TOKEN RING

SINCRONIZZAZIONE con distribuzione di Ticket per mezzo di TOKEN

Si vogliono regolare gli accessi in mutua esclusione ad una risorsa condivisa tra più utenti.
Gli utenti (“Clienti”) sono dislocati su nodi distinti e accedono alla risorsa segnalando la richiesta e seguendo poi un opportuno protocollo di comunicazione.
Ogni nodo del ring è provvisto di un servitore (“Nodo”) che raccoglie le richieste dei processi appartenenti al proprio nodo e le invia ai gestori del ring (“Seccord” e “Cordtok”) registrandole sul token.
Un coordinatore provvede ad avvertire i processi che si sono prenotati quando giunge il loro turno fornendo le informazioni necessarie per accedere alla risorsa(che può essere dotata di opportuni controlli di accesso), contemporaneamente predispone il gestore della risorsa (“Gestore”) ad accettare la richiesta dal cliente di turno
(il “Gestore” vuole simulare una risorsa con un minimo di “intelligenza” software).

SPECIFICHE
La realizzazione deve garantire:

1. Mutua esclusione sulla risorsa
2. Dinamicità

3. Trasparenza
4. Tolleranza ai guasti
5. Scalabilità
6. Efficienza

1. Mutua esclusione

Tale specifica è naturalmente garantita in modo deterministico.

2. Dinamicità

Qualunque nuovo cliente può inviare richieste alla risorsa: è sufficiente che sia allocato su un nodo in cui è presente un gestore e che rispetti il protocollo.
Il sistema è in grado di evolvere consentendo l'inserimento o l'uscita dei nodi partecipanti al ring.
Il gestore del ring si occuperà dell'accettazione di nuovi elementi:
dopo un breve protocollo in cui avviene lo scambio delle necessarie informazioni, indicherà al nuovo arrivato gli endpoint dei nodi tra cui inserirsi (scelti opportunamente in base alla topologia attuale del ring, tenuta costantemente aggiornata in memoria) ed avvertirà i due nodi interessati di attivare il protocollo di inserimento.
Un protocollo non dissimile partirà quando un nodo vuole uscire dal ring: il cliente segnala al coordinatore la sua intenzione sul token e poi il protocollo si riaggancerà a quello appena descritto.

3. Trasparenza

I "Clienti" non sono tenuti ad avere una conoscenza pregressa della allocazione della risorsa o del suo stato; si limitano ad informare il gestore del proprio nodo della necessità di usare la risorsa ed attendono poi di essere contattati quando giunge il loro turno.
Non sono nemmeno a conoscenza della presenza di un token né ovviamente si occupano della sua gestione.
Non risentono nemmeno di eventuali crash momentanei della rete.
I "Nodi" sono tenuti a conoscere solo i 2 vicini (o meglio ancora solo il proprio successore) benché il protocollo non precluda la possibilità di dotare della conoscenza dell'intero ring anche i singoli nodi.
Il coordinatore conosce la struttura dell'intero ring.

4. Tolleranza ai guasti

Siamo nell'ipotesi di guasto singolo.
In tal caso è possibile accorgersi rapidamente di eventuali cadute dei gestori dei nodi controllando lo stato della connessione (TCP) usata per il passaggio del token o più in generale osservando ritardi eccessivi nel passaggio del token.
In ogni caso il nodo che rileva l'anomalia invia un token di servizio informando il coordinatore che gestisce la situazione inviando ai nodi interessati le informazioni necessarie per effettuare il recovery.
Non occorrono segnalazioni broadcast (si utilizza il token di emergenza) né protocolli di elezione del token (l'ultimo token ricevuto resta in memoria di ciascun nodo finché non ne arriva un altro normale, cosa possibile solo se nel ring non ci sono guasti).
La caduta di un nodo viene segnalata dal nodo successivo a quello caduto.
Tale segnalazione potrebbe essere erronea e va pertanto considerato solo come una segnalazione di "possibili" malfunzionamenti, in base alla quale il coordinatore può effettuare dei controlli sulla reale situazione.
"Nodi" che per un qualunque motivo sono dichiarati caduti pur non essendolo, vengono esclusi dal ring e sono costretti ad effettuare di nuovo la richiesta di inserimento.

E' previsto che il coordinatore sia replicato in due soggetti al fine di tollerare il guasto singolo e partizionato nei suoi compiti al fine di bilanciare meglio il carico, con la possibilità di assumere ciascuno il compito dell'altro in caso di caduta.
Nell'implementazione pratica il gestore risulta solo partizionato, benché concettualmente il protocollo non impedisca di implementare la totale replicazione.

5. Scalabilità

La scalabilità è limitata dalla durata della circolazione del token nel ring.
Bisogna altresì notare che un aumento del numero di clienti fruitori della risorsa non induce un degrado delle prestazione se non in termini di attesa del proprio turno, problema questo legato alla natura seriale della risorsa e non ai procedimenti utilizzati per garantire la sincronizzazione.

6. Efficienza

Overhead:
Non vengono usati messaggi broadcast o multicast e più in generale la quantità di traffico in messaggi è in ogni momento molto limitata (passaggio del token e risveglio dei clienti).
Ogni processa utilizza una sola porta per le comunicazioni fuori dal nodo. All'interno della stessa località si fa per lo più ricorso a canali IPC.
In termini di risorse i gestori devono mantenere in memoria solo la struttura del ring e la lista delle richieste pendenti; i "Nodi" devono solo tenere l'ultimo token arrivato.

A questo punto risulta inevitabile un confronto con il Token Passing in cui il token è utilizzato direttamente dai clienti per l'accesso alla risorsa.

