

Phase 3 Grading Sheet, CSE535, Scott Stoller, Stony Brook U, 2017-09-23	POSSIBLE	ACTUAL
Team:	POINTS	POINTS
TA:		
CLIENT		
ask Olympus whether configuration changed (periodically or as needed)	4	
check that dictionary contains expected content at end of test case	4	
OLYMPUS		
upon reconfiguration-request, send wedge requests (2 pts), validate wedged	24	
messages (2 pts), compute initial running state (incl. replica catch-up) (18 pts),		
create keys and create and setup processes for new replicas (2 pts)		
REPLICA		
head: send reconfiguration-request if timeout waiting for result shuttle	4	
non-head: send reconfiguration-request if timeout waiting for result shuttle	4	
after forwarding request to head		
detect provable misbehavior and send reconfiguration-request	10	
head: periodically initiate checkpoint, send checkpoint shuttle	6	
non-head: add signed checkpoint proof, send updated checkpoint shuttle	6	
handle completed checkpoint shuttle: validate completed checkpoint proof,	8	
delete history prefix, forward completed checkpoint proof		
handle catch-up message, execute operations, send caught-up message	6	
fault-injection: additional triggers for phase 3 (1 pt each)	4	
fault-injection: additional failures for phase 3 (1 pt each)	4	
CONFIGURATION FILES		
support configuration files specified in project.txt (if not, causing inability to test	2	
some functionality with instructor's testcases, also lose 1/3 of points for it)		
LOGS		
detailed and readable logs (If some functionality cannot be adequately verified	6	
due to inadequate logs, deduct suitable points for each affected item above)		
DOCUMENTATION		
README and testing.txt contain all information specified in project.txt	8	
PENALTIES		
bugs/limitations not reported in README (additional 50% off)		
poor code quality (up to -10, depending on severity)		
late submission (5%/day). submission date+time:		
late demo (10%/day). demo date+time:		
TOTAL	100	

For client/Olympus/replica functionality: Aspects not covered by a test case in the submission get at most 80% credit (penalty for inadequate testing). Aspects not covered by a test case during the demo get at most 50% credit, if the code looks reasonable.