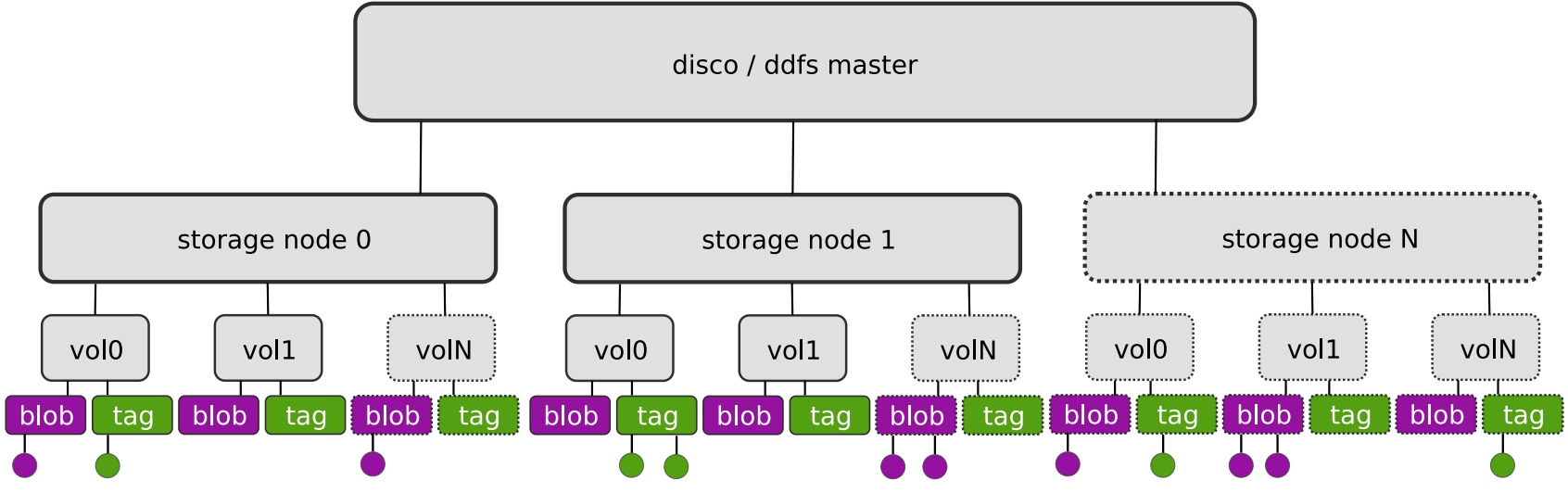

data:log:website
daily-log-2009-12-03_gz
daily-log-2009-12-03_gz
daily-log-2009-12-03_gz


data:log:website
daily-log-2009-12-03_gz
daily-log-2009-12-03_gz
daily-log-2009-12-03_gz
daily-log-2009-12-04_gz
daily-log-2009-12-04_gz
daily-log-2009-12-04_gz
daily-log-2009-12-05_gz
daily-log-2009-12-05_gz
daily-log-2009-12-05_gz

Pica (pronounced /ˈpaɪkə/ PY-kə) is a medical disorder characterized by an appetite for substances largely non-nutritive (e.g. metal, clay, coal, sand, dirt, soil, feces, chalk, pens and pencils, paper, batteries, spoons, toothbrushes, soap, mucus, ash, gum, etc.[1])


daily-log-2009-12-03_gz
data:log:website
daily-log-2009-corrupt!
data:log:website:fixed
daily-log-2009-12-05_gz
daily-log-2009-12-04_gz-fixed


data:log:website
data:log:current
data:log:website-v0.2
daily-log-2009-12-03_gz
daily-log-2009-12-04_gz
daily-log-2009-12-05_gz
daily-log-2009-12-06_gz
daily-log-2009-12-07_gz
daily-log-2009-12-08_gz


disco / ddfs master
storage node 0
storage node 1
storage node N
vol0
vol1
volN
vol0
vol1
volN
vol0
vol1
volN
blob
tag
blob
tag
blob
tag
blob
tag
blob
tag
blob
tag
blob
tag
blob
tag
blob
tag

