Distributed Polling System: Algorithms to calculate Final result of voting

Version 1.0

Table of Contents

1. Introduction

1.1 Purpose of this document

This document is a deliverable of architecture and design phase from DPS project team as derived from requirement analysis. In this document the algorithms for calculating the result of polls is described in detail. Different aspects that should be considered for calculating and announcing the results are discussed and also detailed architecture for calculating result is explained.

1.2 Intended Audience

Intended Audience of this document is:

- Customers
- Supervisors
- Project Team
- Other interested parties

1.3 Scope

This document gives detailed design description for vote capturing and how it is being processes in Web-DPS data store. Web-DPS application is mainly emphasized in this document for deriving, announcing and maintaining voting information through the means of database objects and underlying algorithm.

2. Architecture of capturing votes and result announcement

Figure 2-1 depicts detailed architecture of Web-DPS vote calculation methodology and capturing votes from SMS Gateway, Email server. Stored procedure API does all verifications of duplicate voting, medium of voting and so on during receiving a vote, voted by a member.

Once the API accepts a vote, it inserts in Poll_answer table the vote of that specific member. On each insert there is a trigger that calculates whether votes received for a particular poll is more than 50%.

If it's more than 50%, verifyVote() stored procedure is invoked that verifies whether a particular option (in a poll) has received maximum(>50%) and result can be announced.

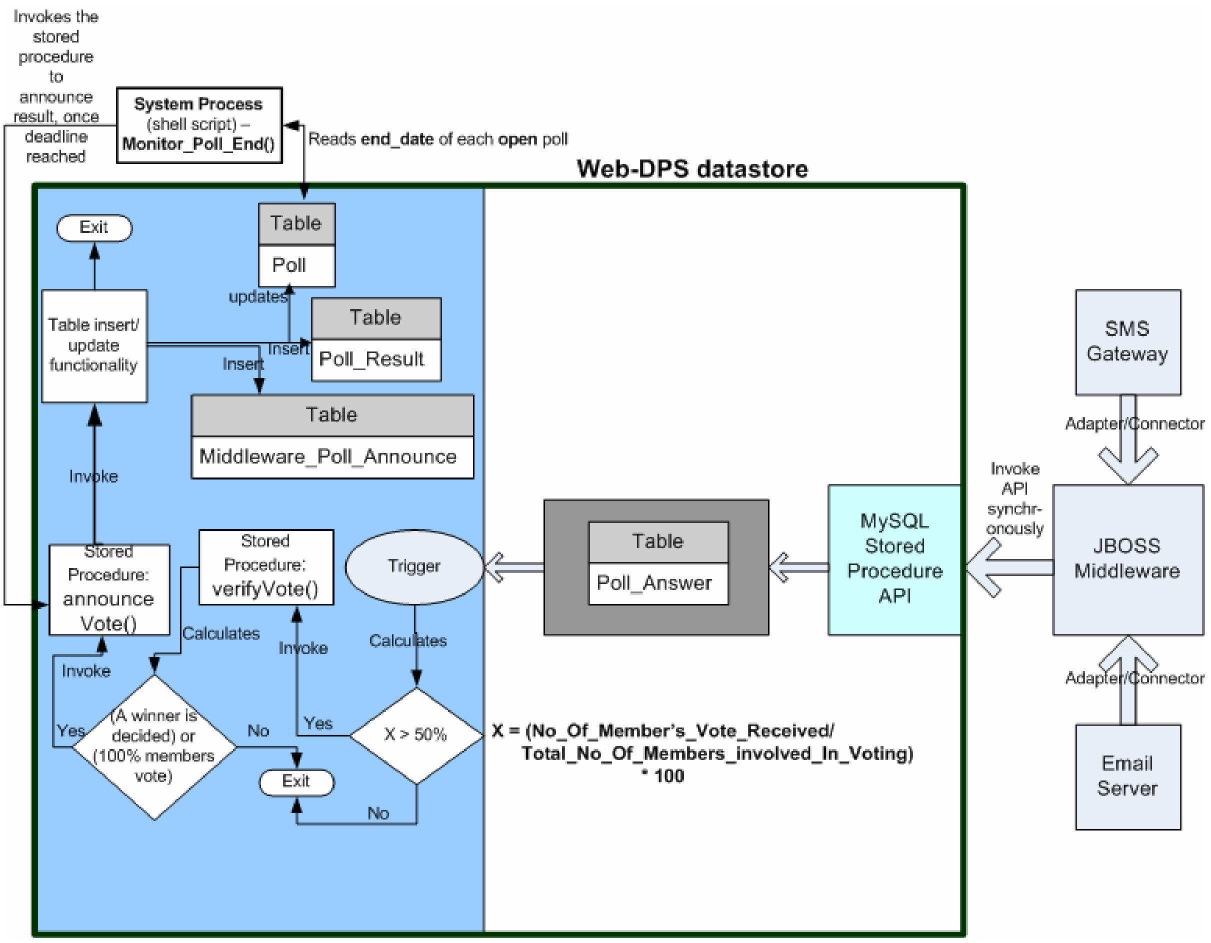
If any particular option receives more than 50% votes or when all the members involved in that poll have voted, we invoke announceVote() stored procedure.

