Valutazione delle prestazioni

1. Introduzione

Con il termine **Valutazione di prestazioni** (performance evaluation) si intende un insieme di attività rivolte alla determinazione delle caratteristiche di un sistema sulla base del suo comportamento.

Attraverso la valutazione di prestazioni di un sistema è possibile sapere "cosa" esso può offrire sotto determinate condizioni operative, in modo da delimitare l'area applicativa dove può essere utilizzato con successo.

La valutazione delle prestazioni di un sistema è utile ad ogni fase della sua vita , cioè nella fase di progetto, costruzione, vendita, uso ed aggiornamento, perché in ognuna di queste fasi ci si trova spesso di fronte alla necessità di fare delle scelte, confrontando le prestazioni di un sistema con quelle di altri sistemi, oppure di capire quali siano le condizioni che permettano di rendere massime le prestazioni.

I sistemi da valutare sono in genere così diversificati uno dall'altro che non è possibile definire una metodologia standard , per cui occorre ogni volta selezionare:

- ***la corretta misura di prestazioni***: cioè occorre che i risultati ottenuti siano quelli giusti, cioè quelli che descrivono in modo completo le prestazioni fornite dal sistema. Ciò richiede una accurata analisi degli indici prestazionali in modo da individuare quelli più significativi. Individuare gli indici prestazionali adatti non è operazione semplice. A parte alcuni indici comunemente usati, uno dei compiti di chi esegue la valutazione di prestazione è verificare se gli indici usati descrivono in maniera completa le caratteristiche del sistema e ricercare eventualmente altri indici prestazionali, utili per lo specifico sistema sotto esame.
- ***il corretto ambiente di misura***: cioè occorre definire bene le condizioni al contorno del sistema in modo da operare in condizioni simili a quelli reali in uso. E' questo un altro punto critico della valutazione di prestazioni che è fortemente influenzata dalle condizioni operative. Per tale motivo, è necessario uno studio approfondito per individuare i parametri che caratterizzano le condizioni operative, in modo da assegnare ad essi i valori corretti.
- ***la tecnica di valutazione più adatta***: le tecniche più usate sono quelle di tipo analitico e quelle di tipo simulativo. La scelta di una o dell'altra tecnica è legata al tipo di problema e al tipo di valutazione che si vuole realizzare. In alcuni casi la valutazione può essere ottenuta attraverso il monitoraggio di un sistema reale, anche se ciò pone diversi problemi pratici legati alla difficoltà di analizzare variabili fisiche interne al sistema.

Ogni valutazione richiede:

- ***un'intima conoscenza del sistema modellato***: cioè non è possibile valutare un sistema se prima non abbiamo capito veramente a fondo come esso funziona. Infatti, solo così è possibile realizzare un modello che tenga conto di tutti gli aspetti del sistema, anche quelli più subdoli e sottili che spesso sfuggono ad una prima analisi superficiale. Va sottolineata la criticità di questo punto, poichè è sufficiente un errore nella rappresentazione di un aspetto apparentemente secondario, per falsare in maniera significativa i risultati della simulazione.
- ***un'accurata selezione della metodologia, del workload, e dei tools***: La metodologia è importante poichè sistemi diversi (o fasi diverse nella vita di un sistema) possono essere analizzati meglio con una metodologia piuttosto che con un'altra. Il workload costituisce il carico che viene offerto al sistema per la valutazione. A seconda del tipo di sistema in esame, il workload assume aspetti diversi; se ad

esempio si sta valutando un sistema di elaborazione, il workload sarà costituito da istruzioni da eseguire, mentre se si sta valutando una Computer network, il workload sarà costituito da messaggi da trasmettere. Infine, il tool è lo strumento di simulazione più adatto al problema (spesso vengono utilizzati dei tools general-purpose già pronti. In alternativa è possibile realizzare dei tools dedicati al modello e quindi molto più veloci ed efficienti). La scelta del tool è importante perchè non esistono strumenti universali, adatti per tutte le occasioni e quindi, a seconda del tipo di valutazione che si vuole effettuare, può essere più adatto un tool invece che un altro.

Il primo passo nella realizzazione delle valutazioni di prestazioni è comunque la definizione del problema reale e la sua conversione (rappresentazione) in una forma in cui sia possibile usare le tecniche e i tools più adatti. E' questo un punto chiave nella valutazione del sistema, che richiede uno studio per mettere in risalto tutti i suoi aspetti significativi (trascurando quegli aspetti che pur essendo importanti nel sistema reale, sono irrilevanti per il tipo di valutazione che si vuole realizzare) e li traduce in forma modellabile attraverso il tool utilizzato.

2. Principali errori

I risultati ottenuti attraverso una valutazione di prestazioni vanno esaminati con molta attenzione, ed in modo critico.

Bisogna assolutamente evitare di prendere per buoni dei dati affetti da errore e bisogna sempre mettere in atto strategie di debug che permettono di rivelare la presenza di errori e di validare i risultati ottenuti. La presentazione di risultati errati, oltre che a far perdere credibilità all'autore della valutazione ed alle tecniche utilizzate, può produrre effetti disastrosi sotto diversi aspetti: economici, di affidabilità, di qualità del servizio ottenibile, ecc.

I principali errori che vengono fatti nelle valutazioni delle prestazioni sono i seguenti:

- ***Assenza di obiettivi***: non esistono modelli general-purpose ma ogni modello deve essere sviluppato con un chiaro obiettivo in mente: è importante capire il problema ed identificare il modello più adatto al problema che bisogna risolvere. Questo significa che il modello deve essere fatto per mettere in risalto solo gli aspetti che ci interessano senza dovere rappresentare tutti i dettagli dell'intero sistema: quindi è importante che, quando facciamo la valutazione di prestazioni, abbiamo chiaramente in mente cosa vogliamo ottenere da essa.
- ***Approccio non sistematico***: i parametri, le variabili da misurare e il workload non possono essere scelti in modo arbitrario, ma, avendo bene in mente i risultati a cui miriamo.
- ***Performance metrics inadatte***: le performance metrics sono tutti i parametri che noi misuriamo in una valutazione di prestazioni (throughput, tempo di ritardo, affidabilità del sistema, ecc.).Una scelta inadatta di tali parametri fornisce una valutazione incompleta del sistema.
- ***Workload non rappresentativo delle condizioni reali***: il workload che utilizziamo per la valutazione del sistema deve avere una corrispondenza con il workload reale che il sistema troverà in condizioni operative, altrimenti la valutazione effettuata non avrà alcuna utilità.
- ***Tecnica di valutazione inadatta***: le 3 tecniche utilizzabili sono: la simulazione, i modelli analitici, le misure. Allora in base ai risultati che vogliamo ottenere è importante utilizzare la tecnica adatta: normalmente vengono utilizzate le simulazioni e i metodi analitici, mentre le misure si usano solo quando il sistema è fisicamente disponibile.

- ***Trascurare parametri importanti***: se nel fare il modello non si tengono in considerazione alcuni parametri che possono invece avere una notevole importanza, allora si otterranno dei risultati che poi saranno notevolmente differenti da quelli veri.
- ***Livello di dettaglio inappropriato***: occorre evitare formulazioni del problema troppo dettagliate o troppo generiche. Infatti nell'eccessivo dettaglio si rischia spesso di perdersi nei particolari e magari poi di non mettere in evidenza gli aspetti più importanti: il modello che si ottiene in questo caso è complicatissimo, con in genere un numero elevato di errori.Nel caso di un simulatore sarà un problema farlo girare. D'altro canto l'eccessiva genericità è anch'essa da evitare: non consente di mettere in risalto gli aspetti da valutare.
- ***Errata analisi dei risultati***: la valutazione fornisce dei risultati, ma bisogna poi capire ciò che essi rappresentano, cioè dai risultati bisogna estrapolare il comportamento del sistema. A parte i problemi legati agli errori o alle approssimazioni che noi facciamo, a volte c'è il problema di capire se qualche risultato è oppure no significativo, cioè se ha senso oppure no: in modo da evitare di prendere grosse cantonate. Ciò si collega ancora al problema della correttezza dei risultati ottenuti dalla simulazione, cioè un'attenta analisi dei risultati consente di capire se essi hanno senso e se quindi la valutazione è stata condotta correttamente. Va comunque sottolineato che, anche se la simulazione è stata condotta correttamente fornendo risultati esatti, l'analisi dei risultati potrebbe essere errata. Ciò può derivare da una scarsa conoscenza del sistema, per cui non ci si riesce a spiegare alcuni aspetti del suo comportamento, oppure da errata interpretazione delle cause che hanno determinato i valori ottenuti degli indici prestazionali.
- ***Assenza di analisi di sensitività***: l'analisi di sensitività permette di capire come la variazione di un parametro iinfluenzi il comportamento del sistema: ci sono parametri che influenzano poco il comportamento del sistema, e parametri particolarmente critici (una cui leggera variazione comporta una notevole variazione del comportamento del sistema). Ad es. nel calcolo del tempo di ritardo nelle reti con trasmissioni di messaggi a diversa priorità, il carico ad alta priorità influenza moltissimo il comportamento della rete, mentre il carico a bassa priorità lo influenza meno. Il comportamento della rete pertanto dipenderà fortemente dal carico ad alta priorità, a cui il sistema darà la precedenza rispetto al carico a bassa priorità. Se non si è coscienti della diversa sensibilità di un sistema rispetto a certi parametri, sarà impossibile comprendere a fondo il suo comportamento.
- ***Trattamento inadatto dei valori singolari (outliers***): nel fare delle campagne di misure capita che ci sia ogni tanto qualche punto, detto valore singolare, il cui valore risulta completamente scorrelato dagli altri: questo valore deve essere ovviamente scartato in modo da non falsare i risultati.
- ***Inadatta presentazione dei risultati***: i risultati ottenuti possono essere tantissimi e per meglio valorizzare il lavoro fatto è estremamente importante presentare i risultati in modo adatto. A tale scopo è bene non limitarsi a delle semplici tabelle bensì graficare i risultati: in questo modo è possibile presentare chiaramente i risultati e sono anche possibili confronti con altre curve in condizioni di carico differente per lo stesso sistema. E' inoltre importante che le curve riportino negli assi le grandezze espresse nel modo più significativo, in modo da semplificarne la lettura. Per esempio, il workload può essere espresso in Kbytes/Sec. Oppure in percentuale della banda occupata (che spesso è più significativa del valore assoluto del workload).
- ***Omissione di assunzioni e limitazioni***: spesso quando si fanno valutazioni di prestazioni si omettono, volontariamente o involontariamente, alcune assunzioni e le limitazioni di base, che invece dovrebbero sempre essere ben chiare. Infatti, chi analizza i risultati delle valutazioni effettuate deve sempre conoscere le assunzioni e limitazioni in modo da avere una chiave per interpretare tali risultati. L'arte del *retail game* (gioco del venditore) è l'arte di interpretare i risultati della valutazione in modo da

presentarli sempre in positivo, cioè in modo da dimostrare che in nostro sistema è migliore di quello degli altri, senza però mentire sui risultati, ma semplicemente presentando bene tutti gli aspetti positivi del prodotto e minimizzando gli aspetti negativi.

