Network Management Project

Jiannan Guo - ICT-Innovation Niklas Semmler - ICT-Innovation

29. November 2013

1 Task

1.1 Pseudocode

Listing 1: inspired by [1]

```
(ResponseTimeArriveMessage, responseTime)
    (UpdateVector, neighbor)
    (TimeOut, responseTime)
5
6
    objects and help functions:
7
          # storing values for and performing aggregation
8
            # communication messages
                           # aggregate local response times for local statistics
# combine local values with children's for statistics
9
    A.aggregate_local
10
    A.aggregate_sub_tree
                             # of the whole subtree
11
12
                             # values of statistics of subtree (including own)
                             # send a message to the own node with a delay
   Comm.send_to_self
13
14
    Comm.send_to_neighbors # send to all specified neighbors
                             # send to parent
15
    Comm.send_to_parent
                             # among neighbors, choose first with minimum level as
16
   find_parent
17
                              # new parent
18
19
    procedure GAP( )
20
        DELAY = # fixed time window
        Neighbors := empty;
21
22
        ResponseTimes := empty;
23
24
        if v = root then
25
            level = 0
            parent = 0
26
27
28
            level = INF
            parent = INF
29
        end if
30
31
32
        A.initiate()
33
        while true do
34
            read message;
35
            switch (message)
36
                 case (ResponseTimeArriveMessage, responseTime)
37
                     ResponseTimes.add(responseTime)
```

```
38
                    # to keep only a set of responseTimes in cache
39
                    Comm.send_to_self(DELAY, TimeOut(responseTime))
40
                    A.aggregate_local(ResponseTimes)
41
                    A.aggregate_sub_tree(Neighbors)
42
                case (UpdateVector, neighbor):
                    Neighbors.add(neighbor)
43
                    (newParent, newLevel) = find_new_parent(Neighbors)
44
45
                    A.aggregate_sub_tree(Neighbors)
46
                    if level != newLevel then
                        level = newLevel
47
48
                         parent = newParent
49
                         Comm.send_to_neighbors(Neighbors, level, parent, A.value())
50
                         continue
                    end if
51
52
                # to keep only a set of responseTimes in cache
53
                case (TimeOut, responseTime):
54
                    responses.remove(responseTime)
55
                    A.aggregate_local_value(ResponseTimes)
56
                    A.aggregate_sub_tree(Neighbors)
57
            end switch
58
            Comm.send_to_parent(Neighbors, A.value)
59
        end while
        end procedure
```

1.2 Implementation Details

A new base class was implemented which all of the tasks extend: peersim.EP2300.vector.GAPNode. In peersim.EP2300.message three new messages were introduced: ResponseTimeArriveMessage, UpdateVector, TimeOut. The first two roughly equal LOCALVAR and UPDATE from the original technical report (see [1]).

Each node stores all its information on it's neighbors as NodeStateVector in a SortMap named neighborList. Details on the nodes response times are stored in the ArrayList requestList. To focus only on the response times in the current time window, every response time is assigned a time out.

For the implementation with rate control, every message send to the parent will trigger a decrease of the msg budget. As soon as the msg budget is lower or equal to 0, no more messages are sent out. The message budget is reset via a new control peersim.EP2300.control.ResetMsgBudget.

1.3 Results

- (i) time series of f(t) and f(t) for $r = \{0.2,0.4,0.8\}$ and $\{R1\}$ from the first 5 minutes
- (ii) time series of f(t) and f(t) for $r = \{0.2, 0.4, 0.8\}$ and $\{R1\}$ after the first 5 minutes
- (iii) trade-off plots for both {R1,R2} (and all rate options?? 0.1,0.2,0.4,0.8,1.6)
- (iv) density plots for $r = \{0.2, 0.4, 0.8\}$ and $\{R1,R2\}$