This procedure updates internal tables to indicate the poll status, its result and inserts an event in Middleware_Poll_Announce table for middleware to intimate members through SMS and email.

Steps:

1. All user votes from Email server and SMS Gateway are captured by middleware with the help of its adapters/connecters. Then parsed and processed by Jboss middleware at its hub.
2. Middleware invokes Web-DPS stored procedure API which performs initial validation, authentication, duplicate verification and so on functionalities.
3. Once successfully verified, a vote is updated in two main tables in Web-DPS database as **Poll_Response** and **Poll_Answer.**
   Poll_Response defines whether an user has voted or not and with some other details.
   Poll_Answer table stores actual voting information details
4. A database trigger (on insert) associated with the *Poll_Answer* table is triggered to verify whether total number of votes (for voting type: Yes/No, One Option among multiple Options, Multiple Choice Option) received (X) is more than 50 percent, so that we can start verifying if any option has received more than 50 percent and announcement can happen.
5. if total votes received is more than 50 percent, we start to invoke **verifyVote()** stored procedure to verify if any option has received more than 50 percent votes. (Note: Even when votes received is hundred percent, the same process will follow and in **verifyVote()** procedure we will just pass the handle to the next steps)
   else exit();
6. If a single option has received more than 50 percent votes or 100% members voted, we invoke **announceVote()** stored procedure to calculate and announce the poll result. This **announceVote**() procedure inserts the Poll_result table to maintain poll results, Poll table to

7. There is one more process, an Unix shell script, that reads the end_date of each open (poll_status column) poll from **Poll** table and as soon as any dead-line is reached, it invokes **announceVote()** procedure to calculate and announce the voting result.



Invokes the stored procedure to announce result, once deadline reached
System Process (shell script) – Monitor_Poll_End()
Reads end_date of each open poll
Web-DPS datastore
Exit
Table
Poll
updates
Table
Poll_Result
Table insert/ update functionality
Insert
Insert
Table
Middleware_Poll_Announce
Invoke
Stored Procedure: announce Vote()
Stored Procedure: verifyVote()
Trigger
Calculates
Invoke
Calculates
Invoke
Yes
(A winner is decided) or (100% members vote)
No
Yes
X > 50%
Exit
No
X = (No_Of_Member's_Vote_Received/ Total_No_Of_Members_involved_In_Voting) * 100
Table
Poll_Answer
MySQL Stored Procedure API
Invoke API synchr-onously
SMS Gateway
Adapter/Connector
JBOSS Middleware
Adapter/Connector
Email Server

Figure 2-1: Low level architecture of Vote capture, calculation and announcement

Note: In announceVote(), if result contains equal percentage of vote received for two or more options, still the vote will be announced and poll creator will decide how to resolute the same. We feel, one solution might be to create a new poll asking members to choose one most preferred choice among the options.