Tempi di risposta:
Cominciamo con il dire che, soprattutto in caso di scarso traffico, la nostra soluzione presenta tempi di risposta superiori, in quanto all'arrivo non si può accedere subito alla risorsa ma bisogna attendere che il token torni ai coordinatori del ring (=> dipende dalla dimensione del ring) e che questi risveglino il processo in corso (dopo aver effettuato la connessione).
La situazione cambia in caso di alto traffico in cui la coda delle richieste nei coordinatori è piena. In tal caso il coordinatore può subito iniziare il servizio del turno successivo, mentre in generale nel Token Passing si deve attendere che il token giunga ad un cliente che ne abbia bisogno.
Siamo nelle ipotesi di alto traffico quindi è probabile che il token dopo pochi passi finisca ad un processo che lo utilizzerà, quindi la differenza di prestazioni tra le due soluzioni sarà legata al tempo necessario al coordinatore per connettersi e svegliare un cliente.
A tal proposito è possibile introdurre un'evoluzione: duplicando la parte dei processi del gestore del ring deputata al risveglio dei clienti, è possibile stabilire il collegamento con il cliente del turno successivo prima che l'attuale servizio sia completato.

Scalabilità:
La scalabilità è limitata dalle dimensioni del ring, in quanto ciò determina il tempo necessario per il completamento di un giro da parte del token e quindi per poter servire le richieste.
D'altra parte questa limitazione è legata al numero di nodi non al numero di fruitori effettivi della risorsa presenti nel ring, come avviene invece nel Token Passing, su ogni nodo potrebbe teoricamente esserci un numero qualunque di clienti.
Consideriamo poi che la struttura potrebbe essere modificata introducendo il concetto di sottoring: ogni coordinatore raccoglie le richieste del suo sottoring e facendo poi parte del superring attende un supertoken in cui introdurre in blocco le richieste ricevute, gestite poi da un supercoordinatore.
In ogni caso è possibile ottenere un numero di utenti che utilizzano la risorsa senza degrado di prestazioni, presumibilmente superiore rispetto al Token Passing.

Gestione dei crash:
La nostra soluzione presenta il limite di un coordinatore centralizzante la cui caduta implica il blocco totale del servizio, ma l'ipotesi di guasto singolo insieme alla replicazione del coordinatore ci consente di eliminare questo handicap.
Notiamo invece che la rilevazione di guasti negli elementi del ring avviene in tempi più brevi, in quanto la durata massima per un giro del ring è predicibile con maggiore precisione dipendendo esclusivamente dal numero degli elementi del ring; le operazioni che devono effettuare i "Nodi" prima di passare nel ring sono infatti molto più veloci rispetto ai tempi necessari per passare il token o per accedere alla risorsa.
Un crash di un cliente comporterà del tempo per la rilevazione da parte del coordinatore del ring, ma non influirà in alcun modo sulla gestione del token.
Il token non viene mai perso e non occorrono protocolli di elezione.

Fairness:
Se si vuole garantire solo l'accesso in mutua esclusione e non l'ordine di prenotazione, si possono implementare politiche diverse nell'ordine di servizio delle richieste. I coordinatori infatti hanno l'elenco delle richieste raccolte in un intero giro possono dunque scegliere liberamente l'ordine con cui eseguirle(es: si può favorire chi ha fatto meno richieste o richieste meno onerose, oppure si può semplicemente invertire l'ordine di servizio dei nodi: (1,2...N) (N,N-1,..,1) (1,2...N)ecc. oppure ruotarlo (1,2,..,N) (2,3,..N,1) ecc. eliminando in tal modo l'ordine di precedenza che si verrebbe naturalmente a creare nella struttura a ring).
Limitarsi ad attendere per un certo tempo le richieste o attenderne un certo numero prima di iniziare a servirle non sortisce lo stesso effetto perchè finisce con il privilegiare i clienti più veloci o in qualche modo favoriti nell'accesso alla risorsa.

Trasparenza:
I clienti non sono tenuti a conoscere direttamente la risorsa e non devono gestire direttamente il token e i problemi della rete.

Nel caso in cui si abbia a disposizione un pool di risorse anzichè una singola risorsa, avere a disposizione tutte le richieste consente ai coordinatori di ridistribuire al meglio il carico sulle varie risorse.

INTRODUZIONE

L'idea che ha originato questo lavoro e' stata quella di realizzare una gestione di risorsa generica condivisa da piu' utenti e controllata da un gestore. In termini generali si tratta di effettuare la sequenzializzazione di un problema che per sua natura si presenta parallelo.
L'assunto di base e' che una molteplicita' di clienti dislocati su nodi diversi possano accedere alla risorsa semplicemente inviando una richiesta seguendo un opportuno protocollo. Ogni singolo utente lavora in modo asincrono, manifestando le proprie esigenze quando vuole, indipendentemente dallo stato della risorsa in quell'istante.
E' la struttura a ring del sistema, gestita dai processi coordinatori, ad occuparsi della manipolazione dell'informazioni in modo da effettuare le dovute coordinazioni
Siamo coscienti che il contesto ingegneristico dove prende forma questa relazione detta implicitamente degli standard di comunicazione dai quali le questioni non strettamente "tecniche" rimangono escluse, tuttavia affrontando il progetto di questo "sistema" si sono presentate dinamiche molto interessanti che, passando per strade alternative alla fredda razionalita' ingegneristica, vorrebbero ora venire alla luce.
Il dover coordinare delle "parti" per soddisfare le esigenze pratiche di un "tutto" porta naturalmente l'occhio attento oltre i confini delle macchine. Se abbiamo la fortuna di ricordare ancora che anche noi, come essere umani, siamo parte di un tutto allora ci apparira' chiaro come le regole che coordinano un sistema di processi non siano poi troppo diverse da quelle che governano una societa' di uomini. In entrambi i contesti emergono, ad esempio, delle relazioni gerarchiche che separano le parti. Si distinguono i consapevoli dai non consapevoli , i controllori dai controllati, i produttori dai consumatori e cosi via. Risulta tracciata quindi una linea di demarcazione netta che, quasi a voler dividere il cielo dalla terra, separa chi svolge un ruolo attivo da chi invece e' passivo.
Ma la maggiore attivita' la si incontra dove c'e' la conoscenza dell'intero sistema, dove vive la consapevolezza di quello che accade dentro e fuori ogni processo o uomo che dir si voglia. Questo pero' non e' accessibile a tutti, non lo e', ad esempio, per una singola entità che opera su un nodo, a meno che non si modifichi innalzandosi al livello dei coordinatori ma questo e' naturalmente impedito dalle specifiche con il quale e' progettato il sistema che tendono sempre e comunque a conservare.
La conclusione a questa introduzione porta con se' la riflessione sul significato della parola liberta' oggi, in un mondo dove gli uomini possono essere paragonati a processi e dove i bisogni individuali sono gestiti da coordinatori che hanno il compito di far rispettare le specifiche del sistema.