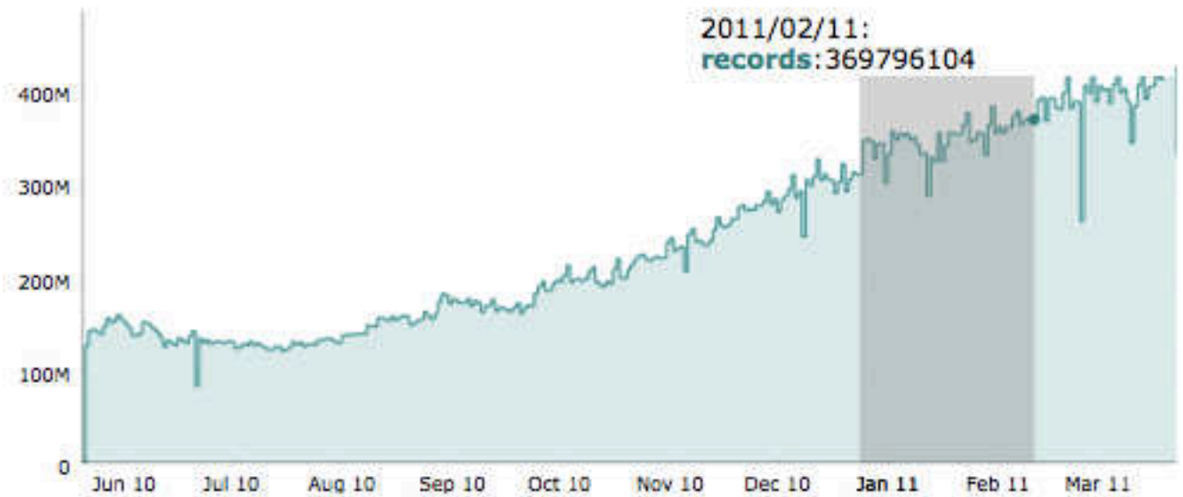
Show events

Tags checked ok	5745 / 5745
Root tag(s)	pica:vcdn_p_
Report generated	2011/03/23 16:42:52
Total size on disk	3668878 MB
# Files	54290
# Parsing errors	108580 records
# Outliers	0 records