3. Selezione della tecnica di valutazione

Vediamo quali sono i criteri in base ai quali bisogna selezionare una delle 3 tecniche di valutazione:

CRITERIO	MODELLI ANALITICI	SIMULAZIONE	MISURE
Stadio del sistema in cui si può usare la tecnica	In qualunque stadio	In qualunque stadio	Dopo che il sistema è stato realizzato
Tempo richiesto dalla tecnica	Breve (per un esperto analista)	Medio	Variabile a seconda la complessità del sistema
I tools usati nelle varie tecniche	Gli analisti	I linguaggi dei computer	Gli strumenti di misura
Accuratezza	Bassa	Moderata	Variabile a seconda i strumenti usati e il tipo di misura (diretta o indiretta)
Compromesso tra complessità della tecnica e bontà della valutazione	Facile	Moderato	Difficile (perché mettere insieme la soluzione è spesso abbastanza complicato
Costi	Bassi	Medi	Elevati
Vendibilità	Bassa	Media	Elevata

Quando si possono usare le varie tecniche di valutazione?

- Le **misure** sono possibili solo se esiste il sistema da valutare o qualcosa di simile al sistema da valutare.La misura è una forma di valutazione di prestazioni normalmente complessa e difficile che richiede la disponibilità di adeguati strumenti di misura ed una stretta correlazione con l'HW ed il SW del sistema. Ciò implica un'approfondita conoscenza del sistema (non facile da acquisire) . E' un approccio interessante perchè i risultati ottenuti non si riferiscono ad un modello del sistema (come nel caso delle simulazioni o delle analisi) ma al sistema reale e pertanto sono estremamente significativi. Presenta però diverse limitazioni:
    - Innanzitutto occorre evitare che i modelli HW/SW usati per la valutazione interferiscono col sistema falsandone il comportamento e quindi i risultati ottenuti.
    - Agendo su un sistema fisicamente reale è in genere difficile apportare modifiche alle strutture, per cui la valutazione fa riferimento ad un ben preciso contesto applicativo. Ad esempio, nel caso si stia valutando una rete, sarà abbastanza difficile operare delle modifiche sul numero di nodi, sulla lunghezza della rete, sul bit rate, etc.
    - Non è facile valutare il comportamento in condizioni anomale (ad esempio guasto di qualche nodo) poichè tale condizione non è facile da realizzare(a meno di non decidere di sacrificare parte del sistema per la valutazione di prestazioni).

I modelli analitici e le simulazioni possono sostituire le misure in assenza di sistemi disponibili.

- L'**approccio analitico** consiste nella rappresentazione in forma matematica del sistema e nella sua valutazione attraverso opportune formule risolutive. Ampiamente usati sono i modelli basati sulla teoria delle code che permettono di utilizzare risultati analitici già collaudati.
  Il principale vantaggio dell'approccio analitico è insito nell'approccio stesso e consiste nella trasparenza del modello che essendo rappresentato in forma analitica può essere facilmente verificato da chiunque. Per tale ragione i modelli analitici sono preferiti quando si deve valutare un nuovo sistema ed occorre valutare i risultati. Questi possono essere facilmente verificati e pertanto sono molto credibili.
  Per contro il modello analitico si presta solo a rappresentare aspetti molto generali di un sistema (e questo potrebbe essere in qualche caso un aspetto positivo) che sono utili nella prima fase di valutazione di un sistema.
  L'approccio analitico limita inoltre fortemente il tipo di valutazione che è possibile realizzare, a causa della difficoltà a rappresentare analiticamente, in modo corretto e con il sufficiente livello di dettaglio, i vari indici prestazionali.

- L'**approccio simulativo** si basa sulla realizzazione di un modello del sistema e sulla valutazione attraverso un opportuno strumento di simulazione.
  Il modello, va realizzato con una tecnica adatta al tool di valutazione impiegato. Il principale vantaggio della simulazione risiede nel fatto che se il modello è ben realizzato e sufficientemente dettagliato è possibile ottenere dei risultati molto simili a quelli che sarebbero stati forniti dal sistema reale. Il modello può inoltre essere realizzato in modo da evidenziare specifici aspetti del sistema che si vogliono investigare. E' quindi possibile variare le condizioni a contorno e valutare come queste influiscono sul comportamento del sistema.
  Mediante la simulazione è possibile analizzare il comportamento del sistema sia in condizioni stazionarie (a regime), che in transitorio. Normalmente il sistema è valutato a regime ed in tal caso occorre prestare attenzione a non includere i risultati relativi a fasi transitorie.

3.1 Precisione offerta da varie tecniche di valutazione

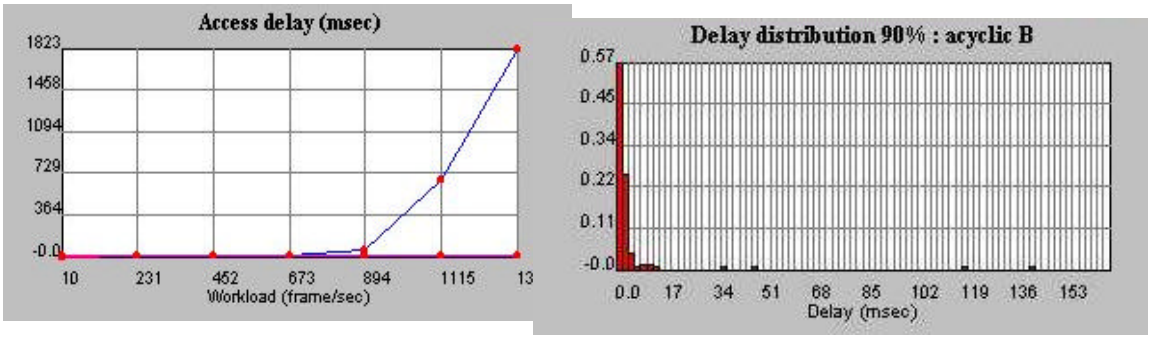
- I modelli analitici sono imprecisi perché quasi sempre, ad eccezione del caso di sistemi molto semplici, occorre fare un gran numero di semplificazioni per poter ottenere una formulazione matematica del sistema. Tuttavia è una tecnica importante perché fornisce dei risultati generali sugli aspetti fondamentali del sistema che poi possono essere particolarizzati con la simulazione.
- Le simulazioni sono più precise ma possono richiedere molto tempo perché mentre il modello analitico alla fine è una formula matematica risolvibile in forma chiusa, il modello di simulazione è un software, in genere abbastanza complicato, che fornisce dei risultati affidabili solo dopo un lungo tempo di simulazione necessario per far andare a regime il sistema stesso.
- Le misure possono non fornire risultati accurati a causa della unicità (non ripetibilità) di alcuni parametri. Mentre nel caso di un modello analitico o di un modello simulativo si possono fare delle assunzioni di tipo teorico sul carico (spesso si stabiliscono alcuni tipi di carico standard che vengono utilizzati per valutare le prestazioni), nel caso delle misure il carico del sistema è quello reale di quel momento (unico e non ripetibile) ed il comportamento del sistema potrebbe non essere quello generale ma potrebbe dipendere da questo carico specifico.

Un altro aspetto importante nella valutazione di un sistema è la correlazione tra i vari parametri sistema:

- I modelli analitici sono quelli più vantaggiosi da questo punto di vista perché nella formula matematica si vedono subito le relazioni tra i vari parametri, pertanto essi permettono di evidenziare l'effetto mutuo di più parametri in modo chiaro.
- Con le simulazioni, invece, a volte non è chiaro il trade-off fra i diversi parametri a meno che il progettista del software simulativo non lo metta espressamente in evidenza.
- Anche le misure rendono difficile interpretare il legame fra i vari parametri, non esistendo alcuna regola teorica in tal senso. Non è facile capire se una variazione nelle prestazioni dipende da un cambiamento casuale dell'ambiente o dal valore di qualche particolare parametro.

Una cosa molto interessante da osservare è che 2 o più tecniche possono essere usate in modo sequenziale: ad es. prima con un modello analitico si trova il range adatto dei parametri e poi con la simulazione si studiano le prestazioni del sistema in quel range. L'uso abbinato delle due tecniche risulta molto vantaggioso perchè un modello analitico semplificato (e quindi facile da realizzare) permette di determinare velocemente, le condizioni operative che mettono in risalto i comportamenti desiderati, che possono poi essere investigati in modo accurato attraverso la simulazione.

Per molte "metrics" il valore medio è quello importante nel senso che le prestazioni del sistema vengono di norma valutate in termini di comportamento medio a regime. Tuttavia non bisogna trascurare la variabilità tra i vari valori ottenuti che in alcuni casi può essere più pericolosa del valore stesso. Ad esempio, in una misura di tempo di ritardo fra i messaggi spediti in una rete per controllo di processo, il valore del tempo di ritardo medio è un'informazione insufficiente perchè se si ha un'elevata variabilità di valori, un basso valore medio non esclude che ci possano essere diversi valori singoli molto più elevati del valore medio stesso. Ciò può essere molto pericoloso in un ambiente operativo di questo tipo, dove i ritardi devono, in genere, essere limitati.



Questo concetto è chiarito nelle due figure mostrate sopra. Nella figura a sinistra è possibile vedere l'andamento del Delay medio di una rete (ethernet nell'esempio considerato) al variare del workload. Come si vede, per valori del workload fino a circa 700 frames al secondo, il ritardo è molto basso, quasi nullo. Tuttavia, la figura di destra che mostra la distribuzione percentuale di messaggi (ad un valore di workload pari a circa 700 frames al secondo) in funzione del loro ritardo, evidenzia come ci sia una percentuale (anche piccola) di messaggi con un ritardo molto maggiore degli altri. Questa informazione può essere estremamente utile nella valutazione dei sistemi per controllo di processo.

Quando la valutazione si riferisce ad un sistema distribuito, con diversi agenti, bisogna distinguere fra prestazioni individuali e collettive. Le prestazioni individuali descrivono il comportamento di un singolo agente e danno un'indicazione sul tipo di servizi che il singolo utente può aspettarsi dal sistema. Le prestazioni collettive descrivono invece il comportamento dell'intero sistema e danno indicazioni sulla tipologia di servizi globalmente offerti. Ad esempio, nello studiare una "Computer network" può essere utile

analizzare separatamente il comportamento della rete da quello delle singole stazioni. Alcuni parametri (quale ad esempio la "fairness" sono strettamente legati a singole stazioni e dipendono dalla loro posizione, dal workload che offrono, ecc.)

Utilizzazione delle risorse, affidabilità e disponibilità sono "metrics" globali, mentre tempo di risposta e throughput possono essere visti come "metrics" sia globali che individuali.

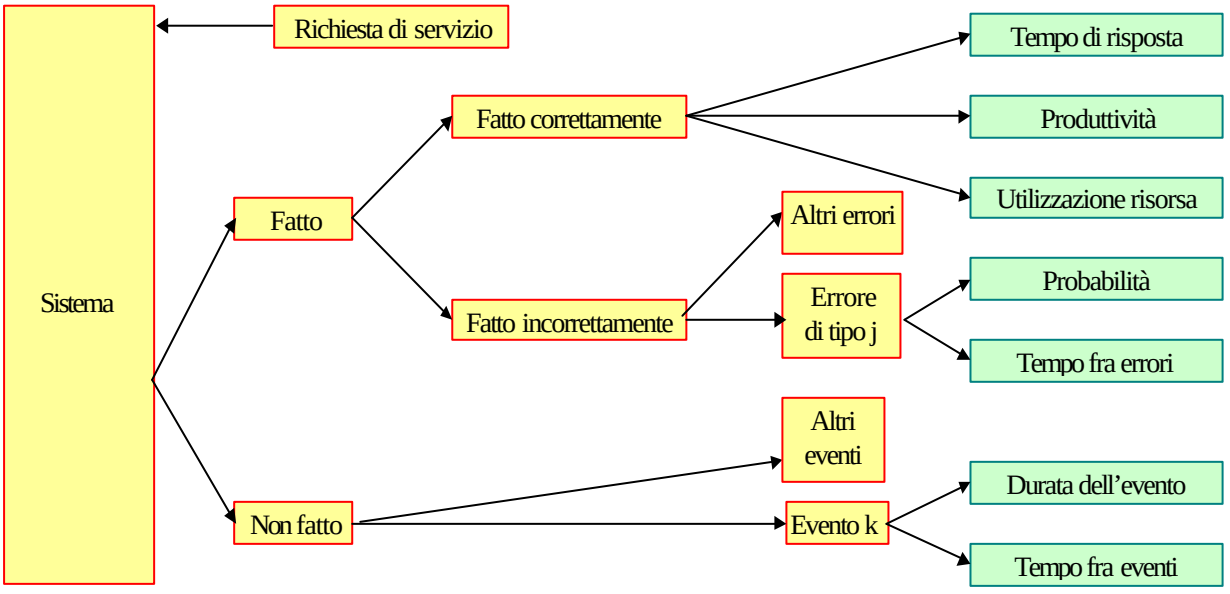
Nella scelta del tipo di "metrics" da usare per valutare un sistema è opportuno riferirsi ai seguenti criteri:

- "**Low variability**", cioè selezionare metrics che producono campioni. caratterizzati da bassa variabilità. Ciò permette di ridurre il numero di misure che occorre fare per valutare il sistema, il che significa, nel caso di una simulazione, poter ridurre la durata della simulazione stessa.
- "**Non redundancy**" cioè selezionare metrics che contengono indicazioni diverse. Usare due metrics diverse per valutare lo stesso parametro serve solo a creare confusione.

