

CS 6378: Advanced Operating Systems
Section 501
Notes on Voting Protocols

Instructor: Neeraj Mittal

1 Steps in Static Voting Protocol

Let V_i denote the vote assigned to site S_i , and let v denote the total number of votes in the system. Further, let r and w denote the size of read and write quorum, respectively. Note that r , w and v satisfy the following properties:

$$\begin{aligned} r + w &> v \\ w &> \frac{v}{2} \end{aligned}$$

The first inequality ensures that read and write operations cannot execute concurrently. The second inequality ensures that no two write operations can execute concurrently. The two inequalities together guarantee the read-write mutual exclusion property. Now, assume that site S_i wishes to execute an operation (read or write). Let P denotes the set of sites that S_i is able to lock, that is, the set of sites in S_i 's partition. Site S_i computes the following:

$$\begin{aligned} M &= \max\{VN_j \mid S_j \in P\} \\ Q &= \{S_j \in P \mid VN_j = M\} \\ V &= \sum_{S_j \in P} V_j \end{aligned}$$

If the operation is read, then S_i can execute the operation provided $V \geq r$. Likewise, if the operation is write, then S_i can execute the operation provided $V \geq w$. If $S_i \notin Q$ (that is, S_i does not have the most recent version of the object among sites in P), then S_i obtains the most recent version from a site in Q . Once a write operation completes, site S_i updates the variables as follows:

$$VN_i = M + 1$$

Site S_i then propagates the value of VN_i (along with all relevant changes) to all sites in P . If any failure occurs while the operation is in progress, the operation is aborted and S_i simply tries again.

2 Steps in the Hybrid Voting Protocol

Assume that site S_i wishes to execute an update operation. Let P denote the set of sites that S_i is able to lock, that is, the set of sites in S_i 's partition. Site S_i computes the following:

$$M = \max\{VN_j \mid S_j \in P\}$$

$$\begin{aligned}
Q &= \{S_j \in P \mid VN_j = M\} \\
N &= RU_j, \text{ where } S_j \in Q \\
DS &= DS_j, \text{ where } S_j \in Q
\end{aligned}$$

Site S_i 's operation is enabled if one of the following conditions holds:

1. $\text{cardinality}(Q) > N/2$
2. $\text{cardinality}(Q) = N/2$ and $DS \in Q$
3. $N = 3$ and $\text{cardinality}(P \cap DS) \geq 2$

If $S_i \notin Q$, then S_i obtains the most recent version from a site in Q . Once the operation completes, site S_i updates the variables as follows:

$$VN_i = M + 1$$

Variables RU_i and DS_i are updated only if $N \neq 3$ or $\text{cardinality}(P) \geq 3$, and are updated as follows:

$$\begin{aligned}
RU_i &= \text{cardinality}(P) \\
DS_i &= \begin{cases} \text{site with highest identifier in } P & - : (RU_i \neq 3) \wedge (RU_i \text{ is odd}) \\ & : RU_i \text{ is even} \\ P & : RU_i = 3 \end{cases}
\end{aligned}$$

Site S_i then propagates the values of VN_i (along with all relevant changes), RU_i and DS_i to all sites in P .

2.1 Exercise

Suppose there are six sites A, B, C, D, E and F . The linear ordering of the sites is $B > D > A > F > E > C$. The current state of the system is as follows:

	A	B	C	D	E	F
VN	4	3	4	4	2	4
RU	4	5	4	4	6	4
DS	D	-	D	D	B	D

Show the system state after each operation with the following *sequence* of update requests.

- Request by A with partition $\{A, B, F\}$

	A	B	C	D	E	F
VN						
RU						
DS						

- Request by B with partition $\{A, B, D\}$

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
<i>VN</i>						
<i>RU</i>						
<i>DS</i>						

- Request by *E* with partition $\{A, E, F\}$

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
<i>VN</i>						
<i>RU</i>						
<i>DS</i>						

- Request by *E* with partition $\{A, E, D\}$

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
<i>VN</i>						
<i>RU</i>						
<i>DS</i>						

- Request by *A* with partition $\{A, E\}$

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
<i>VN</i>						
<i>RU</i>						
<i>DS</i>						

- Request by *A* with partition $\{A, D\}$

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
<i>VN</i>						
<i>RU</i>						
<i>DS</i>						

- Request by *B* with partition $\{A, B, C, E, F\}$

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
<i>VN</i>						
<i>RU</i>						
<i>DS</i>						

- Request by *C* with partition $\{A, C, D, F\}$

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
<i>VN</i>						
<i>RU</i>						
<i>DS</i>						