Download as CSV

Daily data

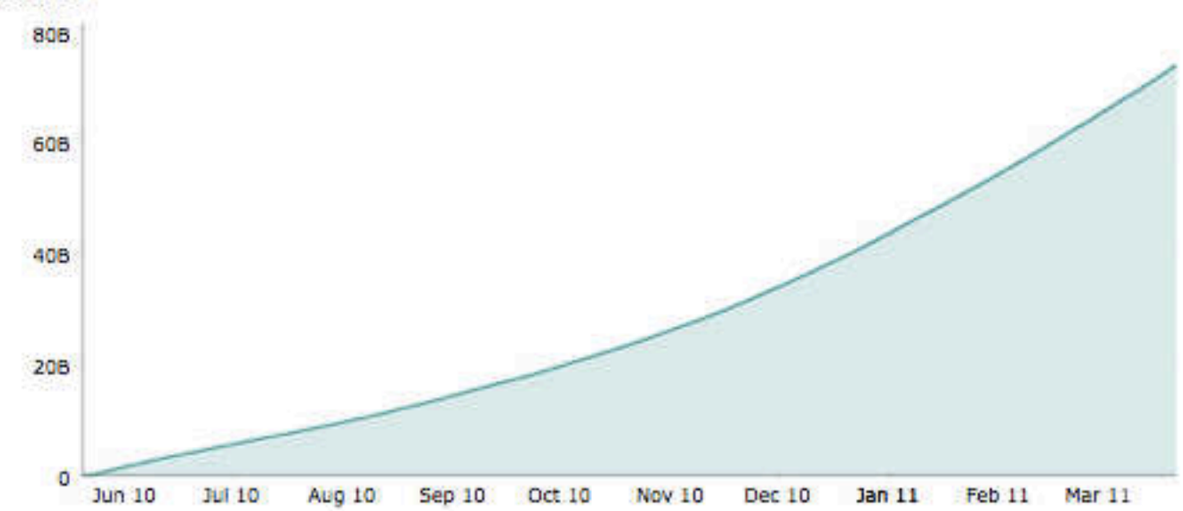
Records



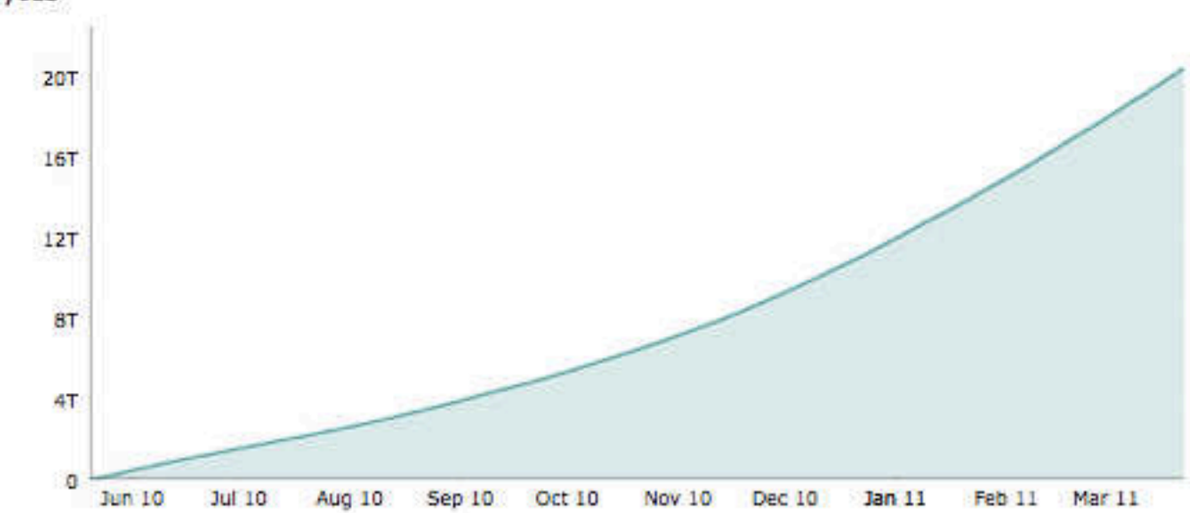
Bytes

Cumulative data

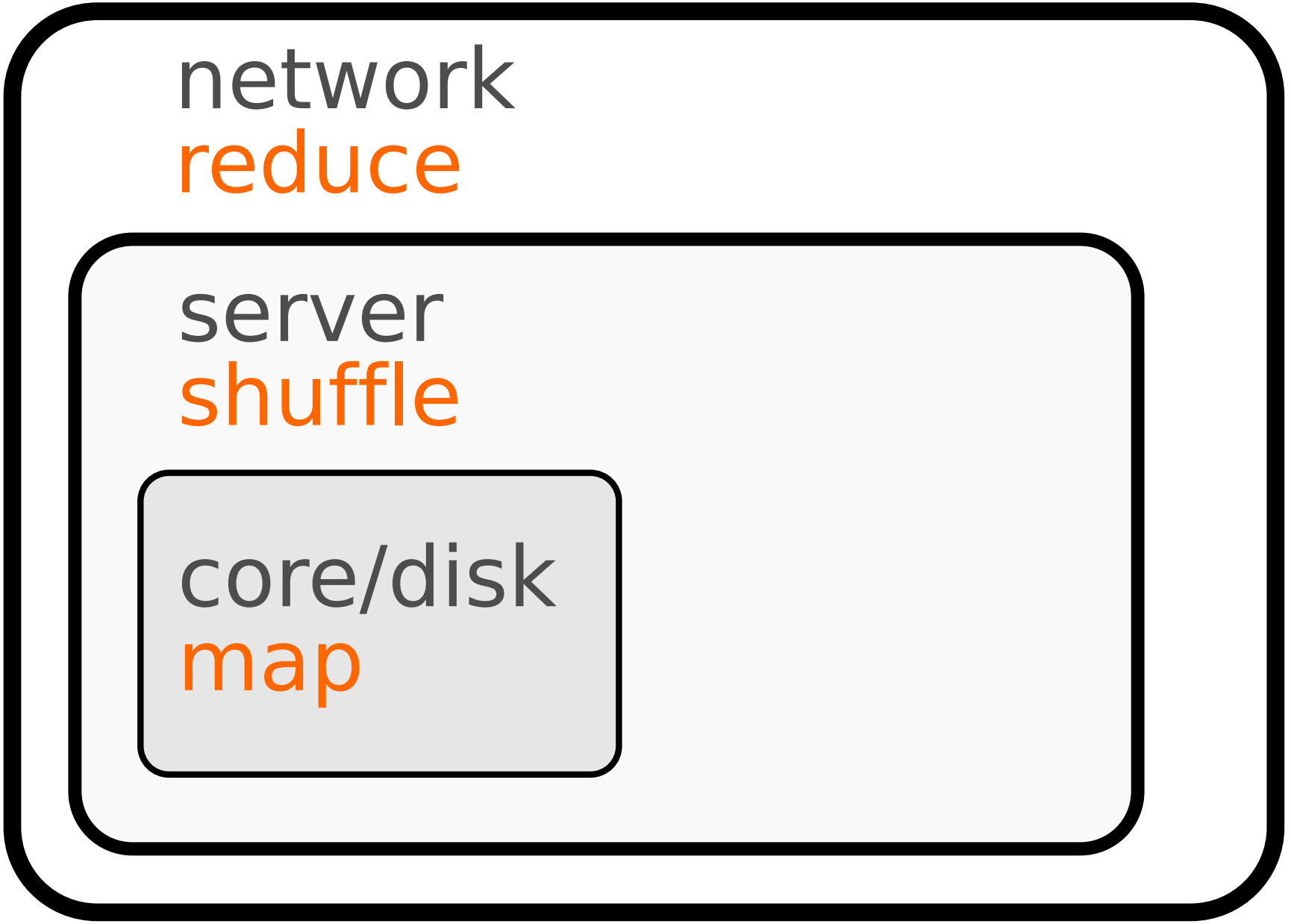
Records

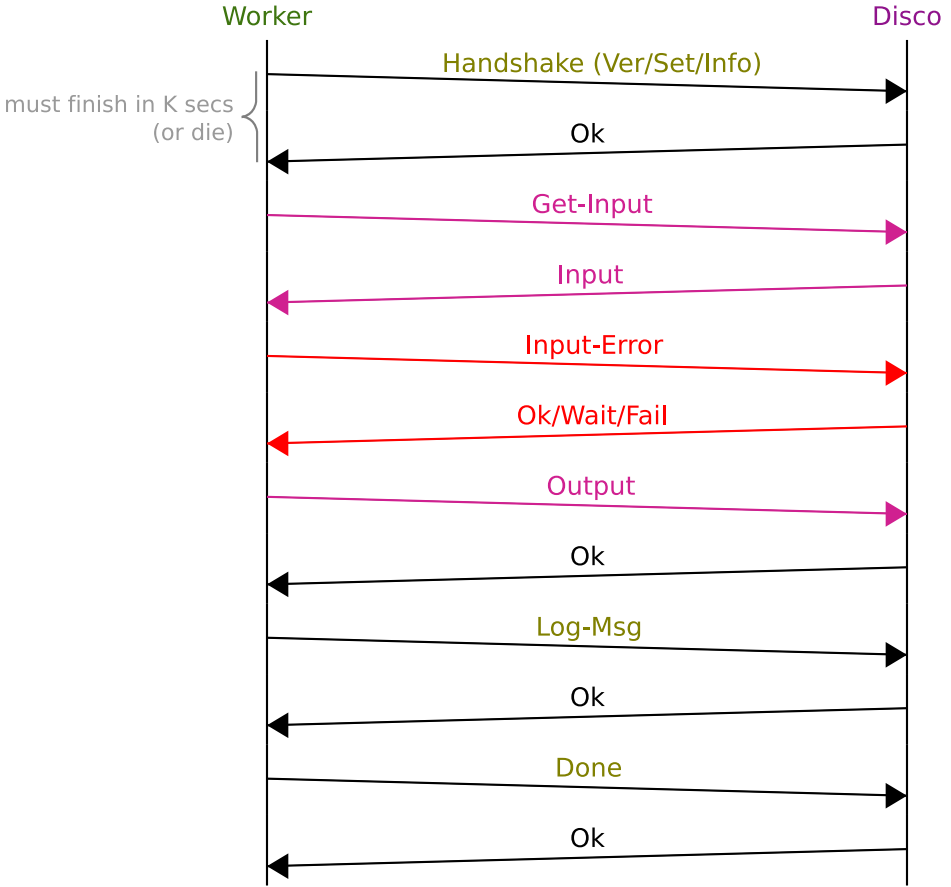


Bytes



Disco 0.4


network
reduce
server
shuffle
core/disk
map


Worker
Disco
Handshake (Ver/Set/Info)
must finish in K secs
(or die)
Ok
Get-Input
Input
Input-Error
Ok/Wait/Fail
Output
Ok
Log-Msg
Ok
Done
Ok

```
type disco = {
  taskname: string;
  input_url: string;
  input_size: int;