"**Completeness**", cioè le metrics utilizzate devono essere sufficienti a definire il comportamento del sistema in modo completo. Ciò significa che di volta in volta può essere necessario inventare delle metrics specifiche, utili alla valutazione del particolare ambiente operativo.

3.2 Selezione degli indici prestazionali (performance metrics)

Gli indici prestazionali sono i valori di quei parametri che descrivono il comportamento del sistema. Un sistema può effettuare una richiesta di servizio in modo corretto, incorretto, o non effettuarlo affatto. Per ogni tipo di comportamento sono diversi gli indici prestazionali cui occorre fare riferimento. La figura sotto riportata mostra un sistema cui l'analista chiede di svolgere una qualche funzione, cioè di eseguire un servizio (rappresentato da un certo workload) :



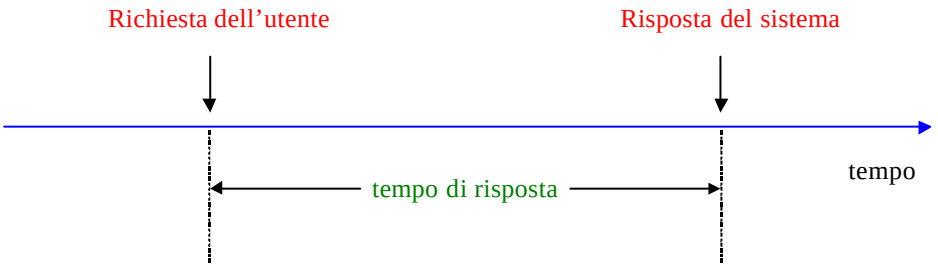
- se il sistema esegue il servizio correttamente allora le "metrics", cioè quei parametri che misuriamo come indici prestazionali del sistema, sono chiamate *responsiviness* (cioè tempo di risposta, tempo di attesa

in coda, ecc.), *productivity* (throughput), *utilization* (dà un'indicazione della percentuale di uso della risorsa, cioè misura l'efficienza nell'utilizzo delle risorse del sistema).Complessivamente definiscono la velocità (speed) del sistema.

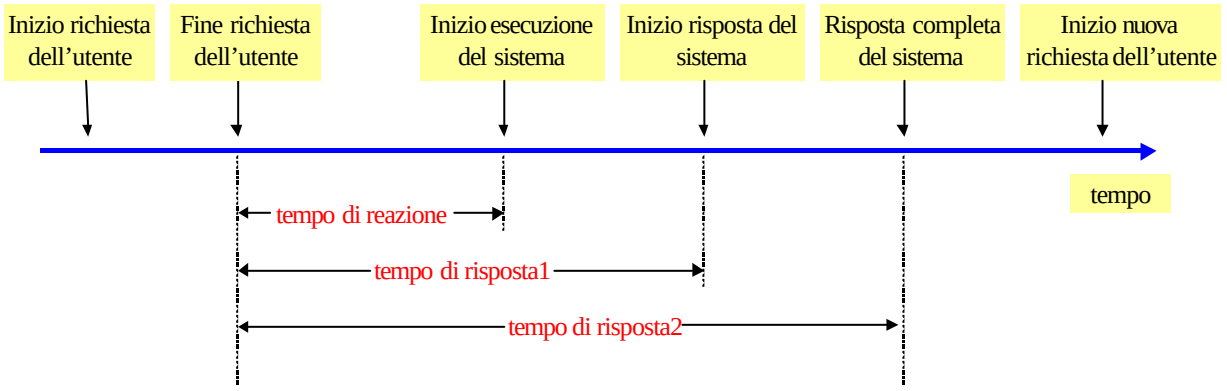
- se il sistema non opera correttamente vuol dire che si è verificato un certo tipo d'errore per cui ciò che possiamo misurare e poi modellare è la probabilità che quel tipo d'errore si manifesti. Le metrics associate sono espresse in termini di affidabilità (reliability) del sistema.
- se il sistema non funziona (unavailable, cioè non disponibile) vuol dire che si è verificato un certo evento (ad es. un guasto) per cui si tratta di misurare la durata dell'evento e il tempo fra 2 eventi consecutivi della stessa classe, in questo modo possiamo classificare i *modelli di fallimento* (crash dell'intero sistema o solo di qualche suo nodo, oppure errori di omissione in trasmissione e/o in ricezione, oppure errori di falsi contatti sporadici) e determinare la probabilità di ciascuna classe d'evento. Il tempo medio tra 2 guasti è un indice molto importante per misurare l'affidabilità di un sistema.

3.3 Performance metrics più usate

- **Response time** (tempo di risposta): è l'intervallo di tempo fra la richiesta dell'utente e la risposta del sistema. È importante mettere in evidenza quali sono le componenti del sistema che intervengono nella definizione di un tempo di risposta tra 2 eventi ben precisi.



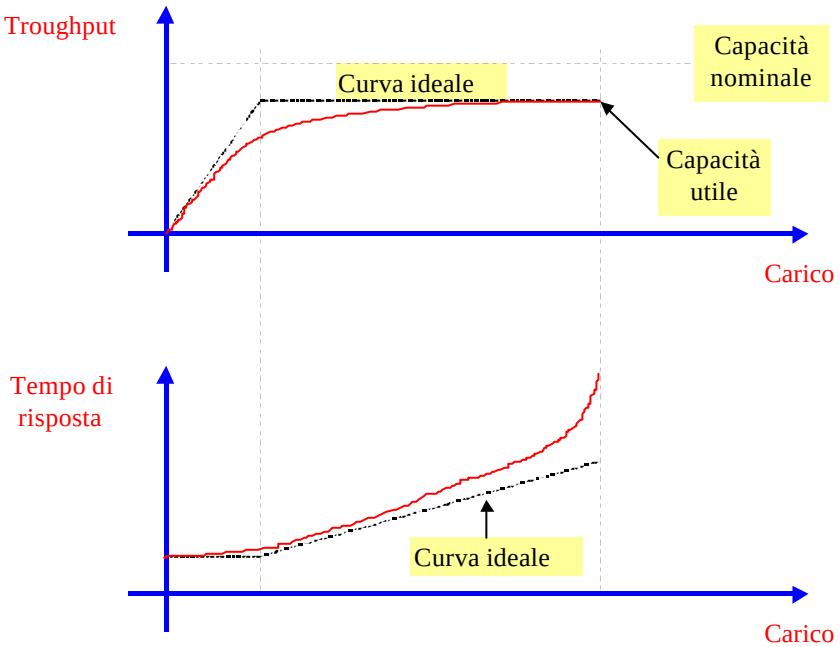
- **Reaction time** (tempo di reazione): è l'intervallo di tempo fra la sottomissione di una richiesta e l'inizio della sua esecuzione.



- **Stretch factor** (fattore di sforzo): è il rapporto fra il response time ad un certo carico e quello a minimo carico. Questo è un parametro molto importante perché permette di evidenziare il peggioramento della prestazioni del sistema rispetto al massimo che il sistema può fornire.
- **Throughput** (produttività):è la frequenza a cui le richieste possono essere servite dal sistema. Nel caso delle reti di calcolatori, le richieste sono rappresentate da messaggi (frames o pacchetti) mentre il

servizio consiste nella loro trasmissione, per cui il throughput viene rappresentato in terimini di messaggi trasmessi per unità di tempo. Se volessimo valutare il throughput di un sistema in termini di bit d'informazione trasmessi, allora tutta la parte header e trailer della frame deve essere scartata. Pertanto nella figura il load sarà espresso in frames/sec (carico effettivo della rete) mentre il throughput sarà espresso in bytes/sec.

- **Nominal capacity** (capacità nominale): è il massimo throughput in condizioni ideali. Nel caso delle reti essa coincide con la "bandwidth", cioè con la larghezza di banda del canale trasmissivo. Il throughput non può quindi essere **mai** superiore alla larghezza di Banda del canale. Ad es. se una rete trasmette con un bit-rate di 1Mbps al massimo, nel caso ideale, essa potrà avere un throughput di 1Mbps; poi in realtà la capacità nominale del canale sarà inferiore ad 1Mbps perché occorrerà considerare : gli intervalli tra le varie frames, gli errori di trasmissione, i vari header e trailer per ogni frame che non costituiscono informazione utile, ecc. Ciò deve essere sempre ben chiaro in mente ed utilizzato come riferimento per non commettere errori di valutazione.
- **Usable capacity** (capacità utile): è il massimo throughput ottenibile senza superare un delay prefissato. In genere nelle reti la curva reale del throughput tende asintoticamente a quella ideale al crescere del carico, in quanto solo per un carico maggiore della massima capacità del canale quest'ultimo viene sfruttato al massimo (se il protocollo di accesso è ben progettato). Però in corrispondenza i tempi di ritardo crescono anch'essi a dismisura; pertanto di norma per stabilire la capacità utile si fissa un tempo di ritardo massimo ammissibile ed in corrispondenza si legge il massimo throughput ottenibile.



- **Efficienza**: è il rapporto fra l'usable capacity e il nominal capacity, cioè è il rapporto tra il massimo throughput ottenibile e quello ideale. Questa efficienza si lega all'efficienza globale del sistema; supponiamo ad es. di avere 2 sistemi confrontabili costituiti da 2 protocolli simili: uno che utilizza frames con pochi bits di header e l'altro che invece utilizza frames con parecchi bytes di header, ovviamente questo secondo protocollo anche se fosse più affidabile nella consegna dei frames sarebbe ugualmente meno efficiente del primo perché la quantità di informazione che riuscirebbe a trasmettere sarebbe sempre molto minore del primo.

- **Utilizzazione**: è la frazione di tempo in cui una risorsa è impegnata per servire una richiesta. E' quindi misurabile come rapporto fra il tempo in cui la risorsa è occupata (busy time) ed il tempo totale considerato. Il tempo in cui una risorsa non viene utilizzata si chiama "idle time". Nella progettazione di un sistema è importante ottenere un bilanciamento del carico che permette di rendere massima l'utilizzazione delle risorse.
- **Reliability** (affidabilità): è misurabile in termini di probabilità d'errore o di tempo medio fra gli errori.
  - **Availability** (disponibilità): è la frazione del tempo totale in cui un sistema è disponibile per le richieste dei vari utenti. Il tempo in cui il sistema non è disponibile è chiamato "down time". Il tempo in cui il sistema è disponibile è chiamato "uptime". Il valore medio dell' "uptime" anche chiamato "mean time to failure" è un ottimo indicatore della disponibilità e dell'affidabilità di un sistema.

