

Seminar Report: [seminar ID] (e.g. Paxy)

Name of team members

September 21, 2022

Upload your report in PDF format.

Use this LaTeX template to format the report.

A compressed file (.tar.gz) containing all your source code files must be submitted together with this report.

1 Introduction

Introduce in a couple of sentences the seminar and the main topic related to distributed systems it covers.

2 Code modifications

In this section we are going to show the code introduced in order to make the algorithm work.

2.1 Proposer.erl

```
1      %Provided code
2
3      round(Name, Backoff, Round, Proposal, Acceptors,
4      PanelId) ->
5          io:format("[Proposer ~w] Phase 1: round ~w proposal
6      ~w~n",
7          [Name, Round, Proposal]),
8          PanelId ! {updateProp, "Round: " ++ io\__lib:format(
9      "~p", [Round]), Proposal},
10         case ballot(Name, ..., ..., ..., PanelId) of
11             {ok, Value} ->
12                 {Value, Round};
13             abort ->
14                 timer:sleep(rand:uniform(Backoff)),
15                 Next = order:inc(...),
16                 round(Name, (2*Backoff), ..., Proposal, Acceptors
17                     , PanelId)
18
19      %Changed code
```

```

17
18     round(Name, Backoff, Round, Proposal, Acceptors,
19     PanelId) →
20         io:format("[Proposer ~w] Phase 1: round ~w proposal
21         ~w~n",
22         [Name, Round, Proposal]),
23         PanelId ! {updateProp, "Round: " ++ io\__lib:format(
24         "~p", [Round]), Proposal},
25         case ballot(Name, Round, Proposal, Acceptors,
26         PanelId) of
27
28             %Consensus, return {Value, Round}
29             {ok, Value} →
30                 {Value, Round};
31             abort →
32                 timer:sleep(rand:uniform(Backoff)),
33                 % Try again after sleeping, increment round and
34                 sleeptime
35                 Next = order:inc(Round),
36                 round(Name, (2*Backoff), Next, Proposal,
37                 Acceptors, PanelId)
38         end.

```

In this part of the code, we can see what we have introduced to make *round* function work. The first change are the parameters added to the case statement, which are the *Round*, *proposal* and *acceptors*. The next modification is in case we receive an *abort*. As observed above, *Next* variable is defined as the increment of *Round* and we use this new incremented variable as *Round* in the recursion calling *round* function.

3 Experiments

Provide evidence of the experiments you did (e.g., use screenshots) and discuss the results you got. In addition, you may provide figures or tables with experimental results of the system evaluation. For each seminar, we will provide you with some guidance on which kind of evaluation you should do.

4 Open questions

Try to answer all the open questions in the documentation. When possible, do experiments to support your answers.

5 Personal opinion

Provide your personal opinion of the seminar, indicating whether it should be included in next year's course or not.