  out_channel: label:int option -> out_channel;
  ...
}
module type TASK = sig
  type map_init
  val map_init: disco -> map_init
  val map: map_init -> disco -> in_channel -> unit
  val map_done: map_init -> disco -> unit

  type reduce_init
  val reduce_init: disco -> reduce_init
  val reduce: reduce_init -> disco -> in_channel -> unit
  val reduce_done: reduce_init -> disco -> unit
end
```

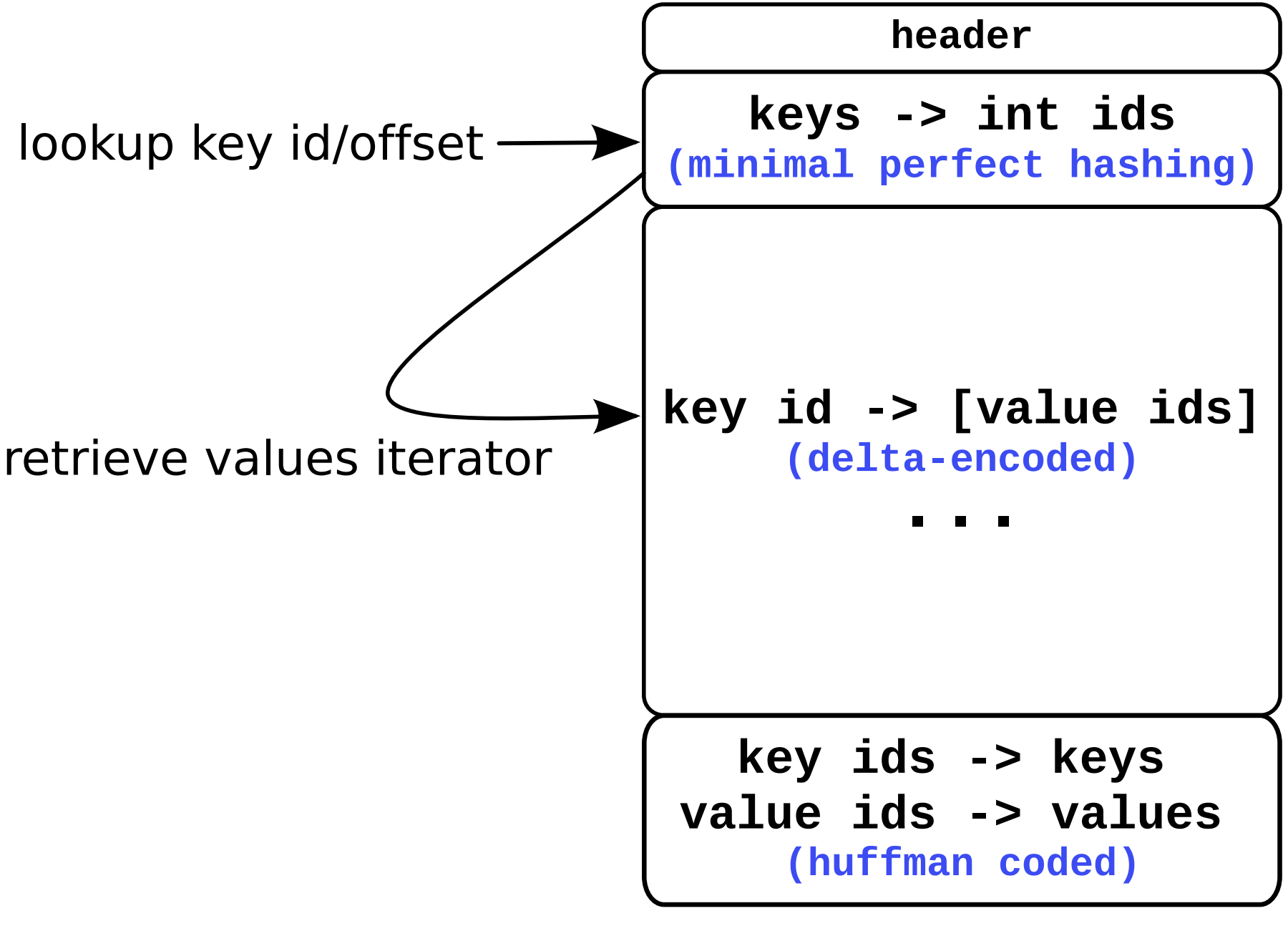
```
module U = Utils
module TestTask = struct
 type map_init = unit
 let map_init _ = ()

 let map _ disco in_chan =
  let rec loop () =
   List.iter (fun w ->
    match U.strip_word w with
     | "" -> ()
     | key -> output
        (disco.Task.out_channel ~label:None)
        ~key "1"
    ) (U.string_split (input_line in_chan) ' ');
    loop ()
  in try loop () with End_of_file -> ()

  let map_done _ _ = ()
```

```
type reduce_init = (string, int) Hashtbl.t
let reduce_init _ =
  (Hashtbl.create 4096 : (string, int) Hashtbl.t)

let reduce tbl _ in_chan =
 let lookup w = try Hashtbl.find tbl w with Not_found -> 0
 let rec loop () =
  let kv = U.string_split (input_line in_chan) ' ' in
  let k,v = (List.hd kv), int_of_string (List.nth kv 1) in
   Hashtbl.replace tbl k ((lookup k) + v);
   loop ()
 in try loop () with End_of_file -> ()

let reduce_done tbl disco =
  Hashtbl.iter (fun k v ->
                 output
                  (disco.Task.out_channel ~label:None)
                  k (Printf.sprintf "%d" v)
               ) tbl
```

```
let output oc ~key value =
  output_string oc (Printf.sprintf "%s %s\n" key value)

let _ =  Worker.start (module TestTask : Task.TASK)
```