4. Workload

Il workload, cioè l'insieme di richieste che un utente fa ad un sistema, è uno degli aspetti più importanti quando si parla di valutazione di prestazioni di un sistema. Esistono 2 tipi di workload:

- **Real workload**: è quello osservato su un sistema durante le normali operazioni. Eè il workload con cui il sistema si troverà ad operare in condizioni operative reali, ma non è normalmente ripetibile.
- **Syntetic workload**: è quello che viene generato in modo sintetico, cioè esso ha caratteristiche simili a quelle del real workload di cui costituisce un modello. La sua caratteristica più importante è che può essere applicato ripetutamente in maniera controllata, potendo così effettuare le opportune valutazioni del sistema. Il syntetic workload può inoltre essere modificato facilmente, può essere adottato a diversi sistemi.

Il **syntetic workload** è quindi quello usato di solito. Esso va adattato alle caratteristiche del sistema da valuatare in modo da mettere in rilievo le caratteristiche.

Nel caso di sistemi di elaborazione i workloads sono in genere rappresentati da :

- **Addition instruction** : era il più usato tipo di workload nei vecchi sistemi dove le prestazioni dipendevano esclusivamente dalla cpu e le addizioni erano le operazioni più usate.
- **Instructions mixes** : è una specifica di varie istruzioni insieme alle loro frequenze d'uso. Nei sistemi attuali sostituisce le "Addition instruction" troppo semplici. Se ne usano diversi tipi; il più usato è il "**GIBSON mix**", sviluppato nel 1959 da Jack C.Gibson, che contiene 13 differenti classi di istruzioni. L' "instruction mix" presenta notevoli limiti per i sistemi attuali che contengono classi complesse di istruzioni difficili da considerare in **mix standard**. Inoltre nei sistemi attuali il tempo di esecuzione dipende dai modi di indirizzamento, dall'uso di "caches", dal livello ed efficienza del pipeline e dall'interferenza di altri dispositivi presenti nel sistema. Le "instruction mixes sono usate per misurare l'efficienza dei processori in termini di MIPS (milioni di istruzioni al secondo) o in MFLOPS (milioni di "floating-point operations" al secondo).
- **Kernels**, è una generalizzazione dell'instruction mixes, ed è rappresentato da una particolare funzione (di solito quella più usata frequentemente) formata da un certo gruppo di istruzioni. I Kernel più usati sono **Sieve Puzzle, Ackermann's Function, l'inversione di matrice e gli ordinamenti.** Il principale svantaggio dei Kernels è che essi di solito non usano le istruzioni di I/O e pertanto trascurano nella valutazione un fatto che influenza fortemente le prestazioni del sistema..
- **Syntetic programs** : sono dei programmi di test che attraverso dei "Loop" eseguono un numero definito di operazioni che fanno riferimento non solo a istruzioni ma anche ad operazioni di I/O. In tal modo viene superato il limite dei Kernels. Ne sono stati sviluppati diversi per essere usati in particolari

aree applicative. Particolarmente importante è la '**SPEC benchmark suite**" sviluppata da System Perforance Evaluation Cooperative (SPEC) che ha realizzato un gruppo di dieci test che permettono di stressare la CPU, l'unità aritmetica floating-point e la memoria.

Nel caso di sistemi distribuiti, la comunicazione influenza fortemente le prestazioni. Pertanto, la valutazione delle reti costituisce un importante componente della valutazione di tutti i sistemi distribuiti. Il workload, in tal caso, è costituito da messaggi da scambiare fra i vari nodi, ed il problema è quello di definire un "syntetic workload" che rispetti le condizioni operative dei sistemi reali. La caratterizzazione del traffico è un aspetto importante dei vari ambienti operativi in cui un sistema si trova ad operare.

- Nei normali sistemi per "office automation" il workload può essere rappresentato sotto forma di messaggi generati secondo una opportuna distribuzione (la distribuzione esponenziale è la più usata). Quando però l'ambiente operativo cresce per dimensioni e tipologia di dati scambiati (includendo traffico di tipo audio e video), la caratterizzazione del workload diviene più complessa e viene descritta mediante modelli matematici sviluppati ad hoc. Quando si ha a disposizione un sistema reale, del tipo di quello che si vuole analizzare, è possibile ricorrere a "trace" del traffico misurato.

- Nei sistemi per controllo di processo, il workload è spesso costituito da messaggi con stringenti vincoli temporali, o con caratteristiche di ciclicità. Anche per tale tipo ambiente operativo vengono definiti dei modelli ad hoc che tengono conto delle caratteristiche delle applicazioni supportate dalla comunicazione.

E' opportuno introdurre le seguenti definizioni:

- **System Under Test** (SUT): denota l'insieme di componenti che si stanno valutando.
- **Component Under Study** (CUS): denota il singolo componente del SUT considerato.

Ovviamente il workload va selezionato in base al sistema da valutare e non al singolo componente, per cui se ad es. dobbiamo valutare le prestazioni di una rete dobbiamo pensare ad un workload di rete complessivo e non dobbiamo invece pensare ad un workload per il singolo nodo della rete.

4. Workload

4.1 Caratterizzazione del workload

Il workload consiste di richieste di servizi oppure di utilizzo di risorse da parte degli utenti di un sistema. Poiché l'ambiente vero dell'utente (real-user environment) non è in genere ripetibile, per realizzare il giusto workload è necessario studiare tale ambiente, osservarne le caratteristiche chiave e costruirne un modello da potere usare ripetutamente.
Ad esempio vogliamo valutare la capacità di una rete Ethernet per trasmissione dati, per quanto riguarda il traffico vocale, cioè vogliamo vedere se è possibile una tale comunicazione telefonica su una rete ad accesso casuale (in questo caso infatti se la rete è congestionata si hanno molte collisioni e tempi di accesso possono diventare troppo lunghi per una conversazione vocale in cui le informazioni devono essere trasmesse con continuità per brevi intervalli di tempo): dopo aver realizzato la conversione analogico/digitale della conversazione sarà effettuato un campionamento a blocchi della voce per ottenere pacchetti contenenti più campioni, e saranno spediti in rete insieme agli altri pacchetti contenenti i dati. Per valutare il comportamento della rete Ethernet dobbiamo a questo punto caratterizzare il workload della rete: da una parte considereremo un modello di carico dati caratterizzato tipicamente da una distribuzione esponenziale,

e dall'altro un modello di carico vocale caratterizzato da raffiche di pacchetti d'informazione successivi seguiti da lunghe pause.
I parametri che caratterizzano il workload (valori delle richieste d'utente, delle richieste d'uso di risorse, ecc.) devono dipendere dal workload e non dal sistema. Quelle caratteristiche che hanno un impatto significativo sulle prestazioni del sistema vanno incluse fra i parametri del workload; nel nostro es. mentre potrebbe essere ininfluente il fatto che la voce sia maschile o femminile, potrebbe essere importante il fatto che si parli in italiano o in inglese nella durata delle pause tra 2 raffiche. Il livello di dettaglio del workload è relativo al dettaglio con cui all'interfaccia SUT-user viene specificato il workload.

Ci sono 4 modi in genere per caratterizzare un workload:

- il modo più semplice è quello di utilizzare la *richiesta più frequente*, cioè poiché si fanno richieste di tipo differente al sistema noi possiamo pensare di usare un workload che è semplicemente costituito soltanto dalla richiesta più frequente;
- oppure utilizzare un *miscuglio di richieste a frequenza diversa*, in questo caso bisogna generare un workload misto che tenga conto di un po' di tutto;
- oppure, nel caso delle misure, utilizzare un *Trace di richieste di un sistema reale*, cioè di utilizzare come workload una registrazione degli eventi reali che avvengono nel sistema in un certo intervallo di tempo arbitrariamente lungo. Il problema principale del trace è che la sequenza degli eventi registrati, per quanto lunga essa sia, è sempre troppo limitata rispetto alla grande quantità di memoria necessaria per contenerla.
- L'ultimo approccio che possiamo utilizzare è quello di fare *richieste mediate di servizi* con una prefissata probabilità di distribuzione nel tempo: tipicamente distribuzioni esponenziali o gaussiane.

Per *rappresentatività del workload* s'intende che il test workload deve essere rappresentativo del real workload. Questa rappresentatività deve essere verificata sotto 3 aspetti:

- *Arrival rate (frequenza d'arrivo) delle richieste*, cioè la frequenza di generazione delle richieste deve essere confrontabile con il real workload, ovvero devono avere la stessa distribuzione;
- *Resource demands* (richieste di risorsa) relativamente a ciascuna delle risorse chiave, cioè le richieste delle risorse chiave devono essere uguali sia nel syntetic workload che nel real workload;
- *Resource usage profile (profilo di uso delle risorse)*, cioè il nostro workload deve essere fatto in modo tale da sfruttare le risorse del sistema secondo un certo profilo di uso che deve essere uguale a quello del sistema reale (in un computer: stesso tempo d'uso della memoria centrale, del disco, ecc.).

Il workload permette di caricare il sistema con un certo numero di attività; il modo in cui l'attività viene assegnata al sistema permette di dedurre informazioni sul sistema stesso.

4.2 Media, Varianza e Dispersione

Uno dei modi più' semplici per caratterizzare il workload offerto ad un sistema è presentare un singolo numero che riassuma tutti i valori osservati nel sistema reale.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Il valor medio può essere considerato come una buona rappresentazione del workload, soltanto se non abbiamo grosse variazioni dei dati. In genere conviene caratterizzare il workload fornendo anche la

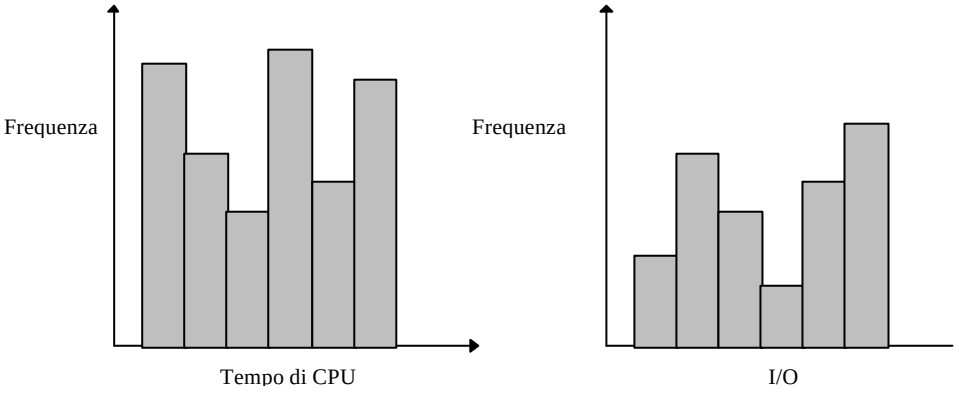
varianza dei campioni che si stanno osservando. Ovvero si può fornire al sistema un insieme di valori diversi del carico che vengono caratterizzati attraverso la varianza (s^2) e la deviazione standard (s)

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

La deviazione standard S, (che rappresenta la radice quadrata della varianza) è spesso più usata poichè è espressa nella stessa unità della media.
Può essere utile definire pure il Coefficiente di Variazione (CdV), ovvero il rapporto fra deviazione standard e media. Se CdV è zero, il parametro misurato è una costante, ed in tal caso ha senso rappresentare il workload con un valore medio. Ma se il CdV è elevato allora la media non è sufficiente e bisogna utilizzare un metodo diverso per la rappresentazione dei dati.

