Relazione del progetto di laboratorio di Sistemi Operativi

Zakaria Abourida - 950120 - zakaria.abourida@edu.unito.it
Alessio Magnea - 962522 - alessio.magnea@edu.unito.it
Tommaso Remondino - 947410- tommaso.remondino@edu.unito.it

anno accademico 2022/23

Riassunto trama progetto:

Il progetto richiedeva di simulare il traffico di navi cargo per il trasporto di merci di vario tipo, attraverso i porti. Ciascuna nave e porto rappresentano un processo. L'obbiettivo consiste nello sviluppare un metodo per far comunicare i processi tra loro e quindi permettere di completare/svolgere le loro routine.

Comandi utili per la simulazione:

- make all: compilazione di tutti i file necessari per un corretto utilizzo
- make run: viene fatto il linking dei vari file e viene fatta partire la simulazione incominciando dal main (master)
- make clean: serve per cancellare tutti i file ".o"

Scelte implementative aggiunte:

- Per semplicità e comodità abbiamo deciso che metà dei porti genererà domande e l'altra metà genererà le offerte (es. 5 porti = 2 fanno offerte e 3 fanno domanda, 21 porti = 10 fanno offerte e 11 fanno domanda)
- Dal giorno 0 vengono create di giorno in giorno "*SO_FILL / (SO_PORTI / 2) / SO_DAYS*" lotti in offerta.

Riassunto routine dei vari processi:

Master: Genera i processi figli (navi, porti e meteo) e gestisce l'andamento della simulazione assieme al dump.

- Inizializza i segmenti di memoria condivisa, le code dei messaggi ed i semafori necessari
- Legge da file la configurazione dei dati e li inserisce in un segmento di memoria condivisa

- Creare le stringhe contenenti le chiavi d'accesso per i segmenti di memoria condivisa, i semafori e la coda di messaggi... da passare ai processi figli come argomenti
- Genera le offerte giornalmente
- Si occupa dell'invecchiamento delle merci
- Gestisce il timer per il tempo limite della simulazione
- Stampa il dump (ogni secondo)
- Alla terminazione di tutti i figli stampa il dump finale e libera tutte le risorse allocate

Porto: gestisce le banchine e le operazioni di carico/scarico delle navi che attraccano.

- Quando una nave attracca: manda un messaggio alla coda di messaggi del porto.
- Il porto riceve il messaggio e aggiorna il dump in base ai dati presi dal messaggio
- Il messaggio contiene: l'id del porto a cui va consegnato, il tipo della merce e la sua quantità

Nave: compie il grosso delle operazioni e aggiornamenti del dump.

- Compie la ricerca degli annunci (offerta e domanda corrispondenti)
- Fa il controllo scadenza delle merci
- Aggiorna lo stato e le informazioni del dump

Una nota sulla ricerca degli annunci: si tratta di un algoritmo di ricerca iterativo greedy. La ricerca controlla solo che l'offerta e la domanda siano valide e compatibili, non controlla se la merce potrebbe scadere durante il viaggio. Inoltre, seleziona prima le offerte che sono all'inizio dell'array. Questo comporta un picco nel numero di merci che scadono in nave dopo un certo numero di giorni. Un modo per minimizzare lo spreco di merci

scadute in mare potrebbe essere quello di controllare l'array dalla fine (basta scorrere l'array di offerte partendo dall'ultimo elemento al posto che dal primo). Purtroppo questo porterebbe una sostanziale diminuzione delle prestazioni data dal numero di confronti molto elevato che si va a fare, soprattutto nei primi giorni di simulazione, e un aumento delle merci scadute in porto se le navi non riescono a completare le comande abbastanza velocemente.

Meteo: genera casualmente gli eventi che colpiscono le navi (tempeste e maelstrom) ed i porti (mareggiate).

