Stili dell’architettura e qualità del software

Ingegneria del Software B

Laura Semini

Scale up, scale out

Application scalability can be defined as the ability to increase the application throughput in proportion to the hardware that is being used to host the application. In other words, if an application is able to handle 100 users on a single CPU hardware, then the application should on be able to handle 200 users when the number of process are doubled.

Vertical scalability is adding more memory and CPUs to a single box, or scaling up. Vertical scalability or scaling up is well suited for database tier. Databases have the following needs:

- Large shared memory space,
- Many dependent threads,
- Tightly-coupled internal interconnect

Horizontal scalability is adding more boxes of similar memory and CPU, or scaling out. Scale out is ideal for web-tier, and has some of the following characteristics:

- Small non-shared memory space
- Many independent threads
- Loosely-coupled external interconnect [an important point]
- Possibly many OS's

N-tier

Disponibilità	**I server di ogni tier(ordine, fila) possono essere replicati, quindi se uno fallisce c'è solo una minor QoS.**
Fault tolerance	**Se un cliente sta comunicando con un server che fallisce, la maggior parte dei server reindirizza la richiesta a un server replicato in modo trasparente all'utente.**
Modificabilità	**Il disaccoppiamento e la coesione tipici di questa arch. favoriscono la modificabilità**
Performance (efficienza)	**Performance ok, ma da tenere sott'occhio: numero di threads paralleli su ogni server, velocità delle comunicazioni tra server, volume dati scambiato**
Scalabilità	**Basta replicare i server in ogni tier. Unico collo di bottiglia la base di dati**

Publish-subscribe

Disponibilità	**Si possono creare clusters di dispatcher**
Fault tolerance	**Si cerca un dispatcher replica**
Modificabilità	**Si possono aggiungere publisher e subscribers a piacere. Unica attenzione al formato dei messaggi.**
Performance (efficienza)	**Ok. Ma compromesso tra velocità e altri requisiti tipo affidabilità e/o sicurezza.**
Scalabilità	**Con un cluster di dispatchers si può gestire un volume molto elevato di messaggi.**

Comunicazione asincrona

Disponibilità	**Basta replicare le code**
Fault tolerance	**Se una coda fallisce, si cerca una replica**
Modificabilità	**Unico vincolo il formato dei messaggi**
Performance (efficienza)	**Si possono spedire migliaia di messaggi al secondo. Compromesso tra affidabilità/sicurezza e performance**
Scalabilità	**Le code possono essere ospitate presso origine e destinazione della comunicazione, ma anche da clusters grandi a piacere di servers.**

Coordinatore di processi

- Il Coord di processi conosce la sequenza di passi necessari per realizzare un processo aziendale (PA).
- Riceve la richiesta, chiama i servers secondo l'ordine prefissato, fornisce una risposta
- Normalmente usato per realizzare processi aziendali complessi
- Disaccoppiamento: i server non conoscono il loro ruolo nel PA complessivo né l'ordine dei passi del processo. I servers semplicemente definiscono un insieme di servizi
- Comunicazione flessibile: sincrona o asincrona.

Coordinatore di processi

Disponibilità	**Il coordinatore è un punto critico: deve essere replicato se si vuole garantire disponibilità.**
Fault tolerance	**Occorre specificare compensazione. Occorre ridirigere su un coordi8natore replica.**
Modificabilità	**Posso modificare liberamente i servers purché non cambino le funzionalità esportate.**
Performance (efficienza)	**Il coordinatore deve essere in grado di servire più richieste concorrenti. La performance del processo è limitata dal server più lento. Se non tutti i servers sono necesari, si usa un time-out.**
Scalabilità	**Replicando il coordinatore. Scala sia up che out.**

P2P

Disponibilità	**Dipende dal numero di nodi in rete, ma si assume si.**
Fault tolerance	**Gratis**
Modificabilità	**Si, se dell'architettura interessa solo la parte di comunicazione**
Performance (efficienza)	**Dipende dal numero di nodi connessi, dalla rete, dagli algoritmi. Per esempio BitTorrent ottimizza scaricando per primo il file/pezzo più raro.**
Scalabilità	**Gratis**

Pipes and Filters

Disponibilità	**Avendo "pezzi di ricambio" (componenti e possibilità di connetterle) sufficienti a ricostruire una catena.**
Fault tolerance	**Occorre riparare una catena interrotta usando componenti replica.**
Modificabilità	**Si, se le modifiche interessano una o comunque poche componenti**
Performance (efficienza)	**Dipende dalla capacità del canale di comunicazione e dalla performance del filtro più lento.**
Scalabilità	**Ok anche scale out.**