4.3 Istogramma di un singolo parametro.

Si è già detto come il workload possa essere rappresentato mediante un "trace" di richieste di un sistema reale. Ciò può essere realizzato mediante un istogramma completo di campioni, cioè un istogramma in cui in funzione del tempo viene fornita una serie di workload che rappresentano il workload che noi forniremmo ad un sistema reale.
Si consideri, ad esempio, un sistema che deve eseguire un certo numero di operazioni sulla CPU, alcune delle quali richiedono anche operazioni di I/O. E' possibile rappresentare in funzione del tempo la frequenza con cui vengono eseguite le operazioni che utilizzano la CPU e la frequenza con cui vengono eseguite le operazioni di accesso al disco.



In questo modo si ottiene una descrizione abbastanza accurata del workload anche se non necessariamente limitata, data l'impossibilità di registrare sequenze infinite di richieset. Se il "trace" che stiamo considerando è continuo, occorre allora renderlo discreto. A tale scopo, l'intero range è diviso in slots e si calcolano i valori che cadono in ogni sub-range.

Utilizzare un istogramma di un singolo parametro può comportare dei problemi qualora si abbia a che fare con un workload costituito da un numero elevato di campioni, questo significa andare a memorizzare una traccia di campioni che può essere molto lunga. Fissati n slot per ogni istogramma, se m è il numero dei parametri per ognuno dei componenti e k il numero di componenti selezionati, questo metodo richiede di presentare $n \bullet m \bullet k$ valori. Ciò può richiedere una quantità di memoria eccessiva e quindi questo metodo andrebbe utilizzato solo quando si ha una elevata varianza fra i singoli valori dei parametri.

5. Analisi dei sistemi da campioni

Quando eseguiamo delle misure sui sistemi otteniamo delle misure puntiformi, ovvero le valutazioni di prestazioni che noi eseguiamo non sono relative al sistema nella sua globalità, ma a specifici punti di lavoro del sistema. Otteniamo cioè, solo dei campioni che occorre quindi analizzare per comprendere quale significato attribuire ai valori trovati. Il problema che emerge con evidenza, è che risulta impossibile ricavare le caratteristiche esatte di un sistema basandosi solo sull'osservazione di alcuni campioni. Se, ad esempio, vogliamo valutare le caratteristiche di una rete, quello che facciamo è:

- Fornire un workload sotto forma di pacchetti dati;
- eseguire delle misure su questi pacchetti;

da queste misure cerchiamo di estrapolare il comportamento della rete.

Appaiono subito evidenti 2 considerazioni:

- Perchè la valutazione che stiamo effettuando sia significativa (nel senso che i dati siano in qualche modo interpetrabili) occorre definire delle condizioni al contorno, ben delimitate. Occorre cioè scegliere un workload ben identificabile (definito come frequenza media di arrivo dei messaggi, loro lunghezza media, probabilità di errore, ecc.), fissare un numero di nodi, la dimensione della rete, ecc. Ciò consente di ottenere una valutazione del sistema *in un ben preciso punto di lavoro*. Ribadiamo quindi che il risultato ottenuto non è una valutazione del sistema nella sua globalità, ma solo in un punto di lavoro. Si intuisce chiaramente come una valutazione esaustiva (o quantomeno, sufficientemente completa) di un sistema sia un'operazione lunga, che richiede diverse valutazioni puntiformi e consuma tempo.
- Ma, anche la valutazione esatta *di un singolo punto di lavoro* è un problema di non facile soluzione. Infatti, per determinare il comportamento esatto della rete occorrerebbero un numero infinito di campioni, il che ovviamente non è possibile visto che, considerata la finitezza della vita umana, siamo costretti a limitare la durata della nostra valutazione.

Quella che otteniamo è pertanto una rappresentazione, necessariamente approssimata, della realtà. Quindi, caduta l'illusione di poter valutare in modo preciso le prestazioni del sistema considerato, accontentiamoci di determinare con quale probabilità le caratteristiche del sistema sono contenute in un certo range, cioè di capire, quando facciamo una valutazione di prestazioni, in quale intervallo hanno significato i risultati che abbiamo trovato. Più piccolo è tale range, migliore è la valutazione.

Riprendiamo l'esempio di prima, supponiamo di voler misurare il tempo medio che impiega un pacchetto per giungere a destinazione, qualunque sia l'approccio che stiamo usando, analisi, simulazione, etc, quello che facciamo è generare dei pacchetti e misurare il tempo di ritardo. Alla fine abbiamo un campione (X1,X2,...Xn), cioè **n** valori ricavati; indichiamo con **Xm** la media relativa al campione. Tale media è diversa da quella **m** della popolazione di dati misurati (il valore μ si ottiene considerando un numero infinito di valori). Valgono le seguenti definizioni:

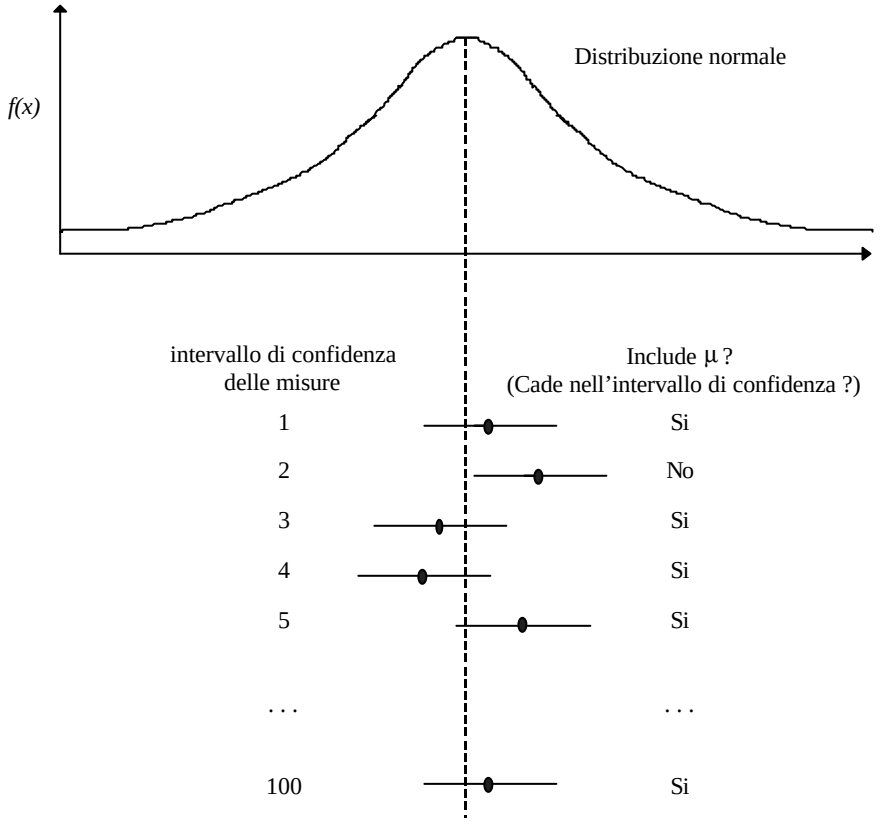
- **Xm** è il valore misurabile, definito *Statistica*
- **m** è il valore cercato, definito *Parametro*

Xm è il valore che otteniamo su un certo numero, finito, di dati; se invece analizziamo un numero infinito di dati otteniamo μ. Per un dato sistema, fissate le condizioni al contorno, i parametri sono fissi poichè

dipendono dalle caratteristiche del sistema, mentre le statistiche sono variabili in modo casuale essendo influenzate da variabili aleatorie presenti in tutti i sistemi (ad esempio i disturbi o gli errori accidentali). La media di un campione rappresenta una stima della media della popolazione. Non è possibile ottenere una stima perfetta della media della popolazione partendo da campioni di dimensione finita. Quando studiamo un sistema (sia attraverso misure che attraverso simulazioni) otteniamo set di valori in numero finito. Il comportamento del sistema, che estrapoliamo attraverso l'analisi dei valori osservati, si avvicinerà sempre più al comportamento reale del sistema all'aumentare del numero di valori osservati.

La stima di un parametro di una popolazione, data da un solo numero (per esempio, il valore Xm) è definita **Stima Puntuale** del parametro stesso. Se la stima è data da due numeri che si possono considerare, uno maggiore ed uno minore del parametro reale, allora si parla di **Stima per Intervallo.** Esempio**:** se diciamo che la misura di un tempo di ritardo è 23 mSec stiamo fornendo una stima puntuale. Se invece diciamo che il ritardo è compreso fra 23 +/- 3 mSec cioè fra 20 mSec e 26mSec, allora stiamo fornendo una stima per intervallo.

Un parametro molto importante è quello che viene chiamato *Intervallo di confidenza*. Questo è un intervallo (C1,C2) che con una elevata probabilità **P = 1- a** contiene la media μ della popolazione. Ad esempio se per un tempo di ritardo l'intervallo di confidenza è compreso tra 3ms e 3.5ms con una probabilità del 90%, significa che su 100 diverse valutazioni, il 90% dei campioni che misuriamo cadono nell'intervallo (3ms, 3.5ms). *L'intervallo di confidenza garantisce quindi la qualità della valutazione che stiamo eseguendo.*



Il grafico fa riferimento alla distribuzione normale, quella che normalmente si ottiene quando si eseguono misure sui reali, purchè il numero dei campioni sia superiore a 30, in cui si osserva il valore medio μ (relativo al campione infinito) e gli intervalli di confidenza relativi a certi campioni. Come si vede alcuni di

questi intervalli contengono il parametro μ, altri invece non lo contengono. Supposto di avere eseguito cento misure, il numero di quelli che cadono dentro sarà $\geq 100(1-\alpha)$, mentre quelli che cadono fuori saranno $\leq 100(\alpha)$. Quindi l'intervallo di confidenza fornisce un intervallo entro il quale è possibile dare significato alle misure effettuate.

- **100(1-a)** che rappresenta la percentuale di confidenza è anche chiamata *livello di confidenza*
- **(1-a)** si chiama *coefficiente di confidenza*

La qualità della misura è quindi legata al valore di α che indica la probabilità che i valori che noi misuriamo cadono dentro l'intervallo di confidenza . Per calcolare l'intervallo di confidenza non è necessario eseguire un gran numero di sequenze di misura, ma è possibile fare riferimento ad un singolo campione (X1,X2, ... Xn) purché le singole osservazioni siano indipendenti e provengano dalla stessa popolazione. Il fatto che le misure siano indipendenti è molto importante, ed è legato al modo con cui viene generato il workload . In particolare si deve considerare un workload che sia rappresentativo del carico reale e che sia costituito da campioni tra di loro scorrelati, cioè indipendenti, in modo da ottenere una distribuzione di tipo normale; se c'è una qualche correlazione tra i valori del workload in ingresso sicuramente ci sarà una correlazione tra i valori in uscita e la distribuzione che otteniamo non è una distribuzione normale.
Sotto questa ipotesi, è possibile utilizzare il "teorema del limite centrale" il quale stabilisce che se le osservazioni in un campione (X1,X2, ... Xn) sono indipendenti e provengono dalla stessa popolazione che ha una media μ ed una deviazione standard σ, allora il valor medio di un campione sufficientemente ampio è distribuito in modo normale, con valore medio μ e deviazione standard $\sigma/\sqrt{n}$

La deviazione standard **s /Ön** del valor medio è chiamata errore standard. Va notato che l'errore standard è completamente diverso dalla deviazione standard dei singoli componenti della popolazione (il primo rappresenta la deviazione standard dei diversi valor medi; la seconda, la deviazione standard dei singoli campioni X1, X2, ...Xn).