3. MySQL Stored Procedure API

```
DROP PROCEDURE IF EXISTS jbpmdb.captureVoting;

CREATE PROCEDURE jbpmdb.`captureVoting`(IN MSISDN varchar(30),IN IMSI varchar(30),IN VOTING_RESPONSE varchar(40),IN ISANONYMOUS
varchar(50),IN POLL_ID varchar(10),IN POLL_STRING varchar (100),IN EMAIL_ID varchar(100),OUT RESULT varchar(500))

BEGIN
declare pollstatus varchar(10) default null; /*set pollstatus="";*/
declare pollid int;
declare pollname varchar(20) default null;
declare polltype varchar(20) default null;
declare isvoted varchar(20) default null;
declare userid int default 0;
declare phoneMSISDN varchar(20) default null;
declare phoneIMSI varchar(15)default null;
declare emailid varchar(40)default null;
declare verify_emailid varchar(40) default null;
declare verify_pollid int default 0;
declare polloptionid int default 0;
set RESULT="THANKS FOR VOTING: WE WILL CONSIDER YOUR OPINION DURING ANOUNCEMENT";

    if(POLL_STRING is not null) then select poll_id into pollid from poll where poll_name=POLL_STRING;
                        else set pollid=POLL_ID;
        end if;
        if(EMAIL_ID is not null) then set emailid=EMAIL_ID;/*set RESULT=CONCAT(RESULT,',',emailid);*/
                    else select corporate_email into emailid from user_corporate_account where corporate_phone_num=MSISDN and imsi=IMSI;
        end if;
        select x.user_id,y.is_voted into userid,isvoted from user_corporate_account x, poll_response y where x.corporate_email=emailid and y.poll_id=pollid and
x.user_id=y.user_id;
        select a.poll_status,
        a.poll_id,
        a.poll_name,
        a.poll_type,
        c.user_id,
            c.corporate_phone_num,
            c.imsi,
            c.corporate_email into pollstatus,pollid,pollname,polltype,userid,phoneMSISDN,phoneIMSI,emailid
            from poll a,user_corporate_account c,poll_members d where d.poll_id=pollid
            and c.corporate_email=emailid
            and c.user_id=d.user_id
            and a.poll_id=d.poll_id
            and c.user_status='ACTIVE';

        IF(pollstatus='open')
              THEN if((MSISDN is not null) and (IMSI is not null) and (MSISDN=phoneMSISDN) and (IMSI=phoneIMSI))
                     then if(isvoted is null)
                             then  if(polltype='1')
                                      then if ((VOTING_RESPONSE='YES') or (VOTING_RESPONSE='NO'))
                                              then  insert into poll_response values (pollid,userid,'YES','SMS',ISANONYMOUS,sysdate());
                                                  insert into poll_answer values (userid,(select poll_option_id from poll_option where poll_id=pollid and
                                                   poll_option=VOTING_RESPONSE),1);
                                              else set RESULT="ERROR: INCORRECT VOTING RESPONSE RECEIVED FROM MEMBER";
                                         end if;
                                       else set RESULT="****NOT YES/NO TYPE VOTING********";
                                  end if;
                             else set RESULT="ERROR: MEMBER IS INTENDING FOR DUPLICATE VOTING THROUGH SMS";
                        end if;
                     else if((EMAIL_ID is not null) and (EMAIL_ID=emailid))/* and (EMAIL_ID=emailid))*/
                          then if(isvoted is null)
                                 then  if(polltype='1')
                                     then if ((VOTING_RESPONSE='YES') or (VOTING_RESPONSE='NO'))
                                                            then insert into poll_response values (pollid,userid,"YES","EMAIL",ISANONYMOUS,sysdate());
                                                                 insert into poll_answer values (userid,(select poll_option_id from poll_option where poll_id=pollid and
                                                                 poll_option=VOTING_RESPONSE),1);
                                                            else set RESULT="ERROR: INCORRECT VOTING RESPONSE RECEIVED FROM MEMBER";
                                          end if;
                                        else set RESULT="****NOT YES/NO TYPE VOTING********";
                                       end if;
                                  else set RESULT="ERROR: MEMBER IS INTENDING FOR DUPLICATE VOTING THROUGH EMAIL";
                                 end if;
                          else set RESULT="ERROR: INVALID EMAIL ID;INTENTION OF FRAUDULENT ACTIVITY";
                    end if;
```

```
                                        end if;
                ELSE  if(pollstatus='closed')
                                    then set Result="UNFORTUNATE: POLL HAS ALREADY BEEN CLOSED";
                                                                      else select corporate_email into verify_emailid from user_corporate_account where corporate_email=emailid;
                                                                           select distinct(poll_id) into verify_pollid from poll where poll_id=pollid;
                                                                            if (verify_emailid is null)
                                                                                                  then set RESULT="EXCEPTION: SUPPLIED EMAIL ID or MSISDN/IMSI IS WRONG";
                                                                                                else  if (verify_pollid=0)
                                                                                                                  then set RESULT="EXCEPTION: SUPPLIED POLL_ID or POLL STRING IS
WRONG";
                                                                                                            end if;
                                                                            end if;
                            end if;
              END IF;
END;
```

4. Algorithms to calculate final result of voting

Before discussing about algorithms for calculating results, different available types of voting and different times for announcing results should be considered. According to types of voting, in some cases the result can be announced even before all members have been voted or before finishing the time of voting.