PROCESSI DEL SISTEMA

I processi distinti che risultano per una simile implementazione sono sette.
Tra questi se ne distinguono tre che hanno il compito di realizzare fisicamente le connessioni

per la circolazione del token. I primi due fanno capo rispettivamente ai programmi coordinatori "Cordtok" e "Seccord" e realizzano, nelle condizioni di ring vuoto, il ring elementare. Il terzo processo "Nodo" e' il prototipo di un gestore di un nodo geografico che si collega al ring. Permette l'interazione dei processi "Cliente" con il processo "Gestore" pur non avendo visibilita' completa del sistema. Per completezza citiamo anche gli ultimi due che mancano all'appello. Fanno capo anche loro ai due coordinatori "Cordtok" e "Seccord" e si occupano rispettivamente di gestire le richieste di inserimento nel ring da parte di nuovi utilizzatori di risorsa e le comunicazioni con il gestore della stessa.

FUNZIONAMENTO DEL SISTEMA

Nella pratica un nodo geografico nel quale si potrebbero manifestare esigenze di utilizzo della risorsa remota deve entrare a far parte del ring tramite una preliminare fase di negoziazione con l'opportuno processo del coordinatore (Cordtok). In base alla posizione del nuovo nodo, determinata dal suo endpoint, il coordinatore decide la posizione più opportuna dove inserirlo nel ring. Una volta conclusa tale fase puo' avvenire il collegamento alla struttura scalabile per poter gestire il token delle prenotazioni. Tale token ha il compito di raccogliere nodo per nodo i bisogni di utilizzo della risorsa che i vari clienti sparsi nella rete hanno manifestato. Tali bisogni sono simulati dal processo "Cliente".
Sono previste sessioni di prenotazione risorsa, naturalmente, ma anche di terminazione e uscita volontaria dal ring. Queste ultime due mettono in moto i meccanismi piu' complessi del sistema che sono quelli che realizzano la dinamicita' e la tolleranza ai guasti.

L'ultimo processo attivato dal sistema, forse il piu' importante, e' il gestore della risorsa. E' naturalmente quello piu' a stretto contatto con essa. Comunicando con il coordinatore (Seccord) che conosce le prenotazioni in sospeso, ha il compito di interagire poi con i clienti in ordinata attesa di veder soddisfatti i propri bisogni di utilizzo della risorsa.

Queste poche righe non hanno la presunzione di esaurire gli aspetti che un tale progetto porta naturalmente con sè. Per un osservatore esterno potrebbero forse anche essere sufficienti per chiarire le dinamiche generali che si muovono nelle molte linee di codice. Per un osservatore piu' curioso e per chi ha realizzato il progetto pero' possono solo rappresentare la punta di un iceberg che nasconde la sua dimensione sotto il livello dell'acqua oltre il quale l'occhio superficiale non arriva. Nell'idea che sia cosa buona non essere superficiali aiutiamo ora lo sguardo ad acquisire profondita'.

OBBIETTIVI E SPECIFICHE DEL SISTEMA

*Trasparenza nella comunicazione.

*Dinamicita'.

*Scalabilita'

*Efficienza del sistema.

*Tolleranza ai guasti.

*Eterogeneita'

TRASPARENZA NELLA COMUNICAZIONE

Nell'idea che si puo' associare al concetto di trasparenza della comunicazione si muovono molte definizioni.
Riconducendoci alle problematiche di comunicazione in rete si potrebbe asserire che si tratta della realizzazione di un meccanismo che svincoli il cliente generico dalla conoscenza dei dettagli realizzativi del sistema ovunque esso si trovi nella rete. Sara' cosi' possibile, per quest'ultimo, esprimere le proprie richieste in modo semplice e spettera' successivamente al sistema il compito di manipolare le informazioni in modo da poterle gestire al suo interno. Tale cliente ha consapevolezza solamente di dover porre le proprie richieste in modo che siano comprensibili al processo che opera sul nodo. Cio' segue naturalmente l'impostazione del progetto nel suddividere le responsabilita' implementative tra i coordinatori e i controllori di nodi.
Si instaura a questo punto una gerarchia di comunicazione e di controllo che vede i processi gestori delle risorse utilizzare le informazioni ottenute dai nodi, i quali a loro volta le hanno ricevute dai clienti. Le capacita' di controllo sono parimenti distribuite. I "clienti" non hanno

nessuna voce in capitolo mentre per i processi che governano i nodi e' prevista la possibilita' di inserire delle "lamentele" nel token se ad esempio il token tarda ad arrivare per un problema sul nodo precedente. Queste "lamentele" una volta giunte ai coordinatori fanno scattare i meccanismi di recupero che si incaricano di mantenere il funzionamento dell'intero sistema.