In tal caso, fissato un livello di confidenza di valore 100 $(1-\alpha)$%, l'intervallo di confidenza per la popolazione è dato da:

$$C_1 = \overline{x} - Z_p \cdot \frac{s}{\sqrt{n}}$$

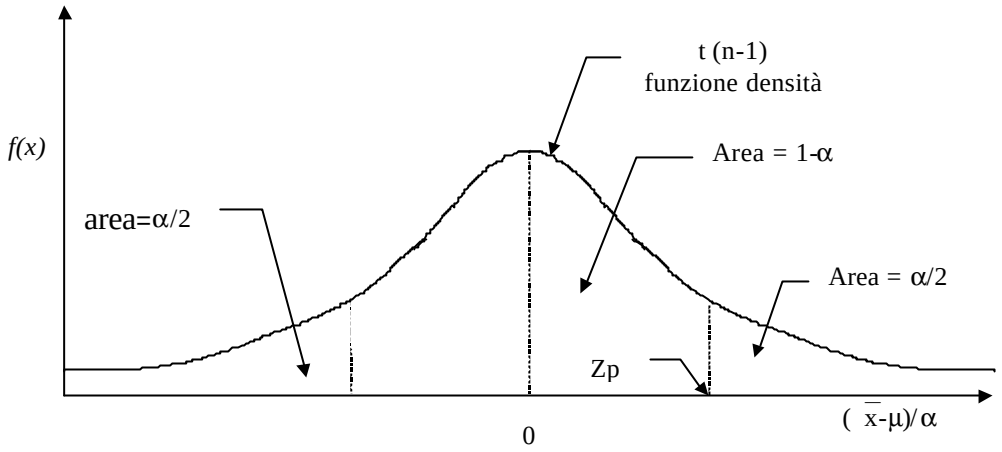
mentre

$$C_2 = \overline{x} + Z_p \cdot \frac{s}{\sqrt{n}}$$

e quindi

$$\left(\overline{x} - Z_p \cdot \frac{s}{\sqrt{n}}, \overline{x} + Z_p \cdot \frac{s}{\sqrt{n}} \right)$$

dove $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ e s = deviazione standard dei singoli campioni.



Se tutta l'area dei campioni (l'area sottesa dalla curva in figura) è uguale a 1, l'area all'interno delle linee tratteggiate vale (1-α). Z_p è un valore che permette di separare la parte di campioni che cadono all'interno della zona tratteggiata (quella che fissiamo) rispetto a quella tratteggiata. Esso è chiamato (1-α/2) quantile di una distribuzione normale unitaria.

Quando effettuiamo una valutazione di prestazioni, ricavata una statistica di valori X1, X2, ...Xn calcoliamo il valore medio $\bar{x}$ e la varianza s^2. A questo punto possiamo fissare la probabilità che i campioni siano contenuti in un certo intervallo, e quindi ricaviamo, mediante apposite tabelle disponibili in letteratura (una sintetica è riportata nel seguito), il valore di Z_p, e quindi l'intervallo di confidenza. In questa maniera per un singolo campione di misura possiamo calcolare tutte le informazioni che ci interessano sull'affidabilità della valutazione effettuata.

Livello di confidenza (%)	99,73	99	98	96	95,45	95	90	80	68,27	50
Zp	3,00	2,58	2,33	2.05	2,00	1,96	1,645	1,28	1,00	0,674

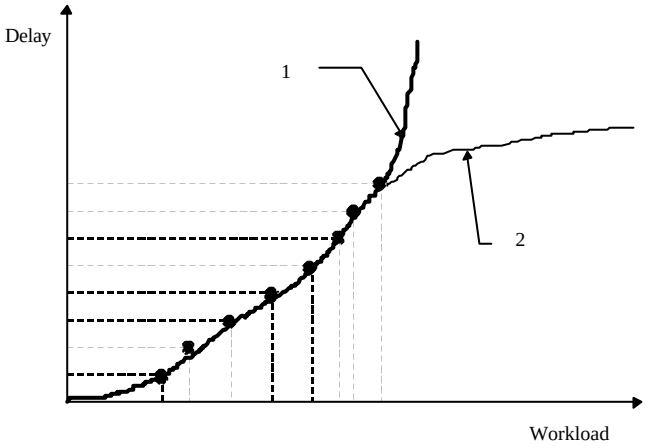
L'intervallo di confidenza è molto importante poiché costituisce un'informazione sulla qualità della valutazione che stiamo effettuando. Una valutazione fatta bene (sia che si tratti di simulazione, sia che si tratti di una misura) ha un intervallo di confidenza abbastanza ristretto, ed un livello di confidenza elevato.

6. Simulazione

La simulazione è utile sopratutto quando il sistema da valutare non è disponibile. Attraverso la simulazione è possibile confrontare alternative diverse con diversi workload ed ambienti. Una volta realizzato il modello del sistema da simulare, è possibile, cambiando il workload e quindi le condizioni al contorno, eseguire diverse valutazioni sullo stesso sistema. Ciò rappresenta un grosso vantaggio della simulazione, che la rende preferibile, nell'investigare le performances di un sistema, anche disponendo del sistema reale. Si pensi ad esempio ad una rete Ethernet in cui si vuole valutare il comportamento di un nodo, al variare del numero dei nodi, dell'entità di disturbi e della dimensione fisica della rete. Realizzare l'ambiente desiderato, in un sistema reale è molto complesso (a volte impossibile), costoso e richiede molto tempo. Con un simulatore, implementare le diverse condizioni al contorno richiede solo la configurazione di un certo numero di parametri.

Va comunque ricordato quello che è il grosso rischio della simulazione: *l'errore è sempre in agguato.* Occorre sempre verificare con molta attenzione il modello realizzato e *non fidarsi mai ciecamente dei risultati ottenuti.*

Un altro dei vantaggi che abbiamo utilizzando una simulazione è che, alla fine della stessa, è possibile ottenere delle curve che forniscono una rappresentazione visiva dell'oggetto della simulazione (rappresentano ad esempio il tempo di ritardo di un sistema, il tempo di trasmissione, il tempo di attesa in coda, etc). Supponiamo di voler ottenere una curva che ci rappresenta il ritardo al variare del carico: è necessario eseguire, attraverso il simulatore, una serie di misure. Queste vengono effettuate variando il workload; per ogni workload otteniamo un certo ritardo (Delay); interpolando i punti ottenuti in un grafico possiamo tracciare la curva. A titolo di esempio, la figura mostra un possibile risultato:



Va sottolineato come ogni singolo punto della curva non è, ovviamente, il risultato di una singola misura, bensì il risultato di una statistica. Ritornando all'esempio considerato, per ogni valore del workload il tempo di ritardo è misurato calcolando la media dei tempi di ritardo di un certo numero di pacchetti (10.000 - 100.000) spediti durante la simulazione. E' importante avere già a priori un'idea del risultato che si vuole ottenere. Se stiamo misurando, ad esempio, un tempo di ritardo, la curva che ci dovremmo aspettare è la (1); non può un tempo di ritardo, in generale, avere un andamento tipo quello descritto dalla curva (2). Quando si esaminano i risultati ottenuti attraverso la simulazione, occorre sempre ricordare che i risultati ottenuti possono essere errati, per cui vanno sempre analizzati con spirito critico. E' pericoloso accettare per fede i risultati di una simulazione, specie se non riusciamo a giustificarli logicamente.

È facile commettere errori, (il simulatore può funzionare bene, ad esempio, per piccoli carichi e generare risultati scorretti quando il sistema va in saturazione, oppure può essere errato il modello di simulazione). Per evitare gli errori, o ridurli, occorre avere una certa preparazione per ciò che riguarda statistica e capacità nello sviluppo di software, oltre ad una approfondita conoscenza del problema da valutare. Vediamo quali possono essere le principali cause di errore:

- **Livello di dettaglio inappropriato**: quando si esegue una simulazione bisogna prima avere chiaro cosa si vuole ottenere da questa, e solo in un secondo tempo farne il modello. Un modello permette di eseguire alcuni tipi di misure, ma non tutti, quindi è necessario scegliere un modello opportuno di volta in volta. Nel caso delle reti, ad esempio, dopo aver studiato a fondo il protocollo di comunicazione, occorre prima decidere che tipi di problemi vogliamo analizzare, poi capire che tipo di misure dobbiamo fare, ed infine generiamo il modello adatto. Il modello può essere molto semplice se vogliamo ricavare delle valutazioni abbastanza generali, mentre se, ad esempio, vogliamo evidenziare il comportamento di una stazione che genera un certo tipo di traffico, rispetto ad una rete che ha un traffico di tipo diverso, è necessario creare un modello più complesso che tiene conto delle particolarità delle stazione rispetto alla rete (ad esempio una stazione che generare traffico di tipo vocale in una rete con traffico dati). Stabilire il livello di dettaglio della valutazione è fondamentale perchè influenza moltissimo il lavoro da svolgere.

- **Linguaggio inadatto**: possono essere usati linguaggi di tipo "general purpouse" (ad es. C, Pascal, Java) oppure linguaggi specifici (ad es. SIMULA). I linguaggi general purpouse permettono di rappresentare dettagliatamente tutti gli aspetti del problema in esame, però non hanno al loro interno delle strutture già predisposte per la simulazione. Ciò significa che il programma di simulazione dovrà essere costruito per intero, partendo dalle istruzioni offerte dal linguaggio prescelto. I linguaggi di simulazione invece hanno delle strutture predisposte per la simulazione, però rispetto ai linguaggi general purpouse sono meno flessibili. Pertanto, se spesso semplificano il lavoro, quando si devono simulare sistemi con caratteristiche particolari, possono presentare limitazioni che richiedono la messa a punto di modelli complicati e poco efficienti .
- **Modello non valido**: se il modello non descrive correttamente il funzionamento del sistema è chiaro che la valutazione darà risultati inesatti. È molto importante allora una fase preliminare di studio. Se, ad esempio, vogliamo eseguire una simulazione su un protocollo di tipo Token Ring, è necessario innanzi tutto studiare il protocollo di accesso al mezzo fisico; non tutto il protocollo, ma almeno la parte relativa ai punti che stiamo investigando attraverso la simulazione.
- **Condizioni iniziali errate**: quando si manda in esecuzione una simulazione occorre inserire delle condizioni iniziali per definire lo stato del sistema. Se, ad esempio, eseguiamo una simulazione basandoci su un modello che fa uso delle reti di Petri è necessario all'inizio stabilire esattamente la posizione delle marche. Un eventuale errore comporterebbe dei risultati errati pur partendo da un modello corretto.
- **Tempo di simulazione troppo breve**: se vogliamo simulare il comportamento di un sistema e le condizioni sono variabili, come normalmente avviene, i valori che si ottengono non sono rappresentativi del sistema. Vi sono delle tecniche che consentono di capire qual è il tempo di simulazione minimo necessario, in ogni caso però occorre scegliere un tempo sufficientemente lungo da assicurare un numero sufficientemente elevato di campioni (diciamo almeno qualche migliaio per rendere significativa la statistica) . Nel caso di una rete di calcolatori, per esempio, alcuni secondi possono costituire un buon valore di tempo di simulazione, dato che in tale intervallo di tempo girano migliaia o centinaia di migliaia

di messaggi. Per la simulazione di sistemi lenti, il tempo di simulazione può dover essere anche molto più lungo.