4.1 Different possible vote types

As per DPS project and its requirements, four different kinds of voting might be created by the members of the system. These are as follows:

- ***Yes/No Choice Option***

  There are two options Yes and No, the voter should select one of them.
- ***One Choice Option***

  One option can be selected among multiple (two or more) options.
- ***Multiple Choice Option***

  Voter is able to select one or more options form the available options. In some cases it might be specified to select just specific numbers of options. In this type of voting the priority of all options are equal.
- ***Priority Option***

  This kind of voting is a kind of 'Multiple Choice Option' with specifying the priority of each selected option.

4.2 When result should/can be calculated

The validity of a received vote is from its defined start time till its end time. Any received vote during this period should be validated and if passed from this step, it will be saved in the system. Therefore different correct votes will be added to the system one by one. The question is when system is able to check the possibility of publishing the result and when it should announce the result. Therefore three different times can be discussed as below:

- End time or deadline of a poll has reached.
- All members of a poll have voted.
- A single option has reached maximum (>50%)

In the first two cases, the result should be calculated and announced to members of the poll. Then the poll result should be saved in the system and the poll status should be changed to 'Closed'.
The third case happens in some cases. This situation might occurs for some kinds of voting (Yes/No, One Choice Option, Multiple Options) where a single option has reached maximum. So there is no need to wait for receiving votes from other poll members. The final result can be calculated and announced to members. This time also, the result will be saved and the poll status should be changed to 'Closed'.

4.3 Algorithms to calculate final result

All received votes of polls are saved in the system; the numbers of voters and the information of polls are also available (in Poll, Poll_Members, Poll_Response, Poll_Answer tables). All checking, validation of received votes is done completely by stored procedure API (invoked from middleware layer) and acceptable votes are saved. For example, checking non-duplicate receiving votes from a member, receiving between start and end time of an open poll, validity of received information and all verification is done.
According to the type of each poll the result of voting can be calculated. In this section, algorithm for calculating final results is described.
In each part, if the final result can be specified, it will be saved in database of the system and related information will be updated. The result will also be announced to all members of the poll.

4.3.1 Algorithm for calculating result in Yes/No Choice Option

Different times and how to calculate the result are as follows:

- A new vote is added in the system,

Checking number of received votes and number of voters for this poll; if number of received votes (which is including the current added vote) is more than half of the number of voters for this poll, then the percentage of each received option will be checked. If the maximum percentage of any of these two options is more than 50%, will be the winner. Otherwise the result can not be calculated I,e announceVote() can not be invoked.

```
If (Number_of_Received_Votes > (Number_of_Voters / 2))
        for each Voting Option  i
                If (Number_of_votes_received_for_i>50% )
                {       Final_Result= Option i
                        announceVote();
                        Break;
                }
```

- At end time of poll i,e deadline reached or when all members have been voted,
  The result is the maximum number of received option Yes or No. If there is a 50-50 scenario, still we will announce the result specifying both the options and poll creator should decide the next steps. Might be he can come up with a new poll to finalize the same.

4.3.2 Algorithm for calculating result in One Choice Option

Calculation of the result in this type of voting is almost the same as the calculation result in Yes/No choice option. We can say 'Yes/No' voting is the subset of one choice option among multiple.

Yes/No $\subseteq$ *One choice among multiple*

Different times and the algorithms are as follows:

- A new vote is added in the system,
  The calculation is exactly the say as what described in Yes/No Choice Option, when a new vote is added and algorithms is the same as what mentioned in that part.
- At the end time of poll (at deadline) or when all members have been voted,
  The option(s) which achieves (achieve) the maximum percentage option which is selected by members is the winner of the poll.

4.3.3 Algorithm for calculating result in Multiple Choice Option

Checking the result and its calculation can be done:

- A new vote is added in the system,
  1. Check number of received votes and number of voters for this poll;
  2. If number of received votes is more than half of the number of voters for this poll (i,e > 50%), then verify if any option has received maximum, otherwise exit (as it's immaterial to calculate percentage of votes of each option.)
  3. If received votes is greater than 50%,
  Then two maximum voted options (o1, o2) and number of their vote counts (max1 and max2) are extracted from the system (max1 is greater or equal to max2).
  4. If max1 > max2, then verify if
  max1 > (max2 + remaining number of voters for this poll)
  5. If step-4 is true, then max1 is the absolute winner
  Note: The reason is:
  If
  A=*Remained_Number_of_voters* is the number of voters who has not yet voted for the poll; B=*Number_of_Voters_For_o1* is the number of voters who select o1 option which has maximum number of votes I,e max1