IPC e modelli dati utilizzati e perché:

Struct:

- *Posizione*: utilizzata per conoscere la posizione dei porti e delle navi durante la simulazione (per gli spostamenti...)
- *Merce*: utilizzata per semplificare l'utilizzo del dump merci e non solo. Contiene le informazioni necessarie e richieste delle merci
- *Domanda*: utilizzata per la generazione delle domande e contiene solo le informazioni necessarie
- *Offerta*: utilizzata per la generazione delle offerte e contiene solo le informazioni necessarie
- *Comanda*: utilizzata nel metodo di ricerca degli annunci, usata anche dalle navi per poter salvare i dati della domanda e offerta durante lo scambio
- *Messaggio*: utilizzato dalle navi quando attraccano per indicare quantità e tipo merce da scaricare o caricare
- *StatoNave*: necessario per capire dove si trovano le navi (serve per il meteo, in modo tale che colpisca solo navi in mare e non al porto)
- *Nave*: come sopra citato (contiene StatoNave)

Segmenti di memoria condivisa:

- *Config*: contiene i dati delle configurazioni iniziali prese dal file di configurazione
- *Dump_navi*: contiene i contatori che indicano gli stati delle navi (mare_scarica, mare_carica, porto_carico, porto_scarico) e il numero di navi colpite dalle tempeste e dalle maelstrom
- *Dump_porti*: contiene i contatori che indicano i porti con lo stato delle merci scambiate, le coordinate ed il numero di banchine totali e quelle occupate (porto_x, porto_y, qta_presente, qta_spedita, qta_ricevuta, banchine_tot, banchine_occupate)
- *Dump_merci*: contiene i contatori delle merci scambiate per tipo (tipo_merce, qta_porto, qta_nave, qta_consegnata, scaduta_porto, scaduta_nave)
- *Domande*: "lista" delle merci richieste dai porti suddivise in porti e tipo merci (generate a inizio simulazione)
- *Offerte*: "lista" delle merci offerte dai porti suddivise sempre in porti e tipo merci (generate anche durante la simulazione di giorno in giorno)
- *Registro:* semplice array di dimensione SO_MERCI, utilizzato per salvare la dimensione dei lotti per ciascun tipo di merce (ogni merce ha i lotti di dimensioni diverse)
- *Navi_shm:* array di dimensioni SO_NAVI, serve per tenersi aggiornati sugli stati delle navi
- *Porti_shm:* serve per potersi segnare quali porti sono stati colpiti dalle mareggiate

Semafori:

Un mutex per ogni memoria condivisa che viene scritta e un semaforo per ogni porto. Quelli dei porti hanno valore uguale a quello delle loro banchi-

ne. Quando una nave attracca e occupa una banchina occupa un semaforo, quando se ne va lo rilascia

Coda di messaggi:

Utilizzata esclusivamente per il passaggio di merci da porto a nave e vice versa.

Gestione dei segnali:

I segnali vengono catturati e gestiti attraverso la funzione *"sigaction". E' stato scelto di utilizzare sigaction perché permette di impostare i gestori personalizzati per i segnali. Inoltre presenta una maggiore precisione, controllo e portabilità a confronto di "signal".* Utile anche per il controllo di errore, restituisce un valore intero per permettere di correzione.

I segnali utilizzati che vengono gestiti sono : *SIGINT, SIGTERM, SIGALRM, SIGCHLD, SIGUSR1, SIGUSR2*.

Ciascun processo ha un Signal_Handler() che gestisce i segnali sopra citati in base alle proprie esigenze.

- SIGINT: (ctrl + c) utilizzato per fare terminare subito la simulazione ed i processi
- SIGTERM: terminazione corretta della simulazione e dei processi
- SIGALRM: viene ricevuto dal master quando il timer è terminato e quindi significa che il tempo per la simulazione è finito
- SIGCHLD: segnale mandato dalle navi in caso affondino, serve per avere il conteggio delle navi ancora in circolazione. In caso tutte le siano affondate la simulazione viene terminata.
- SIGUSR1: questo segnale si differenzia in base a chi lo riceve. Se a riceverlo è una nave allora indica che essa è stata appena colpita da una tempesta, altrimenti se a riceverlo è un porto allora indica che esso è appena stato colpito da una mareggiata

- SIGUSR2: questo segnale viene ricevuto solo dalle navi ed indica che essa è stata colpita da un maelstrom e quindi la nave viene affondata assieme al carico...