6.1 Tipi di simulazione

Sono tre i tipi di simulazione che vengono utilizzati:

6.1.1 Montecarlo Simulation

La simulazione Montecarlo è una simulazione statica, manca infatti l'asse dei tempi. Viene usata per analizzare dei fenomeni di tipo probabilistico e tempo invarianti. È utile nel caso si debbano analizzare sistemi che sono descrivibili analiticamente

6.1.2 Trace Driven Simulation

In questo tipo di simulazione usiamo come ingresso una *trace* di eventi di un sistema reale per ottenere una *trace* di uscita. Una *trace* è un file che descrive una serie di dati.

- Vantaggi: credibilità, workload accurato, facilità di confronto, somiglianza con implementazioni reali. Il workload che viene fornito in ingresso è molto accurato perché è descritto campione per campione; ciò permette di eseguire delle simulazioni diverse utilizzando sempre lo stesso trace e quindi di confrontare sistemi diversi nelle identiche condizioni operative.
- Svantaggi: finitezza del campione, eccessivo dettaglio (sia del campione che del modello).

6.1.3 Discrete Event Simulation

Si implementa un modello a stati discreti del sistema, cioè un modello in cui lo stato del sistema evolve a scatti anziché in modo continuo. In questo modo possiamo far avanzare lo stato del sistema in funzione degli eventi che ci interessa mettere in risalto. (Se vogliamo, ad esempio, analizzare una rete di calcolatori; ogni pacchetto che viene generato rappresenta un evento che influenza il sistema, il quale evolve attraverso un trigger continuo da parte di questi eventi, che sono tempo discreti).
Il tempo cui riferiamo la simulazione può essere sia un tempo continuo che un tempo discreto. Normalmente viene utilizzato un tempo discreto, perché utilizzare un tempo continuo richiederebbe un livello di granularità nelle operazioni che modellano l'evoluzione del sistema troppo fine, che consuma tempo di calcolo inutilmente. Descritto il sistema, possiamo, ad esempio, fare un analisi in cui la risoluzione del tempo sia di un microsecondo: si fa avanzare il tempo di un microsecondo per volta e si osserva ciò che succede nel sistema. In questo modo di operare il tempo viene considerato continuo con risoluzione di un microsecondo; ciò significa che ogni microsecondo occorre verificare lo stato di tutte le variabili sotto osservazione (e ciò richiede tempo) anche se queste variabili non sono cambiate rispetto al microsecondo precedente. Se invece si fa avanzare il tempo a scatti, incrementandolo della differenza di tempo fra due eventi successivi, allora la simulazione viene eseguita più velocemente e la misura che si ottiene è ugualmente rigorosa perché non si perdono eventi e non si approssimano i tempi. In questo caso, il tempo viene considerato discreto. Nel primo caso per ogni avanzamento del tempo si controlla se si è verificato un qualche evento, nel secondo ogni volta che si verifica un evento si incrementa il tempo di una certa quantità.

Vediamo quali sono i componenti del simulatore a eventi discreti.

- **Event Scheduler**: contiene la lista di tutti gli eventi che devono accadere, o meglio, a seconda dell'evento che si verifica, esamina la lista e decide l'operazione che deve essere eseguita. È uno degli elementi più usati della simulazione. Per tale motivo è importante che questo modulo sia implementato in modo efficiente, altrimenti si avrebbe un rallentamento complessivo nella simulazione.
- **Simulation clock & time advancing mechanism**: Il tempo globale del sistema è rappresentato come una variabile, e viene incrementato dallo schedulatore, ogni volte che si verifica un evento. I modelli dei sistemi che dobbiamo simulare sono spesso costituiti da eventi paralleli (fra loro indipendenti). Tenere conto, di tanti eventi tra di loro indipendenti, in un programma sequenziale può sembrare difficile, però non lo è se facciamo riferimento alla variabile tempo globale, per sincronizzare le attività del sistema. E' il verificarsi di un evento che fa incrementare la variabile che rappresenta il tempo.
- **System state variables**: sono tutte le variabili globali che descrivono lo stato del sistema. Possono essere suddivise in tre gruppi:
  - Input variables: rappresentano tutte le variabili connesse ad eventi esterni al sistema. Ad es. in una rete di calcolatori, i messaggi da trasmettere, la loro lunghezza, i disturbi,ecc.
  - Internal variables: rappresentano tutte le variabili che modellano lo stato interno di un sistema. Ad es. la lunghezza delle code nei nodi, il numero di nodi, il numero di linee fisiche, ecc.
  - Output variables: rappresentano tutte le variabili necessarie per derivare i parametri prestazionali desiderati. Queste variabili sono disponibili per il Report Generator evanno definite in funzioni del tipo di misure che si desidera effettuare.
- **Event routines**: aggiornano le variabili di stato e schedulano altri eventi. Sono ad es. le funzioni che generano nuove richieste di messaggi da trasmettere (secondo una definita frequenza e distribuzione) e quindi forniscono il workload alla rete. Ogni volta che una Event routine genera un nuovo messaggio, questo viene accodato e diviene disponibile per la trasmissione.
- **Input routines**: acquisiscono i parametri del modello dall'utente. Permettono all'utente di introdurre ad es. il numero di nodi della rete, il Bit rate, la probabilità di errore, il valore del workload, ecc. Inoltre, nel caso di simulatori basati su Reti di Petri, permettono anche di introdurre il modello della PN, sia attraverso un'interfaccia grafica, quando didponibile, che attraverso un'interfaccia testuale.
- **Initialization routines**: settano il valore iniziale delle variabili di stato. Ad es. nel caso di Reti di Petri, permettono di introdurre la marcatura iniziale.
- **Report generator**: calcola i risultati finali e li presenta in modo opportuno. Possiamo pensare, ad esempio, di generare un file in cui sono memorizzati i valori delle variabili che stiamo misurando, dhe possono poi essere utilizzate da un opportuno programma di visualizzazione grafica.
- **Main program**: lega tutte le routines insieme.

Se invece di realizzare un simulatore si fa uso di un tool di simulazione il lavoro è molto più semplice, nel senso che l'ossatura del sistema è già pronta ed occorre solo realizzare un modello che possa essere utilizzato dal tool. E' chiaro che un tool di simulazione è più limitativo rispetto al generare un proprio programma di simulazione, perché in genere un tool di simulazione offre certe funzioni tipiche che non possono in alcun modo essere modificate. I tools di simulazione di uso più comune sono quelli basati sulle Reti di Petri. Ne esistono diversi tipi, ma fondamentalmente possono essere divisi in due categorie:

- Quelli basati sulle Reti di Petri Stocastiche. Un tool di simulazione di questo tipo è il *GSPN* attraverso cui è possibile modellare in maniera molto compatta sisiemi anche molto complessi. Ad esempio,

normalmente un'intera rete di calcolatori viene modellata con una singola rete di Petri. Il vantaggio di un tool di questo tipo è che si ottengono dei risultati abbastanza generali, simili a quelli ottenibili col modello analitico. Il grosso svantaggio è che non è possibile eseguire misure che non siano "standard". Infatti è possibile, di norma, eseguire due tipi di misure, sul troughput e sul tempo di ritardo. Se ad esempio vogliamo determinare la distribuzione del tempo di ritardo con il GSPN non è possibile farlo. Si può solo ottenere il valore medio del tempo di ritardo. Questa informazione è sicuramente significativa, ma non è esaustiva; se ad esempio stiamo valutando una rete di calcolatori per applicazione tempo critiche, ci interessa conoscere pure il tempo massimo di ritardo. È altresi difficile simulare il transitorio di un sistema, per vedere, ad esempio, cosa succede quando ad un certo istante viene introdotto un burst di traffico (un picco di carico) e come questo si smorza attraverso la rete.

- Quelli basati su Reti di Petri Generalizzate. Come noto, con questo nome intendiamo quelle classi di Reti di Petri che contengono diverse funzionalità aggiuntive, rispetto a quelle fornite dalle Reti di Petri Standard. Per esempio, la presenza di Procedure associate allo scatto delle transizioni. In questo modo è possibile rappresentare, in forma algoritmica, aspetti del nostro sistema non modellabili mediante una Rete di Petri o che richiederebbero modelli molto complessi. Inoltre, mediante le procedure, è possibile interagire col sistema in modo da derivare parametri prestazionali, altrimenti non misurabili.

7. Analisi dei risultati

Durante lo sviluppo di un modello di simulazione occorre garantire che esso sia rappresentativo del sistema reale (validazione del modello) e che esso sia implementato in modo corretto (verifica del modello). La validazione del modello è legata alla correttezza delle assunzioni fatte sul comportamento del sistema. La verifica (debugging sul modello implementato) è relativa alla correttezza della sua implementazione. Sono possibili tutti e quattro i casi:

- valido, verificato
- valido, non verificato
- non valido, verificato
- non valido, non verificato

Dopo lo sviluppo del modello occorre decidere quante osservazioni iniziali devono essere scaricate prima che il modello sia in uno stato stazionario, e quanto tempo debba durare una simulazione. Questi sono due punti cruciali nell'esecuzione di una simulazione: infatti, durante il transitorio iniziale, prima che il sistema si porti a regime, si ottengono dati che in genere non sono significativi poichè di norma viene valutato il comportamento dei sistemi viene valutato in condizioni stazionarie. E' pertanto indispensabile capire quando il transitorio è terminato. E' inoltre importante comprendere quanto tempo debba durare la simulazione, in modo da collezionare una quantità adeguata di dati, da analizzare.
Ovvero, bisogna affrontare i seguenti due problemi:

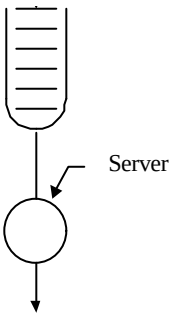
- **Transient removal**: rimozione dei transienti.
- **Stopping criterion**: criteri per bloccare la simulazione.

7.1. Tecniche di validazione del modello

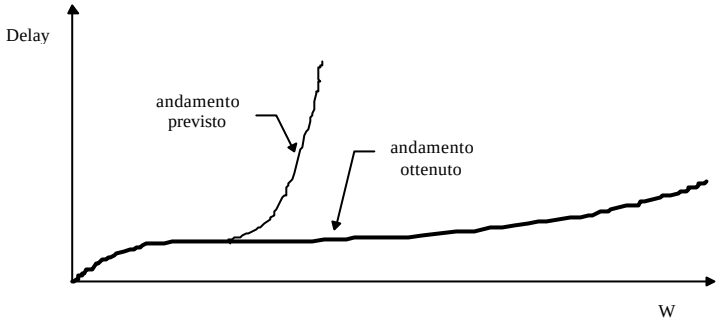
La validazione intende assicurare che le assunzioni usate per il modello siano ragionevoli e che se correttamente implementate forniscano risultati simili a quelli del sistema reale. Tre aspetti chiave vanno validati:

- **Assunzioni**: dato un sistema occorre capire come funziona il sistema, cioè capire quali sono le caratteristiche del sistema che devono essere implementate, trascurando quelle caratteristiche che fanno parte del sistema ma che sono irrilevanti ai fini di ciò che si vuole simulare.
- **Valori dei parametri di ingresso e distribuzioni**: dobbiamo cioè valutare le caratteristiche del sistema che si sta simulando con determinati valori del workload. Occorre, in altre parole, fornire al sistema, un workload significativo in riferimento alle reali applicazioni del sistema stesso. Per far questo può essere utile effettuare inizialmente dei calcoli grossolani che consentono di stabilire il range del workload. La distribuzione del carico deve essere descrittiva del tipo di sistema che si sta valutando (a seconda se stiamo simulando un sistema per il controllo di processo un sistema per trasmissione di dati, etc). Se, ad esempio, la rete è costituita da dieci nodi e lavora a 1 Mbit/s, nell'ipotesi che i messaggi siano di lunghezza fissa 1000 bit il workload che manda in saturazione il sistema può essere calcolato approssimativamente in 1000 messaggi al secondo. Poichè i nodi sono dieci, per mandare in saturazione il canale ogni nodo deve spedire 100 messaggi (stiamo supponendo un carico distribuito in maniera uniforme). Quindi 100 messaggi al secondo, in media, portano verso la saturazione del canale, valori molto più bassi implicano che il canale è scarico o debolmente caricato, valori più alti significano canale oltre la saturazione.
- **Valori d'uscita e conclusioni**. Alla fine della simulazione, occorre analizzare i valori di uscita e trarre da questi le giuste conclusioni. Nell'analizzare i risultati ottenuti, occorre ricordare che se stiamo effettuando una valutazione di prestazioni, in condizione stazionarie, occorre essere certi di trovarsi in tali condizioni prima di giustificare i risultati ottenuti. Un punto critico è il comportamento del sistema quando si è in condizioni di saturazione. In tale condizione, il sistema non riesce ad andare a regime, ed i risultati ottenuti non sono significativi o vanno analizzati con molta attenzione.

