Questão 1.

a) Não, pois como não há falha, também não há perda dos dados e o processo segue da mesma forma. Caso houvesse falha, a etapa deveria ser refeita por outro worker e, com isso, prejudicaria a performance.

b) Sim, pois ao salvar no file system outros workers podem ter acesso às informações no caso de falha e dar continuidade normalmente ao processo mesmo sem o worker que falhou.

Questão 2.

Após fazer um MapReduce para contar a quantidade de cada palavra seria feito um novo MapReduce. Na fase de splitting, as palavras seriam separadas, por exemplo, em dois grupos (A-L e M-Z). Na fase de mapping, <chave, valor> seria processado como <palavra, quantidade, total no split>. No reduce, <chave, valor> seria processado como <palavra, quantidade, total completo>. O final seria dado por <palavra, quantidade/total completo>.

Questão 3.

Tomando como base o exemplo da figura. Mesmo não esperando a maioria responder, o “accept-request 5” é enviado para todos, pois a parte que respondeu enviou “promise 5” e o processo teve prosseguimento.

Porém, o “accept 5” pode não acontecer, indicando que não houve consenso da maioria no momento de envio do “promise 5”. Como não se esperou a maioria enviar, aceitou-se o valor 5 até o momento, que pode não ser o consenso da maioria, ocasionado uma falha na aceitação ao final do processo.

Questão 4.

a) Caso o electionTimeout seja muito pequeno e o líder apresente de fato algum problema, economiza-se tempo e realiza-se logo uma nova eleição. Assim, não se espera um tempo desnecessário para identificar um erro no líder, pois isso é confirmado rapidamente.

b) Caso o electionTimeout seja muito pequeno, pode-se assumir erroneamente que o líder apresentou algum problema, dando início a uma nova eleição sem necessidade. Com isso, mais tempo é gasto.

c) Um líder pode virar um follower quando descobre um servidor com um term maior. Após falhar, o líder antigo volta achando que ainda é líder, mas, após receber um sinal do novo líder com term maior, o líder antigo volta a ser um follower.

Questão 5.

a) Não, pois mesmo sendo maioria não é o termo mais atualizado do líder.

b) Não, pois o termo mais atualizado é maior (5>4).

c) Não, pois não é quem tem o log mais completo.

d) Se houver uma falha do líder, outro toma seu lugar. Caso o primeiro líder retorne, mas agora apenas como follower, seu log não é o mais atualizado. Assim, se o commit for feito pela maioria desse antigo líder, haverá problema de consenso, visto que o novo líder possui log mais atualizado, mesmo não sendo maioria para determinado index. Além disso, um novo líder pode ser eleito e mudar esse termo que atualmente é maioria.

Questão 6.

a) “Minerar” significa adicionar um novo bloco à blockchain, o que só pode ser feito quando um minerador resolve um problema matemático, encontrando um número chamado nonce, que será usado para compor a hash do bloco. Essa hash é utilizada para criptografia por meio de um algoritmo chamado SHA-256.

b) No Proof of Work (PoW), diz-se que o bloco é minerado e o criador do bloco é o primeiro que achar a nonce. Além disso, no PoW há uma tendência a centralizar os nós.

No Proof of Stake (PoS), diz-se que o bloco é forjado e o criador do bloco é escolhido de acordo com sua riqueza. Além disso, no PoS há uma tendência a descentralizar os nós.

c) Para alterar um dado, o minerador deve alterar sua hash. Mas, para manter a blockchain coerente, ele deveria mudar também o hash de todos os blocos posteriores, uma vez que os blocos são conectados por seus hash.

Essa ação é desestimulada pois o minerador deveria alterar muitos dados, necessitando deter boa parte do poder da rede (mais da metade).

Nota: 3,7/5,0

1a) Na verdade, a performance pode piorar pq local é mais rápido.  
  
2) Precisava explicar melhor.  
Fala em ""seria feito um novo MapReduce"", mas parece que faz tudo em um só MapReduce.  
Faltou explicar como calcular ""total\_completo"" somaria todos os ""total\_split"", principlamente qdo um Reduce task não pega nada de um dado split.  
  
5b) Pode sim, caso S1 seja eleito.  
  
5c) Na verdade, não precisa ter o log mais completo para ser eleito. Os server votam em algo mais completo que ele mesmo. Nesse caso, S2 teria só dois votos e não ganharia a eleição.  
  
6c) Melhor então usar o poder computacional para minerar e ganhar recompensa, ao invés de ser malicioso.