DINAMICITA'

Per introdurre questo aspetto del sistema potremmo ricordare che la dinamicità e' un impostazione progettuale mirata ad evitare delle rigidità strutturali. Flessibilità e migliore utilizzo delle risorse sono due forti attrattive per le direzioni di sviluppo di qualsiasi sistema, non solo informatico.
Ricordandoci le esigenze didattiche limitiamo l'attenzione all'ambito del token ring in questione. In tal caso si tratta, praticamente, di rispondere alla domanda se l'intelligenza che opera tra le linee di codice e' votata alla furbizia oppure no.
Possiamo subito dire che nel ruolo di studenti che progettano e realizzano un esercitazione pratica in un "limitato" arco di tempo abbiamo fatto del nostro meglio per far rivivere le astuzie del mondo teorico nel mondo pratico. Notevoli difficoltà si sono presentate lungo il cammino, sia per nostra inesperienza ma anche e non in ultimo per la constatazione, con ingenuo stupore, che le cose non vanno sempre come si crede debbano andare basandosi sulla teoria.
Alla fine dalla buona volontà controbilanciata dalle difficoltà ha preso forma un idea di sistema che realizza un compromesso con le dinamicità teoriche possibili.
La rigidità ha prevalso sulla flessibilità per ciò che riguarda il controllo centrale. L'idea iniziale era di avere due programmi coordinatori, "Cordtok" e "Seccord", completamente interscambiabili, soprattutto per esigenze di tollerare i guasti dei coordinatori, uno pronto ad intervenire per sostituire l'altro impossibilitato a mantenere il controllo del sistema. Per le difficoltà incontrate nella ripartizioni dei compiti tra i meccanismi realizzativi delle funzioni di controllo, siamo stati costretti a decentralizzare su due programmi quello che invece doveva essere centralizzato in uno e duplicato nell'altro.
Allontanandosi per un istante dal contesto informatico questo non stupisce affatto se si pensa a quanti e quali compromessi vivono implicitamente o esplicitamente dove si manifestano esigenze di controllo. Su scala ridotta potrebbe essere metafora di un monito per noi tecnici che nella nostra infinita idea di controllo e razionalità rischiamo di dimenticare che qualcosa scivolerà sempre via dalla rete della ragione, e forse proprio le questioni più importanti.
Tornando comunque con i piedi per terra vediamo dove invece la dinamicità e' riuscita a trovare espressione.
Per incontrarla dobbiamo scendere di un livello nella gerarchia del sistema e fermare l'ascensore sul piano dei nodi geografici che costituiscono il ring. E' dopo aver varcato il confine di demarcazione tra controllori e controllati che si può trovare la flessibilità in opera. I vari nodi con il loro entrare ed uscire dal sistema a seconda di volontà o necessità realizzano un movimento che tiene impegnate le risorse del controllo e al tempo stesso conferiscono un senso di maggiore attività a tutto il sistema nel suo complesso.

Basandosi sul, quasi sempre, piacevole impegno occorso per realizzare questi meccanismi e sulla constatazione che sono proprio quest'ultimi a conferire maggiore interesse e, perché no, anche fascino al progetto possiamo scrivere che forse e' nella dinamicità e nell'evitare rigide strutture chiuse che si nascondono tante verità per ognuno di noi.
D'altronde ce lo dice anche la natura che una goccia di acqua corrente diventa cosa morta se trattenuta nel pugno della mano.

SCALABILITA'

Intesa come capacità del sistema di funzionare in modo corretto indipendentemente dalle dimensioni dello stesso la scalabilita' esprime un significato molto ampio. Nel nostro ristretto campo d'azione andra' intesa come la capacita' di effettuare un numero crescente di comunicazioni in un tempo che rimane comunque sempre entro limiti tollerabili e controllabili. Andando a restringere questa idea all'ambito del progetto in questione possiamo subito dire che il concetto di tempo perde molto del suo significato originario. Essendo previsti solo a livello teorico, dei dispositivi di temporizzazione per le comunicazioni queste non sono vincolate ad avvenire entro limiti prestabiliti. Da un certo punto di vista, troppo ottimistico ce ne rendiamo conto, si potrebbe ipotizzare una dimensione qualsiasi per il sistema, a patto pero' di saper aspettare ognuno con pazienza il proprio turno senza essere prede dell'ansia. In linea di principio cio' potrebbe realizzarsi poiche' dove si e' ritenuto opportuno abbiamo creato strutture dinamiche per svincolarsi il piu' possibile da dimensioni prestabilite, come ad esempio nella lista circolare che memorizza la struttura del ring oppure nella coda delle richieste in attesa.