Approfondiamo questo punto che è particolarmente importante. Nel fare un modello di simulazione ogni nodo viene schematizzato come una coda (o eventualmente più code se abbiamo diversi tipi di traffico o diverse priorità). Se il canale non è saturo normalmente il server ha un tempo di servizio che è più breve del tempo con cui arrivano i messaggi per cui, in tali condizioni, la coda è vuota o parzialmente piena. In queste condizioni è possibile calcolare correttamente i tempi di ritardo della rete.



Quando però forniamo al nodo un carico talmente elevato che il sistema va in saturazione, il tempo di servizio del server risulta più grande del tempo di arrivo dei messaggi. In questa condizione la coda si riempie. Questa è una condizione critica per il simulatore; infatti tutti i messaggi che vengono generati, essendo la coda già piena, vengono scartati (a meno che la coda non abbia lunghezza infinita). Questo fa si che i risultati ottenuti siano falsati. In tali condizioni si possono ottenere curve di ritardo con l'andamento mostrato in figura:



Come si vede, all'aumentare del workload, dapprima il delay si mantiene costante, poi quando si raggiunge la saturazione del canale trasmissivo ci si aspetta che il delay cresca rapidamente (andamento atteso). La curva che invece si ottiene, vede crescere il delay molto lentamente (andamento ottenuto). Questo comportamento è legato al modo in cui è calcolato il ritardo, facendo la differenza fra l'istante di generazione dei messaggi e quello di consegna a destinazione.

Poichè in condizioni di saturazione vengono trasmesso solo pochi messaggi (la maggior parte resta in coda), la statistica viene effettuata su pochi campioni, molti dei quali sono relativi al transitorio iniziale, quando la rete non era ancora congestionata. In tali condizioni il tempo medio di attesa in coda non tiene conto del fatto che ci sono code piene di messaggi che non si riescono a trasmettere.

Per mettere in evidenza quest'ultimo aspetto occorre aumentare il tempo di simulazione lasciano invariato il carico, così facendo la statistica viene influenzata meno dai primi messaggi che in effetti fanno parte del transitorio. In condizioni di saturazione, poichè il calcolo del tempo di ritardo non è molto affidabile, conviene fare riferimento ad altri tipi di performance metrics, quali ad esempio il numero di elementi in coda alla fine della simulazione o il calcolo della pendenza con cui le code crescono nel tempo.

Per eseguire un ulteriore test sulla validità dei risultati ottenuti si può eseguire un confronto con:

- **Intuizione di esperti**: magari attraverso brainstorming meeting. Riunirsi attorno a un tavolo e discutere i risultati può essere molto utile per evidenziare errori o aspetti particolari di un sistema.
- **Misura sui sistemi reali:** Il confronto fra i risultati ottenuti mediante simulazione e quelli forniti da un sistema reale nelle stesse condizioni, costituisce un efficace strumento per validare la simulazione stessa.
- **Risultati teorici:** Anche il confronto con un modello analitico, pur se semplificato, può costituire un utile strumento di validazione dei risultati ottenuti.

3.1.1.1 Tecniche di verifica del modello

Due importanti tecniche per sviluppare, fare il debug, e mantenere programmi di simulazione sono:

- **Modular design**: richiede che il modello sia strutturato in moduli che comunicano fra loro attraverso opportune interfacce.
- **Top-down design**: consiste nello sviluppare una struttura gerarchica del modello in modo che il problema sia ricorsivamente diviso in un set di sottoproblemi.

Altre tecniche di verifica sono:

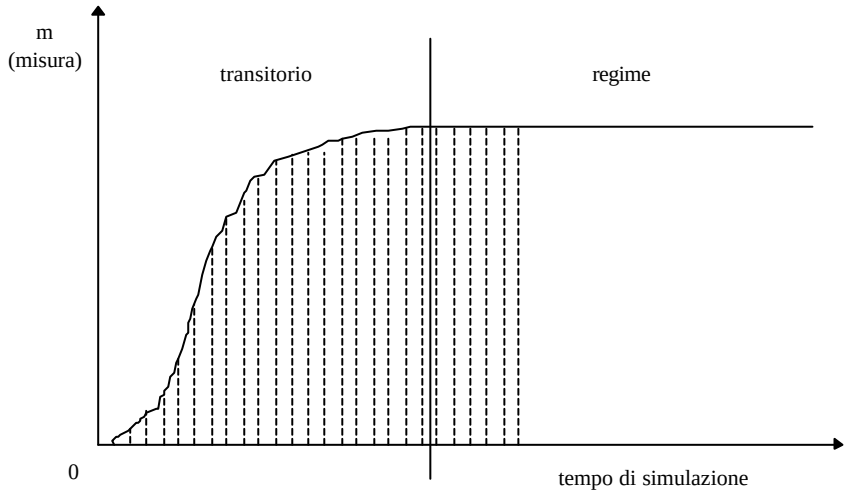
- **Antibugging**: consiste nell'includere checks ed outputs addizionali nel simulatore, che evidenziano gli errori.

- **Modelli deterministici**: usando parametri deterministici è facile fare un debug del modello che andrà poi eseguito con i parametri corretti. Se generiamo un carico fisso deterministico possiamo calcolare in maniera deterministica tutte le grandezze che ci interessano, confrontando tali grandezze con quelle ottenute attraverso la simulazione verifichiamo la correttezza del modello.
- **Eseguire dei casi semplificati**
- **Continuity test**: forti variazioni dell'uscita per piccole variazioni degli ingressi, normalmente sono sospette e vanno osservate con molta attenzione.

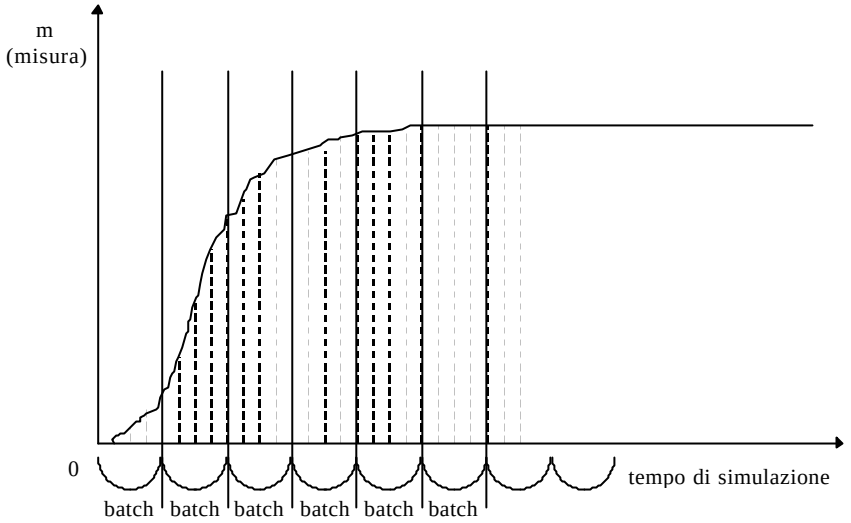
3.1.1.2 Eliminazione dei transienti

Nella maggior parte dei casi in una simulazione interessa solo la *Steady-state performance* (Performance nello stato stazionario). Pertanto i risultati della parte iniziale della simulazione (transitorio iniziale) vanno rimossi. Le strategie che possono essere utilizzate sono le seguenti:

- **Long runs**: la simulazione dura talmente a lungo che le condizioni iniziali non influenzano i risultati. È la tecnica più semplice, però presenta due svantaggi: 1) spreco di risorse nel senso che, per mettersi in condizioni di sicurezza spesso si fa durare la simulazione più a lungo del tempo strettamente necessario; 2) non è sicuro che il run sia abbastanza lungo, per cui può capitare che i valori collezionati nel transitorio iniziale influenziono pesantemente i risultati.
- **Proper initialization**: consiste nel fare iniziare la simulazione in uno stato vicino a quello stazionario. Ciò naturalmente riduce il transitorio. Il problema è determinare lo stato vicino a quello stazionario, da cui fare iniziare la simulazione.
- **Initial data deletion**: durante la fase stazionaria, la media non cambia molto, anche eliminando alcuni campioni. Si eliminano valori iniziali fin quando la loro eliminazione provoca variazioni sensibili nella media.



- **Batch means**: una lunga simulazione è divisa in parti (batch) di uguali durata. Occorre studiare la varianza di questi batch in funzione della loro dimensione. Ciò permette di eliminare i batch relativi al transitorio iniziale, ed utilizzare nelle statistiche solo i dati relativi ai batch in condizioni stazionarie, mediando fra i vari batch.



Spesso la tecnica usata è la Long runs. Anche se come detto spreca risorse, è però facile da usare. . Per determinare il tempo di simulazione, cioè il tempo necessario affinché il sistema sia a regime, basta eseguire due simulazioni alle stesse condizioni ma con tempi diversi; se i risultati ottenuti sono simili allora vuol dire che siamo a regime, e si prende il tempo di simulazione inferiore fra i due.

3.1.2 Distribuzioni più usate

Uno dei problemi di base nella simulazione, è la generazione di numeri casuali da utilizzare per modellare eventi nel sistema simulato: generazione di messaggi di errore, ack, ecc.

Fra le varie distribuzioni con cui è possibile generare tali numeri, le più note sono:

- **Distribuzione di Bernoulli**: una variabile può assumere solo i valori X=0, X=1 che determinano fallimento o successo.

  P = probabilità di successo, 1-P = probabilità di fallimento.

La distribuzione di Bernoulli modella il verificarsi o meno di un evento. La distribuzione di Bernoulli viene utilizzata, ad esempio, per modellare la probabilità che un pacchetto una volta trasmesso venga rovinato dal rumore.

- **Beta distribution**: è usata per rappresentare variabili random, in genere comprese fra 0 ed 1, che modellano proporzioni. Ad esempio la frazione di pacchetti che richiedono la ritrasmissione. Tramite questa distribuzione è possibile modellare il comportamento di un intero sistema.
- **Distribuzione Binomiale**: Il numero di successi X in una sequenza di n tentativi di Bernoulli ha una distribuzione binomiale, adatta a modellare il numero di successi in una sequenza di n tentativi indipendenti. Ad esempio il numero di processori attivi in un sistema multiprocessore, il numero di bit disturbati in una frame, ecc.
- **Distribuzione esponenziale**: questa distribuzione è molto importante perché è quella che si utilizza per generare il carico. È la sola distribuzione continua *memoryless*, questo significa che ogni campione che noi generiamo non ha memoria del campione precedente. Viene usata per modellare il tempo fra due eventi successivi indipendenti (tipicamente per la generazione del traffico in una rete di calcolatori).