Applications

DiscoDB

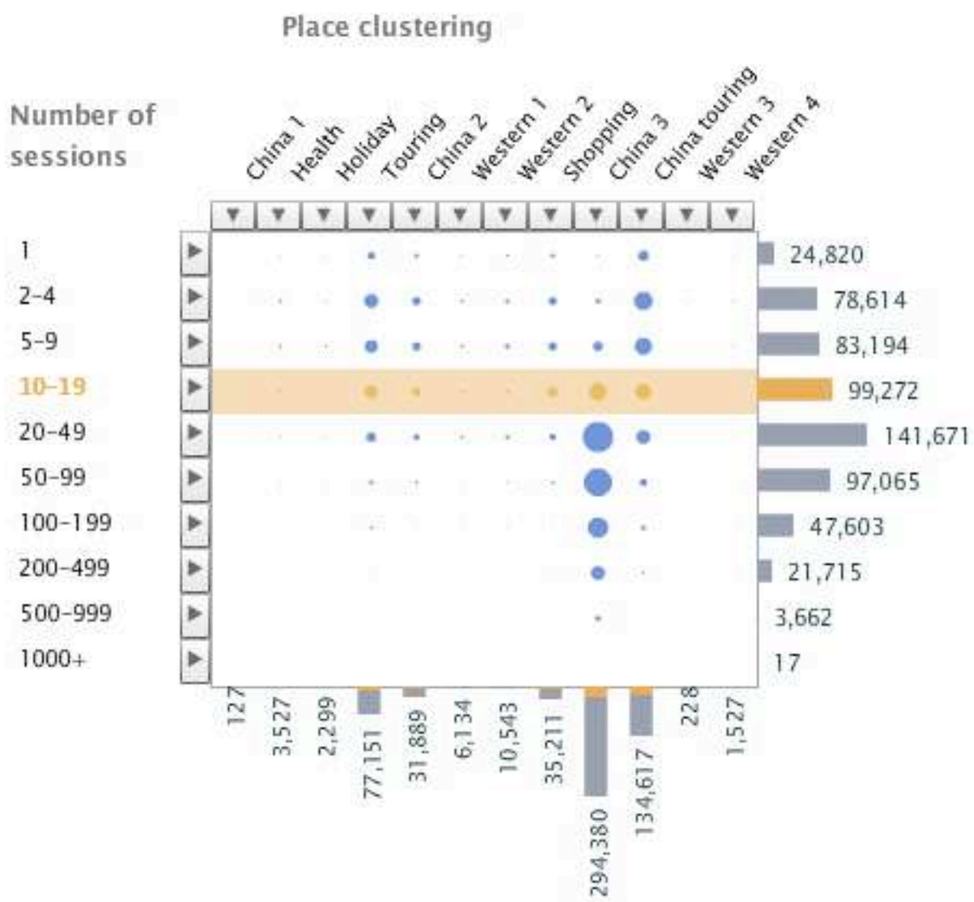

header
lookup key id/offset
keys -> int ids
(minimal perfect hashing)
retrieve values iterator
key id -> [value ids]
(delta-encoded)
. . .
key ids -> keys
value ids -> values
(huffman coded)

```
codewords: 65536
ctor 1391427 3.959 3.959 | 11110
tor: 1386230 3.944 7.903 | 11101
acto 1176920 3.349 11.252 | 10101
ovie 616388 1.754 13.006 | 110010
movi 615639 1.752 14.758 | 110001
vie: 615405 1.751 16.509 | 110000
ecto 210797 0.600 17.109 | 0100010
rect 210149 0.598 17.706 | 0100001
irec 210046 0.598 18.304 | 0100000
dire 209795 0.597 18.901 | 0011111
or:J 164065 0.467 19.368 | 11011010
or:M 124910 0.355 19.723 | 01111011
or:A 111276 0.317 20.040 | 01010111
or:R 96314 0.274 20.314 | 00011110
or:S 90096 0.256 20.570 | 00000001
```

N-MUX:
Nokia Maps Usage eXplorer

Ready to explore Nokia Maps service usage data, having [ ] requests from [ ] users thru [ ]. Choose different people and places for the matrix, navigate the map, change time views, or make selections to see changes in the data.

PEOPLE Number of sessions PLACES Place clustering

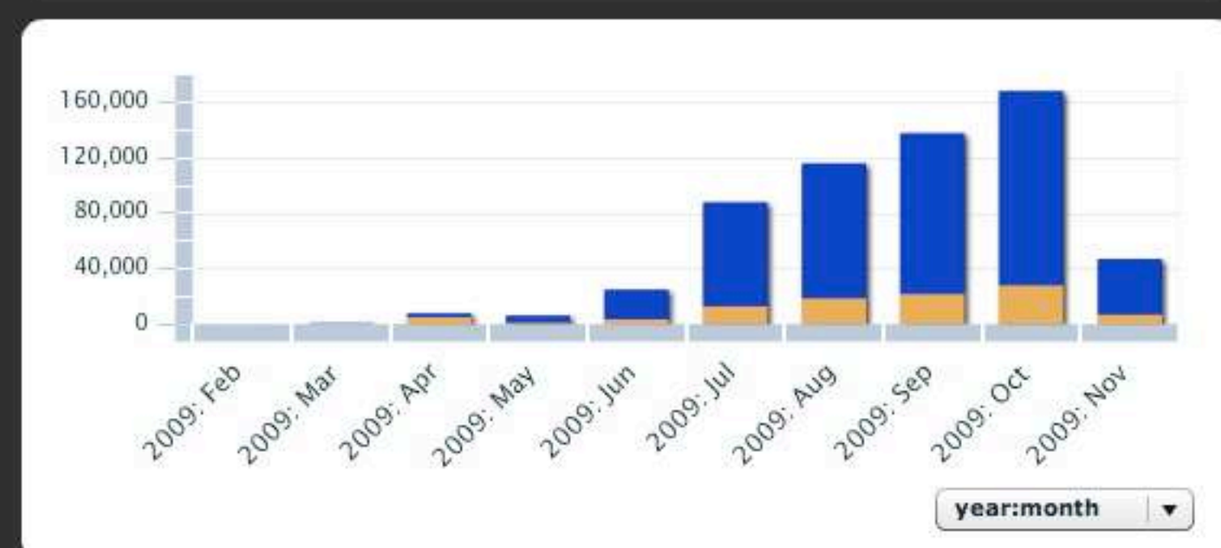


Clear all selections

Display any selection as:
- A: orange center
- B: orange occluding

Num map tiles w... 10 20 30 40

normal | selection | zoombox



discoproject.org

git clone git://github.com/tuulos/disco.git rc0.4

git clone git://github.com/pmundkur/odisco.git