EFFICIENZA DEL SISTEMA

Anche per parlare di efficienza entrano in ballo il concetto di compromesso.
Dove è stato possibile è stata tenuta ben presente l'idea dell'efficienza. La quantità di traffico di messaggi è stata mantenuta al minimo indispensabile evitando implementazioni ridondanti. Non vengono usati messaggi multicast ,ad esempio, ma solo circolazione di token. Nei processi "Nodo", inoltre, è stato possibile utilizzare una singola porta per tutte le comunicazioni avendo avuto cura di utilizzare i meccanismi interni alla primitiva "select". Sempre in riferimento alle località gestite sui nodi, va detto che al loro interno le comunicazioni tra i processi "Clienti" e il gestore del "Nodo" avvengono tramite canali IPC, risultando quest'ultimi di minore impatto sul sistema rispetto alle socket.
Nel valutare "l'impatto ambientale" di questo progetto sulle risorse delle macchine dove si e' sviluppato abbiamo qualcosa su cui riflettere. Ci siamo resi conto che dove questo impatto è stato limitato ciò è successo grazie alla nostra dedizione al lavoro andando però a discapito del tempo utilizzato.
Nel dover procedere rapidamente verso un obbiettivo ben preciso in un tempo limitato ci siamo resi conto di come si sia tentati pericolosamente a fare tutto in fretta. Inizialmente si potrebbe

anche dire che questa e' una nota positiva; ma se ci si ferma a riflettere, pero', a scapito di cosa si ottiene questa rapidita' allora non ci si rallegra nel constatare che spesso e' il sistema con le sue risorse, o memorie, limitate a pagare il conto per la fretta. Essendo il tempo ormai sempre piu' determinante ci rammarichiamo nel constatare come l'eleganza dell'evitare sprechi di risorse venga spesso messa in secondo piano. Per noi studenti sono state le esigenze di dover limitare la gia' molto consistente quantita' di tempo impiegata per questo progetto, a scapito di altri esami, a farci preferire in alcuni casi la rapidita'.
Su altra scala ci potra' essere l'esigenza di arrivare sul mercato prima della concorrenza. Ogni piano della realta' ha il suo perche' che legittima il procedere sempre piu' rapidamente.

Ma... Se ci si ferma a riprendere il fiato e si volge lo sguardo un po' oltre il proprio naso dove i voti su un libretto universitario o le file di zeri su un bilancio economico perdono il loro illusorio fascino, allora ci si potra' anche interrogare su dove ci stia portando questa fretta. Percorriamo veramente la strada del "progresso" che porta in alto o dobbiamo temere che dietro ogni curva possa nascondersi il vuoto.
Noi un domani saremo chiamati Ingeneri e gia' oggi ne sentiamo tutta la responsabilità, perche' per chi non crede alla favola della strada che porta in alto e prospetta un futuro disincantato e pieno di difficolta', emergono oggi fino alla soglia della coscienza molti imperativi morali. Entrando a far parte delle attivita' produttive potremmo diventare, nel nostro piccolo, artefici del destino del mondo che abitiamo. Se rimarremo intrappolati nella gabbia che ci stiamo costruendo intorno senza riuscire a trascendere i confini degli interessi personali a favore degli interessi collettivi, anche di chi verra' dopo di noi, allora avremo fallito sia come uomini che come ingegneri. Non sara' il conto in banca a darci la misura della nostra efficienza all'interno del sistema ma sarà la qualità dell'aria che respireremo o del cibo che mangeremo a farci riflettere su quello che abbiamo creato e su che uomini siamo diventati.

TOLLERANZA AI GUASTI

Forse ci siamo distratti, come studenti di Ingegneria dovremmo concentrare un po' di piu' l'attenzione su questioni tecniche, d'altronde questa e' pur sempre una relazione propedeutica alla discussione finale sul corso di "Reti di Calcolatori" con Antonio Corradi(prof.). Ne siamo consapevoli. Tuttavia pero' riteniamo sia anche giusto cercare di intendere i propri studi in un contesto piu' ampio che vada oltre le alte mura dell'universita'. Abbiamo vissuto anni a rincorrere teoremi e dimostrazioni, le lezioni ci hanno insegnato, e' vero, a ragionare ma non e' piu' sufficiente. Oggi piu' che mai dobbiamo sviluppare la capacita' di guardarci intorno e di allargare i nostri orizzonti per riuscire a vedere le implicazioni pratiche dei nostri studi fin troppo teorici. Un semplice ingegnere e' ancora molto lontano dall' essere anche Uomo, ma bisogna essere prima di tutto uomini per parlare seriamente di tolleranza ai guasti.
Per iniziare potremmo introdurre questa signora come la capacita' del sistema di assorbire gli effetti collaterali di alcune anomalie.

Da quanto espresso in precedenza e se si condivide l'accostamento uomo-processo dovrebbe apparire ora chiaro che descrivendo i nostri meccanismi d'azione in caso di malfunzionamenti stiamo in realta' parlando anche di altro.
Nel sistema sviluppato abbiamo previsto la gestione di un solo tipo di guasto riferito alla caduta di un nodo che rischia di interrompere il ring, in tal caso il nodo caduto viene escluso a seguito di una fase di coordinamento tra gli altri ancora attivi. Quando chi deve ricevere il token si accorge che la connessione con il nodo precedente non e' piu' attiva inoltra un token di emergenza con la segnalazione del nodo interessato al guasto, sara' poi il coordinatore centrale a immettere in rete le direttive, sotto forma di due endpoint, per escludere il nodo e mantenere il sistema.
Ce ne rendiamo conto e' un modo poco elegante di procedere. Si poteva tentare, ad esempio, un contatto con il nodo in difficolta', magari predisponendo un canale per le comunicazioni urgenti, per capire i problemi e proporre eventuali soluzioni. Tanto per dare una seconda possibilita' prima di rinunciare alla connessione; ma non c'e' stato il tempo. La tolleranza ai guasti che e' stata presentata come una signora ha in realta' la falce in mano perche' mantiene il sistema ma non recupera il nodo ne' le richieste dei "clienti". Questo purtroppo e' quello che succede quando non si trova piu' il tempo per fare le cose con calma.