Task 2

Listing 2: inspired by [1]

```
Comm.send_to_neighbors(Neighbors, level, parent, A.value())
continue
                                                                                aggregate local response times for local statistics combine local values with children's for statistics of the whole subtree values of statistics of subtree (including own) send a message to the own node with a delay send to all specified neighbors send to parent among neighbors, choose first with minimum level as
                                                                                                                                                                                                                                                                                                                                                                    case (ResponseTimeArriveMessage, responseTime)

# to keep only a set of responseTimes in cache
Comm.send_to_self(DELAY, TimeOut(responseTime))

A.aggregate_local(ResponseTimes)

A.aggregate_sub_tree(Neighbors)

case (UpdateVector, neighbor):

Neighbors.add(neighbor):

Neighbors.add(neighbor)

(newParent, newLevel) = find_new_parent(Neighbors)

A.aggregate_sub_tree(Neighbors)

if level != newLevel then
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             cache
                                                             aggregation
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              A.aggregate_local_value(ResponseTimes)
A.aggregate_sub_tree(Neighbors)
end switch
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             in
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            responseTimes
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          A.value)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     case (TimeOut, responseTime):
    responses.remove(responseTime)
                                                             and performing
         responseTime)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          Comm.send_to_parent(Neighbors,
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  newParent
                                                                                                                                                                      new parent
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          = newLevel
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             only a set of
                                                                       messages
                                                                                                                                                                                                DELAY = # fixed time window
Neighbors := empty;
ResponseTimes := empty;
                                                         # storing values for
comm # communication messa
.aggregate_local # agg
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                parent =
                                                   functions:
        ResponseTimeArriveMessage
UpdateVector, neighbor)
TimeOut, responseTime)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         level
                                                                                            #####
                                                                                                                                                                                                                                                                                                                                                 read message;
switch (message)
                                                                                                                 A.value
Comm.send_to_self
Comm.send_to_neighbors
Comm.send_to_parent
find_parent
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 į
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             to keep
                                                                                                                                                                                                                                                                                               INF
                                                                                                                                                                                                                                            = root then
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  end
                                                                                                                                                                                                                                                                                     INF
                                                                                                                                                                                                                                                                 0
                                                                                                                                                                                                                                                        0
                                                                                                                                                                                                                                                                                                                                          фo
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               procedure
                                                   and help
                                                                                                                                                                                                                                                                  II
                                                                                                                                                                                       procedure GAP( )
DELAY = # fi
                                                                                                                                                                                                                                                                                                                              A.initiate()
                                                                                                                                                                                                                                                                parent
                                                                                                                                                                                                                                                                                                parent
                                                                                                                                                                                                                                                                                                                                          while true
                                                                                                                                                                                                                                                        level
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    while
                                                                                                                                                                                                                                                                                      level
                                                                                                                                                                                                                                                                                                                                                    read
                               (TimeOut,
                                                                                                                                                                                                                                             >
                                                                                                                                                                                                                                                                           else
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   end
                                                                                                                                                                                                                                                                                                           end
                                                   objects
                                                                       Comm
                                                                                            ¥
```

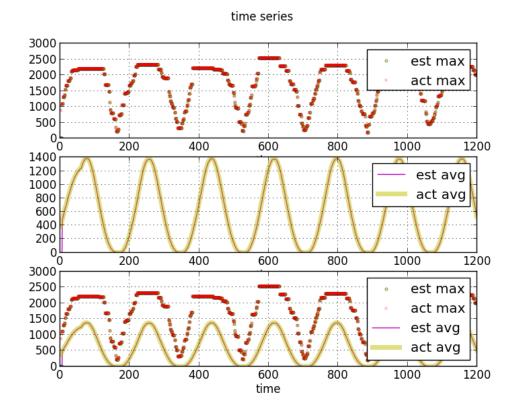


Abbildung 1: Time series plot

- compare performance for R_1, R_2
 - time series of f (t) and f(t) for r = { 0.2, 0.4, 0.8 } amd { R_1 , R_2 } from the first 5 min
 - trade off plot for R_1, R_2
 - density plot for $r = \{0.2, 0.4, 0.8\}$ and $\{R_1, R_2\}$

2 Task II

- pseudo code!
- implementation details
- compare performance for R_1, R_2
 - time series of f (t) and f(t) for r = { 0.2, 0.1, 0.05, 0.025, } amd $\{R_1, R_2\}$ from the first 5 min
 - trade off plot for R_1, R_2

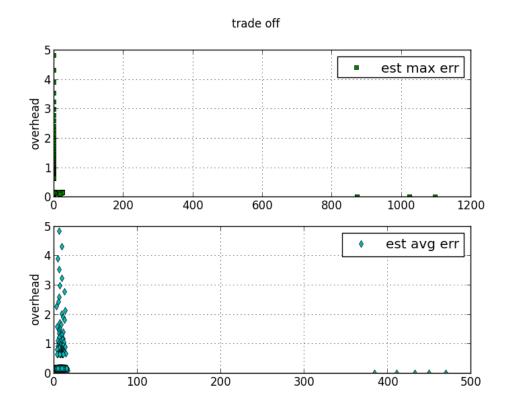


Abbildung 2: Trade off plot

– density plot for r = { 0.1, 0.05, 0.025 } and { R_1 , R_2 }

3 Task III

- pseudo code!
- implementation details
- compare performance for R_1, R_2
 - time series of f (t) and f(t) for r = { 0.2, 0.1, 0.05, 0.025, } amd $\{R_1, R_2\}$ from the first 5 min
 - trade off plot for R_1, R_2
 - density plot for $r = \{ 0.1, 0.05, 0.025 \}$ and $\{R_1, R_2\}$

4 Summary

• Compare TaskI, II, III

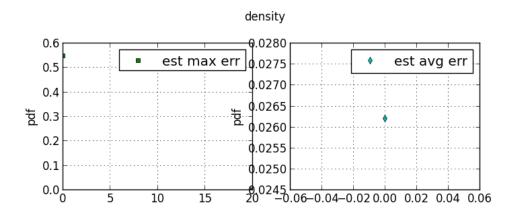


Abbildung 3: PDF plot

• Compare R_1, R_2 globally

Literatur

[1] R. Stadler, "Protocols for distributed management," 2012.