Comportamenti configurazione simulazioni:

****Le configurazioni sono state eseguite più di 10 volte in modo tale da poter avere un campione adeguato su cui basare delle ipotesi. I numeri di scambi terminati e cominciati sono da considerarsi in media****

DSS(Dense, small ships): La particolarità di questa configurazione è che ci sono 1000 navi, 100 porti ed una sola merce, la quale non può mai scadere perché la scadenza minima e massima sono 50 giorni.

Durante l'esecuzione abbiamo notato che la simulazione termina sempre per causa terminazione tempo (timer is up). Essendoci 1000 navi, molte vengono affondate ma non ci sono abbastanza maelstrom per affondarle tutte. Inoltre la navi sono piccole e quindi non riescono a terminare tutti gli scambi presenti. Nonostante ciò vengono effettuati molti scambi.

Ovviamente si presentano molto spesso i maelstrom, a differenza delle tempeste che appaiono solo raramente.

Nei vari test abbiamo notato che in questa configurazione vengono completati 1563,2 e vengono incominciati 2044 scambi.

AST(as above + trashing): Questa configurazione è identica a quella precedente, con la differenza che questa volta le merci hanno una scadenza minima di 3 giorni ed una massima di 10 giorni. Le considerazioni di questa configurazione sono le stesse di quella precedente, con la differenza che molta più merce scade (sia nei porti che nelle navi). Abbiamo notato che con questa "piccola" variante la media degli scambi completati si è ridotta a 825,9 rispetto ai 2507 incominciati.

BTR (Born to run): Questa è la configurazione più "pesante" in quanto sono presenti 10 navi, 1000 porti e 100 merci. Qui le navi sono molto veloci e molto grandi. Nei test abbiamo notato che ogni 2,5 giorni viene affondata una nave (meno frequenti rispetto alle due configurazioni precedenti). Le mareggiate e le tempeste sono all'incirca lo stesso numero delle configurazioni precedenti. Per quanto riguarda la merce purtroppo molti "lotti" rimangono fermi, però poca quantità di merce tende a scadere durante le simulazioni.

In media vengono completati 51,7 scambi sui 96 incominciati (circa).

**Curiosità: Abbiamo testato la configurazione cambiando il numero delle navi da 10 a 500 per vedere se potesse essere "competitiva" sul numero di scambi completati rispetto alle due configurazione precedenti. La media del numero di scambi completati è 2628. Con un numero inferiore a 400 non si riusciva a superare la media delle altre due configurazioni.

CBS (cargos, big stuff) : Questa configurazione è molto simile a AST solo con quantità molto più ridotte rispetto alle navi, porti e merci.

Si verificano molto raramente eventi meteorologici e purtroppo non vengono mai completati gli scambi, rispetto ai 224 incominciati. Questo perché ci sono pochi scambi e navi: la maggior parte delle volte o la merce scade (porto o nave) oppure una qualche nave viene affondata o rallentata facendola arrivare dopo la scadenza.

Vengono incominciati scambi molto "corposi", pieni di merci, che purtroppo non riescono mai ad arrivare a destinazione.

UC (unlucky cargos): Questa configurazione è particolare. Si distingue dal fatto che vengono generati i maelstrom molto spesso (ogni ora).

La simulazione termina sempre per causa terminazione tempo del timer (timer is up) perché in soli 10 giorni non vengono affondate tutte le navi, tanto meno non vengono terminati gli scambi.

Anche qui come in “CBS” non vengono completati gli scambi (rispetto ai 113 incominciati circa) per gli stessi motivi sopra citati.

Codice preso da fonti esterne

- TEST_ERROR: Questa macro viene utilizzata per stampare gli errori e per semplificare la gestione di essi. Utilizza la variabile ‘errno’ che tiene traccia degli errori che si verificano durante le chiamate di sistema. Presa dagli esercizi mostrati in classe.
- BZERO: macro che inizializza a zero una “porzione” di memoria grande ‘x_size’ partendo dall’indirizzo ‘x’. Utilizza “memset” che imposta ogni byte nell’intervallo di memoria, specificato dai dati sopra citati, a zero. In questo modo garantiamo che l’inizializzazione sia realisticamente sempre a 0 durante la simulazione. Inoltre il fatto di non avere valori garbage che posso essere plausibili ci ha aiutato nel debugging.