ETEROGENEITA'

Intenderemo il senso della parola come la capacita' di far comunicare macchine diverse tra loro. Per quanto ci riguarda e' stato fatto l'indispensabile comprendendo nel nostro raggio d'azione solamente le macchine che sono in grado di porre le loro informazioni nella forma standard di rete. Le funzioni di conversione dei dati dal formato interno a quello esterno hanno reso possibile tale compatibilita' se pur superficiale. Rimangono infatti escluse dal sistema quelle macchine che non riescono ad aderire al suddetto Standard.
E' interessante notare ancora una volta come questo nostro modo di procedere rifletta su scala ridotta delle dinamiche che in un altro contesto assomigliano a quelle del mondo globalizzato inspirato dai parametri dei piu' potenti.
Non e' un caso poi se anche noi come singoli individui, parte di un "tutto" sempre più preoccupante, "giudichiamo" i nostri simili aderendo a dei pregiudizi; una macchia d'olio si sta spandendo, ormai non piu' lentamente, e sembra quasi volerci condurre a percepire il mondo a scacchi, sezionato, ancora una volta, da Standard e pregiudizi.
Il sempre crescente bisogno di integrazione ed il buon senso invitano invece a non cadere nella trappola di porsi nella posizione di chi detiene la verita' e che giudica sempre e solo in base a sè stesso.

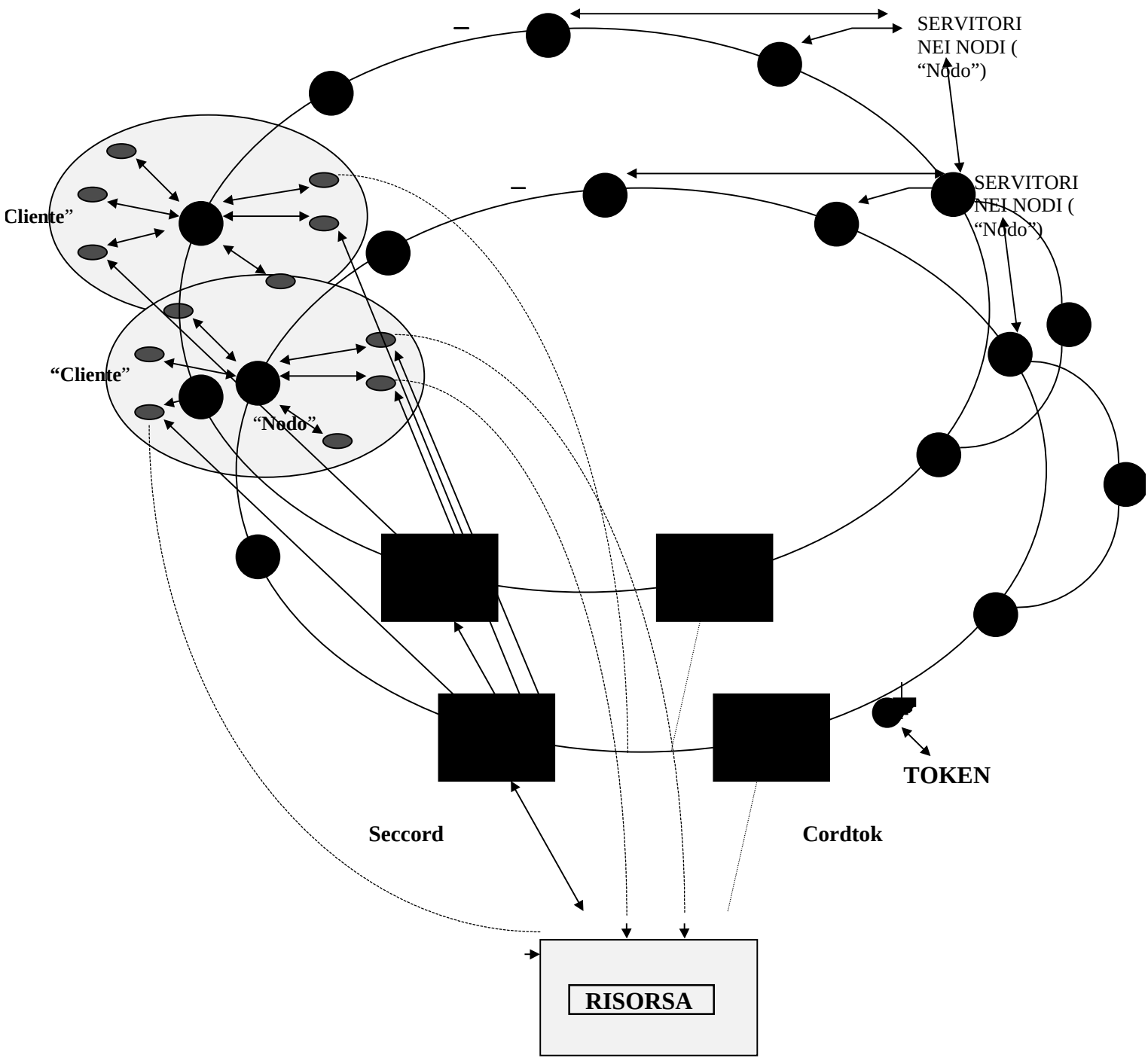
CONCLUSIONE

Avremmo potuto parlare un po' di più di tcp/ip e socket ma confidando nella loro sensibilità e soprattutto nella fiducia riposta nel docente abbiamo modo di sperare che tali argomenti non si siano troppo dispiaciuti per aver dovuto far posto ad altro.
Il nostro e' stato solo un tentativo di comunicare qualcosa che non rientrava nei soliti Standard ma che comunque, sentivamo, andava detto. L'università ci ha istruito su cosa si Deve dire in questi contesti, ma non su quello che si Può dire, e a noi, studenti "fuori" corso, piace pensare che, nel rispetto, si possa ancora comunicare quello che si sente.

