Esame di **Tecnologia per il commercio elettronico (01ENY)**
Corso di Laurea in Ing. dell'Organizzazione d'Impresa

Prova scritta di teoria (12/7/2012)

> NOTA
> Le tracce delle soluzioni fornite in questo testo sono da considerarsi solo come un aiuto per comprendere i principali punti da toccare nel risolvere gli esercizi proposti ma non sono né esaustive né presentate in forma adeguata per l'elaborato da consegnarsi in sede d'esame.
> In particolare per molti esercizi le soluzione è volutamente schematica e ci si attende che il candidato spieghi adeguatamente i singoli punti, per dimostrare reale comprensione dell'argomento invece che semplice capacità mnemonica di ricordare i punti elencati nelle slide (o in queste tracce di soluzione).

Esercizio 1 ***(punti: 6)***
In HTTP/1.1 è disponibile la tecnica del *pipelining*. Illustrare in che cosa consiste e discuterne vantaggi e svantaggi (sia per il client sia per il server) confrontandola con l'apertura simultanea di N canali paralleli tra il client ed il server.

Traccia di una possibile risposta

- *pipelining = il client trasmette tante richieste senza aspettare la risposta del server. che invierà le risposte ordinatamente, tutto sullo stesso canale HTTP*
- *vantaggio = il client non deve attendere la risposta del server, per ogni client si usa un solo canale per tutte le interazioni*
- *svantaggio = lentezza rispetto ad aprire N canali in parallelo (soluzione che però sovraccarica il server e genera un notevole overhead di rete per l'apertura e chiusura di tanti canali TCP e la perdita delle informazioni di congestione TCP)*

Esercizio 2 ***(punti: 6)***
Si desidera attivare un servizio web che renda disponibile un'applicazione in tecnologia ASP che effettua tanti calcoli e tanti accessi ad un database. Proporre un'architettura in grado offrire efficacemente questo servizio, identificando sia i componenti logici sia gli oggetti hardware su cui tali componenti vengono eseguiti. Identificare quindi i punti critici di tale architettura ed indicare come potrebbero essere risolti.

Traccia di una possibile risposta

- *schema (server HTTP + server applicativo ASP + server DBMS)…*
- *il server HTTP deve soprattutto avere un buon collegamento di rete, il server applicativo una buona CPU e tanta RAM, il server DBMS dischi veloci e tanta RAM;*
- *per superare i punti critici si possono moltiplicare i server dell'elemento critico ed usare load balancer per distribuire il carico sui vari server di un determinato livello*

Esercizio 3 ***(punti: 4)***
Se in un form viene inserito un controllo di input di tipo `file` è obbligatorio trasmettere il form al server col metodo `POST` e con tipo MIME `multipart/form-data`. Spiegare perché il W3C ha scelto questa strategia, ossia perché non è ragionevole trasmettere il form (a) con `GET` oppure (b) con POST ma col tipo MIME di default.

Traccia di una possibile risposta

- *a) con GET i parametri dei form sono trasmessi nella URL che spesso ha una dimnesione massima (tipicamente 256 caratteri) e non permette quindi la trasmissione di un file molto grande;*
- *b) con POST i parametri dei form sono passati nel body (che non ha limiti di lunghezza) ma il tipo MIME è* `application/x-www-form-urlencoded` *mentre il file tramesso deve avere uno specifico tipo MIME corrispondente al suo contenuto (es.* `image/jpeg`*).*

Esercizio 4 ***(punti: 5)***
Un server web concorrente è installato su un computer dotato di 4 CPU a 2 GHz, 16 GB di RAM, scheda di rete a 10 Mbps e disco (non frammentato) da 100 GB, 20 ms e 20 MB/s. Sapendo che il server è collegato ad Internet tramite una linea ADSL da 2 Mbps, calcolare il numero massimo di client servibili al minuto, sapendo che il tempo di attivazione di un processo è 20 ms, la dimensione media di una richiesta è 2 kB e di una risposta 128 kB; inoltre, per generare una risposta, il server svolge un'elaborazione che richiede l'esecuzione di dieci milioni di istruzioni e la lettura da disco dei dati di 10 file da 1 MB ciascuno.

Traccia di una possibile risposta
attivazione di un figlio:

$$T_F = 0.02\,\mathrm{s}$$

lettura della richiesta:

$$T_R = \frac{2 \cdot 1024 \cdot 8\,\mathrm{bit}}{2 \cdot 1024 \cdot 1024\,\mathrm{bps}} = 8/1024\,\mathrm{s} = 0.008\,\mathrm{s}$$

lettura dati dal disco:

$$T_D = 10 \cdot \left(20\,\mathrm{ms} + \frac{1 \cdot 1024 \cdot 1024 \cdot 8\,\mathrm{bit}}{20 \cdot 1024 \cdot 1024 \cdot 8\,\mathrm{bps}}\right) = 0.7\,\mathrm{s}$$

esecuzione istruzioni:

$$T_C = \frac{10 \cdot 10^6 \mathrm{istruzioni}}{2 \cdot 10^9 \mathrm{ips}} = 0.005\,\mathrm{s}$$

invio della risposta:

$$T_W = \frac{128 \cdot 1024 \cdot 8\,\mathrm{bit}}{2 \cdot 1024 \cdot 1024\,\mathrm{bps}} = 0.5\,\mathrm{s}$$

tempo per servire un client

$$T_1 \approx T_F + T_R + T_D + T_C + T_W = 0.02 + 0.008 + 0.7 + 0.005 + 0.5 = 1.233\,\mathrm{s}$$

massimo throughput

$$M = N_{CPU}/T_1 \approx 4/1.233 = 3.244\,\mathrm{client/s} \rightarrow 194.64\,\mathrm{client/min}$$

Esercizio 5 ***(punti: 6)***
Dato il seguente form:

```
<form name="StatoCivile" method="POST"
    action="http://anagrafe.torino.it/richiesta.php">
  Nome e cognome: <input type="text" name="N" size="64">
  <br> Anno di nascita: <select name="ANNO">
    <option>1960</option> <option>1970</option>
    <option>1980</option> <option>1990</option>
  </select>
  <input type="hidden" name="KEY" value="poli007">
</form>
```

trasmesso tramite HTTP/1.1, illustrare quali dati transitano a livello di canale logico quando l'utente inserisce il nome `Giovanni Pautasso` e seleziona il valore `1970`.

Traccia di una possibile risposta

Il messaggio HTTP trasmesso avrà almeno le seguenti componenti come header e body:

```
POST /richiesta.php HTTP/1.1
Host: anagrafe.torino.it
Content-type: application/x-www-form-urlencoded
Content-Length: 41

N=Giovanni+Pautasso&ANNO=1970&KEY=poli007
```

Esercizio 6 ***(punti: 6)***

Tipicamente i server di commercio elettronico offrono il loro servizio tramite protocollo *TLS*: indicare quali proprietà di sicurezza offre tale soluzione e quali rischi residui devono ancora essere considerati.

Traccia di una possibile risposta

- *TLS fornisce: autenticazione del server ed opzionalmente anche del client; riservatezza (opzionale), autenticazione, integrità, anti-replay e anti-cancellazione per i dati trasmessi*
- *TLS non protegge da attacchi DoS e non protegge i dati una volta estratti dal canale (devono quindi essere adeguatamente protetti sul server, ad esempio tramite cifratura e/o controllo accessi)*