  So, in worst case, if $\forall$**A** voters vote for o2 option, total vote count for option o2 is **A+max2**.
  If **B > A+max2;** means option o1 is still greater than option o2, so in worst case also o1 is the absolute winner.

```
If (Number_of_Received_Votes >( Number_of_Voters / 2))
{           Max1 =Vote_Count_Of_Maximum_Selected_Option;
            Max2=
Vote_Count_Of_Second_Maximum_Selected_Option_Among_Other_Options_Except_Max1;
            If (Max1> Max2)
            {           if (Max1 > (Max2+ Remained_Number_of_voters ) )
                        {           Final_Result= Option_Of_ Max1
                                    announceVotet();
                        }
            }
}
```

- At the end time of poll (at deadline) or when all members have been voted,
  The option(s) which has (have) the maximum selecting percentage among the selected options by the members is (are) the final result/winner of the poll.

4.3.4 Algorithm for calculating result in Priority Option

The result of this kind of voting will be calculated when:

- A new vote is added in the system,
  Assume, priority-1>priority-2>………>priority-n **i,e** priority-1 is of highest and priority-n is of lowest priority.
  Assume, for a poll
  options available are Oi where $\forall i, i \in \{1,2,3\ldots.p\}$
  voters involved in the poll are Vj where $\forall j, j \in \{1,2,3\ldots.q\}$

```
If (Number_of_Received_Votes > (Number_of_Voters / 2))
{
   for each voting option Oi
    {
     If (sum_of_the_selected_priorities_by_users==(q/2)+1)
            {           Final_Result= Oi
                        announceVote();
                        Break;
            }
        }
  }
```

- At the end time of poll (at deadline) or when all members have been voted,
  The option(s) which achieves (achieve) the maximum percentage option which is selected by members is the winner of the poll.

Examples:

There are two ways of proceeding with priority voting.

1. **According to priority of options chosen by user**
   Here we assume initially there is no weight assigned to the voting options.

   Example:
   For Poll-1, four voters are there and three available options are option-1, option-2 and option-3.
   Assume voter-1 has chosen option-2 as highest priority, then option-1 and option-3 as lowest priority.
   Similarly, we assume the other voters have voted and all received votes are portrayed in the following table:

Table 1: Members Vote

	Priority-1	**Priority-2**	**Priority-3**
Voter-1	Option-2	Option-1	Option-3
Voter-2	Option-3	Option-2	Option-1
Voter-3	Option-3	Option-1	Option-2
Voter-4	Option-1	Option-3	Option-2

The received information shown in Table 1, can be represented for each option according to the selected priorities by voters, in Table 2.

Table 2: Priorities derived from Table 1

	Priority of Voter-1	**Priority of Voter-2**	**Priority of Voter-3**	**Priority of Voter-4**
Option-1	2	3	2	1
Option-2	1	2	3	3
Option-3	3	1	1	2

Explanation: in first row, option-1 has received priority of **second** from voter-1**, third** from voter-2, **second** from voter-3 and **first** from voter-4 **(**in Table 2**).**
Similarly, for option-2 and option-3.

So cumulative points for each option from the chosen priority from Table 2 is as below:
Option-1: 2+3+2+1=8
Option-2: 1+2+3+3=9
Option-3: 3+1+1+2=7

We know,
Priority-1 > Priority-2 > Priority-3
That means, **priority-1** is of highest priority and **priority-3** is of lowest priority.
So, less cumulative point is of highest priority and more cumulative point is of lowest priority.
So, option-3 > option-1 >option-2;
<u>So, option-3 is the winner</u>

2. Assign weight among all the voting options beforehand

Assign weight/priority among all the voting options beforehand by the poll creator and this information can be announced along with the poll.
Example: for Poll-1,
If options are option-1, option-2, option-3 and option-4
poll creator can decide
option-1 should be weighted 8 (i,e W1=8)
option-2 should be weighted 6 (i,e W2=6)
option-3 should be weighted 4 (i,e W3=4)
option-4 should be weighted 2 (i,e W4=2)
Now, once the votes have been received and at any point of time if the *Number_of_votes* received by options are:
option-1 gets P number of votes;
option-2 gets Q number of votes;
option-3 gets R number of votes;
option-4 gets S number of votes;

Then value of option-1, V1 = (P*8)/((P*8)+(Q*6)+(R*4)+(S*2))
Similarly, value of option-2, V2= (Q*8)/((P*8)+(Q*6)+(R*4)+(S*2))
And the same way is of V3 and V4.
Now, according the maximum value of Vi where $\forall$i, i$\in${1,2,3,4} winner can be decided and announced.