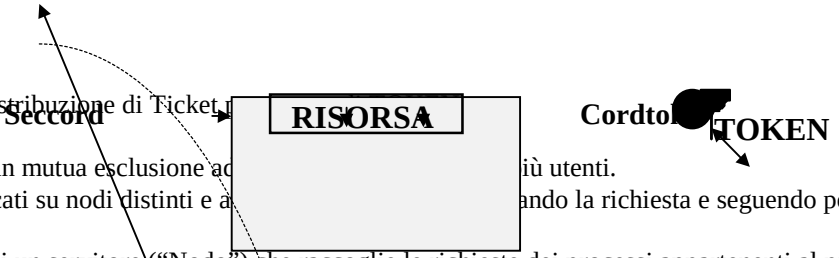
Forse questo scritto sarà un messaggio in una bottiglia che navigherà nel mare che l'informatica ha creato e come tale potrà raggiungere persone anche molto lontane, e' insieme a tutti quelli che realizzeranno questo contatto che si può sperare in un futuro sostenibile dalle economie e dagli uomini.

Prima di salutare, sperando di non essere caduti prede di ingenui entusiasmi di gioventù, ringraziamo per la pazienza e la disponibilità dimostrata fino all'ultima riga.

Spinarelli Mauro
Zappasodi Daniele



SINGRONIZZAZIONE con distribuzione di Ticket



Si vogliono regolare gli accessi in mutua esclusione ad più utenti.
Gli utenti (“Clienti”) sono dislocati su nodi distinti e a ando la richiesta e seguendo poi un opportuno protocollo di comunicazione.
Ogni nodo del ring è provvisto di un servitore (“Nodo”) che raccoglie le richieste dei processi appartenenti al proprio nodo e le invia ai gestori del ring (“Seccord” e “Cordtok”) registrandole sul token.
Un coordinatore provvede ad avvertire i processi che si sono prenotati quando giunge il loro turno fornendo le informazioni necessarie per accedere alla risorsa(che può essere dotata di opportuni controlli di accesso), contemporaneamente predispone il gestore della risorsa (“Gestore”) ad accettare la richiesta dal cliente di turno
(il “Gestore” vuole simulare una risorsa con un minimo di “intelligenza” software).

SPECIFICHE
La realizzazione deve garantire:

1. Mutua esclusione sulla risorsa
2. Dinamicità
3. Trasparenza
4. Tolleranza ai guasti
5. Scalabilità
6. Efficienza

1. Mutua esclusione

Tale specifica è naturalmente garantita in modo deterministico.

2. Dinamicità

Qualunque nuovo cliente può inviare richieste alla risorsa: è sufficiente che sia allocato su un nodo in cui è presente un gestore e che rispetti il protocollo.
Il sistema è in grado di evolvere consentendo l'inserimento o l'uscita dei nodi partecipanti al ring.
Il gestore del ring si occuperà dell'accettazione di nuovi elementi:
dopo un breve protocollo in cui avviene lo scambio delle necessarie informazioni, indicherà al nuovo arrivato gli endpoint dei nodi tra cui inserirsi (scelti opportunamente in base alla topologia attuale del ring, tenuta costantemente aggiornata in memoria) ed avvertirà i due nodi interessati di attivare il protocollo di inserimento.
Un protocollo non dissimile partirà quando un nodo vuole uscire dal ring: il cliente segnala al coordinatore la sua intenzione sul token e poi il protocollo si riaggancerà a quello appena descritto.

3. Trasparenza

I "Clienti" non sono tenuti ad avere una conoscenza pregressa della allocazione della risorsa o del suo stato; si limitano ad informare il gestore del proprio nodo della necessità di usare la risorsa ed attendono poi di essere contattati quando giunge il loro turno.
Non sono nemmeno a conoscenza della presenza di un token né ovviamente si occupano della sua gestione.
Non risentono nemmeno di eventuali crash momentanei della rete.
I "Nodi" sono tenuti a conoscere solo i 2 vicini (o meglio ancora solo il proprio successore) benché il protocollo non precluda la possibilità di dotare della conoscenza dell'intero ring anche i singoli nodi.
Il coordinatore conosce la struttura dell'intero ring.

4. Tolleranza ai guasti

Siamo nell'ipotesi di guasto singolo.
In tal caso è possibile accorgersi rapidamente di eventuali cadute dei gestori dei nodi controllando lo stato della connessione (TCP) usata per il passaggio del token o più in generale osservando ritardi eccessivi nel passaggio del token.
In ogni caso il nodo che rileva l'anomalia invia un token di servizio informando il coordinatore che gestisce la situazione inviando ai nodi interessati le informazioni necessarie per effettuare il recovery.
Non occorrono segnalazioni broadcast (si utilizza il token di emergenza) né protocolli di elezione del token (l'ultimo token ricevuto resta in memoria di ciascun nodo finché non ne arriva un altro normale, cosa possibile solo se nel ring non ci sono guasti).
La caduta di un nodo viene segnalata dal nodo successivo a quello caduto.
Tale segnalazione potrebbe essere erronea e va pertanto considerato solo come una segnalazione di "possibili" malfunzionamenti, in base alla quale il coordinatore può effettuare dei controlli sulla reale situazione.
"Nodi" che per un qualunque motivo sono dichiarati caduti pur non essendolo, vengono esclusi dal ring e sono costretti ad effettuare di nuovo la richiesta di inserimento.

E' previsto che il coordinatore sia replicato in due soggetti al fine di tollerare il guasto singolo e partizionato nei suoi compiti al fine di bilanciare meglio il carico, con la possibilità di assumere ciascuno il compito dell'altro in caso di caduta.
Nell'implementazione pratica il gestore risulta solo partizionato, benché concettualmente il protocollo non impedisca di implementare la totale replicazione.

5. Scalabilità

La scalabilità è limitata dalla durata della circolazione del token nel ring.

Bisogna altresì notare che un aumento del numero di clienti fruitori della risorsa non induce un degrado delle prestazione se non in termini di attesa del proprio turno, problema questo legato alla natura seriale della risorsa e non ai procedimenti utilizzati per garantire la sincronizzazione.

6. Efficienza

Overhead:
Non vengono usati messaggi broadcast o multicast e più in generale la quantità di traffico in messaggi è in ogni momento molto limitata (passaggio del token e risveglio dei clienti).
Ogni processa utilizza una sola porta per le comunicazioni fuori dal nodo. All'interno della stessa località si fa per lo più ricorso a canali IPC.
In termini di risorse i gestori devono mantenere in memoria solo la struttura del ring e la lista delle richieste pendenti; i "Nodi" devono solo tenere l'ultimo token arrivato.

A questo punto risulta inevitabile un confronto con il Token Passing in cui il token è utilizzato direttamente dai clienti per l'accesso alla risorsa.

Tempi di risposta:
Cominciamo con il dire che, soprattutto in caso di scarso traffico, la nostra soluzione presenta tempi di risposta superiori, in quanto all'arrivo non si può accedere subito alla risorsa ma bisogna attendere che il token torni ai coordinatori del ring (=> dipende dalla dimensione del ring) e che questi risveglino il processo in corso (dopo aver effettuato la connessione).
La situazione cambia in caso di alto traffico in cui la coda delle richieste nei coordinatori è piena. In tal caso il coordinatore può subito iniziare il servizio del turno successivo, mentre in generale nel Token Passing si deve attendere che il token giunga ad un cliente che ne abbia bisogno.
Siamo nelle ipotesi di alto traffico quindi è probabile che il token dopo pochi passi finisca ad un processo che lo utilizzerà, quindi la differenza di prestazioni tra le due soluzioni sarà legata al tempo necessario al coordinatore per connettersi e svegliare un cliente.
A tal proposito è possibile introdurre un'evoluzione: duplicando la parte dei processi del gestore del ring deputata al risveglio dei clienti, è possibile stabilire il collegamento con il cliente del turno successivo prima che l'attuale servizio sia completato.

Scalabilità:
La scalabilità è limitata dalle dimensioni del ring, in quanto ciò determina il tempo necessario per il completamento di un giro da parte del token e quindi per poter servire le richieste.
D'altra parte questa limitazione è legata al numero di nodi non al numero di fruitori effettivi della risorsa presenti nel ring, come avviene invece nel Token Passing, su ogni nodo potrebbe teoricamente esserci un numero qualunque di clienti.
Consideriamo poi che la struttura potrebbe essere modificata introducendo il concetto di sottoring: ogni coordinatore raccoglie le richieste del suo sottoring e facendo poi parte del superring attende un supertoken in cui introdurre in blocco le richieste ricevute, gestite poi da un supercoordinatore.
In ogni caso è possibile ottenere un numero di utenti che utilizzano la risorsa senza degrado di prestazioni, presumibilmente superiore rispetto al Token Passing.

Gestione dei crash:
La nostra soluzione presenta il limite di un coordinatore centralizzante la cui caduta implica il blocco totale del servizio, ma l'ipotesi di guasto singolo insieme alla replicazione del coordinatore ci consente di eliminare questo handicap.
Notiamo invece che la rilevazione di guasti negli elementi del ring avviene in tempi più brevi, in quanto la durata massima per un giro del ring è predicibile con maggiore precisione dipendendo esclusivamente dal numero degli elementi del ring; le operazioni che devono effettuare i "Nodi" prima di passare nel ring sono infatti molto più veloci rispetto ai tempi necessari per passare il token o per accedere alla risorsa.
Un crash di un cliente comporterà del tempo per la rilevazione da parte del coordinatore del ring, ma non influirà in alcun modo sulla gestione del token.
Il token non viene mai perso e non occorrono protocolli di elezione.

Fairness:
Se si vuole garantire solo l'accesso in mutua esclusione e non l'ordine di prenotazione, si possono implementare politiche diverse nell'ordine di servizio delle richieste. I coordinatori infatti hanno l'elenco delle richieste raccolte in un intero giro possono dunque scegliere liberamente l'ordine con cui eseguirle(es: si può favorire chi ha fatto meno richieste o richieste meno onerose, oppure si può semplicemente invertire l'ordine di servizio dei nodi: (1,2...N) (N,N-1,..,1) (1,2...N)ecc. oppure ruotarlo (1,2,..,N) (2,3,..N,1) ecc. eliminando in tal modo l'ordine di precedenza che si verrebbe naturalmente a creare nella struttura a ring).

Limitarsi ad attendere per un certo tempo le richieste o attenderne un certo numero prima di iniziare a servirle non sortisce lo stesso effetto perchè finisce con il privilegiare i clienti più veloci o in qualche modo favoriti nell'accesso alla risorsa.

Trasparenza:
I clienti non sono tenuti a conoscere direttamente la risorsa e non devono gestire direttamente il token e i problemi della rete.

Nel caso in cui si abbia a disposizione un pool di risorse anzichè una singola risorsa, avere a disposizione tutte le richieste consente ai coordinatori di ridistribuire al meglio il carico sulle varie risorse.