



## **BAS 2009**

*“Wizard CMS: dimensionamento ed evoluzione”*

Dimensionamento infrastruttura Wizard



## Sommario

Utilizzo delle risorse: creazione istanza .....	3
Utilizzo delle risorse: sviluppo e produzione.....	4

## Introduzione

Il Wizard Kaistar è una applicazione web capace di gestire in maniera automatica e trasparente la creazione e la configurazione di una web application basata sul CMS Kaistar. Tutto il processo prevede:

1. La creazione del database di riferimento e il rispettivo utente;
2. La creazione della CDA e della CMA;
3. Il deploy della webapp all'interno di un Apache Tomcat;
4. La creazione del virtual-host all'interno di Apache Httpd Server nella forma `cms.nomedominio.it/contextapplicazione`.

Come è facile intuire, il processo di creazione implica l'utilizzo di un certo numero di risorse di sistema, in particolare di RAM e di spazio su disco. Quantificare l'utilizzo di queste risorse per singola istanza è indispensabile per un corretto dimensionamento dell'architettura applicativa da dedicare al Wizard, calcolando, così in maniera opportuna, il limite superiore di istanze gestibili dal sistema configurato per ospitare le applicazioni del contratto BAS2009.

E' importante fare una premessa. L'ambiente del Wizard è concepito solo per lo sviluppo delle applicazioni e non per la loro messa in produzione. Di conseguenza gli unici utenti che accedono alle istanze sono gli stessi sviluppatori. Ciò implica un basso consumo di Ram da parte della JVM, rendendo la stima di occupazione media di memoria pressochè costante nel tempo e direttamente proporzionale al numero di elementi compilabili aggiunti in fase di sviluppo (es. pagine JSP). La stessa considerazione può essere fatta per l'occupazione di spazio su disco da parte dei database, i quali in fase sviluppo crescono, verosimilmente, soltanto in maniera funzionale alle necessità del progetto stesso.

E' possibile dividere i momenti di vita di un'istanza in tre fasi:

1. **Creazione di un nuova istanza:** la webapp è stata appena creata e le dimensioni dei vari componenti coincidono con quelle di una installazione base del Kaistar.
2. **Sviluppo:** l'applicazione viene personalizzata e l'accesso è limitato agli sviluppatori.
3. **Produzione:** l'applicazione è stata ultimata e pronta per essere migrata in un ambiente di produzione.

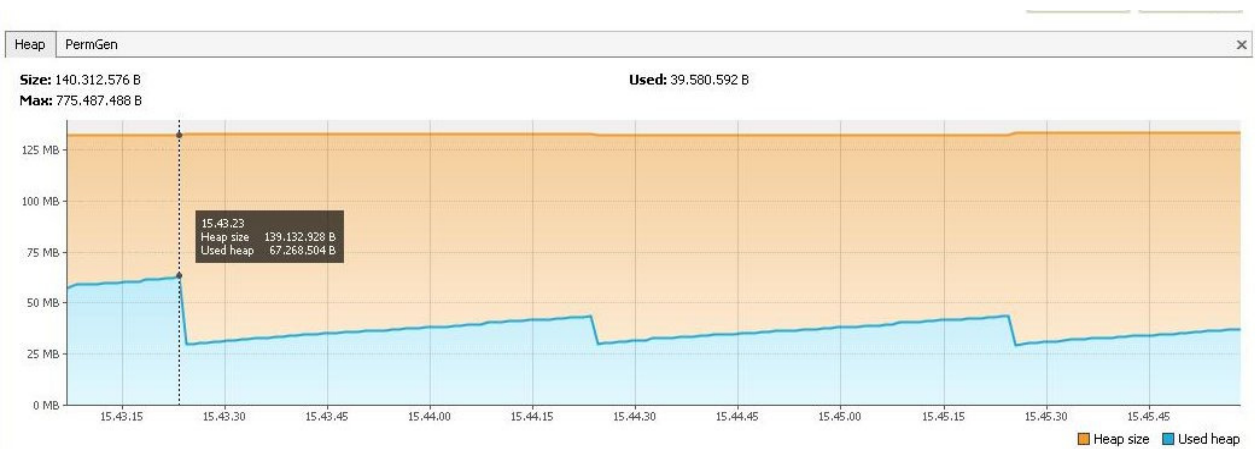
## Utilizzo delle risorse: creazione istanza

### Occupazione su disco per singola app:

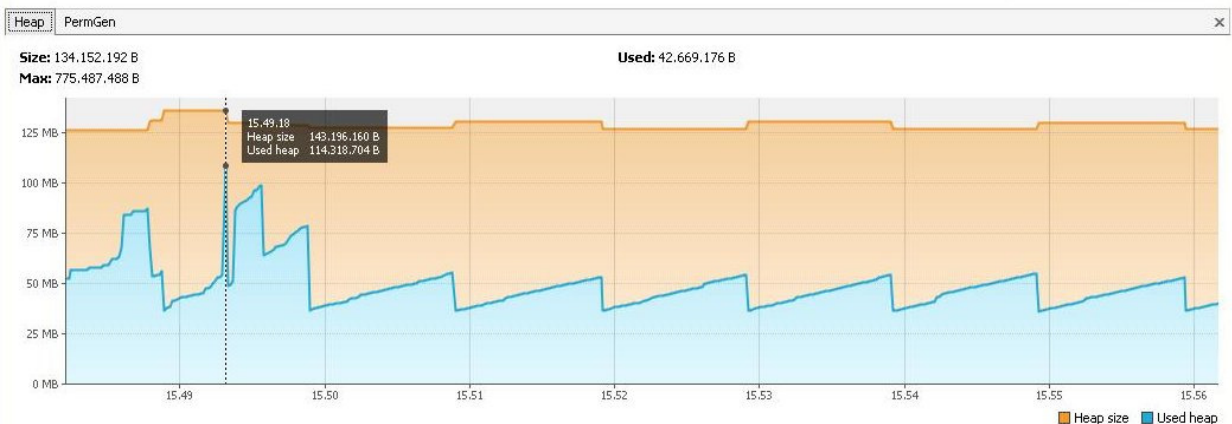
- Database: 7 MB
- Applicazione(CDA/CMA): 60 MB

### Occupazione RAM per singola app:

Nella figura 1 (riportata in basso) è mostrato il grafico della Heap Memory occupata da un Tomcat con una istanza d'applicazione non ancora acceduta. Il picco massimo di utilizzo della RAM è di circa 65MB, dovuto al primo deploy del manager dell'application server. In generale, il Tomcat nell'esempio impiega circa 80 MB di Ram [40MB Heap Memory + 40 MB Permanent Generation Memory] senza applicazioni compilate.



Nella figura 2 viene mostrato l'andamento della Heap con una sola applicazione ma con cinque utenti che vi accedono simultaneamente. Il picco di 114 MB e gli altri valori nel range 75/100 MB, sono frutto della prima compilazione della webapp. Ultimata questa fase, per i successivi accessi alle stesse pagine, l'application server recupera i compilati dalla sua cache su disco, evitando così di ricompilare le pagine e impedendo di fatto alla JVM di impegnare altra RAM. Guardando l'andamento successivo e confrontandolo con quello della figura 1, è possibile affermare che una singola istanza del Kaistar creata con il wizard, occupa nella misura massima, dopo la prima compilazione, circa 20MB di memoria [15MB Heap + 5MB Permanent Generation Memory].



## Utilizzo delle risorse: sviluppo e produzione

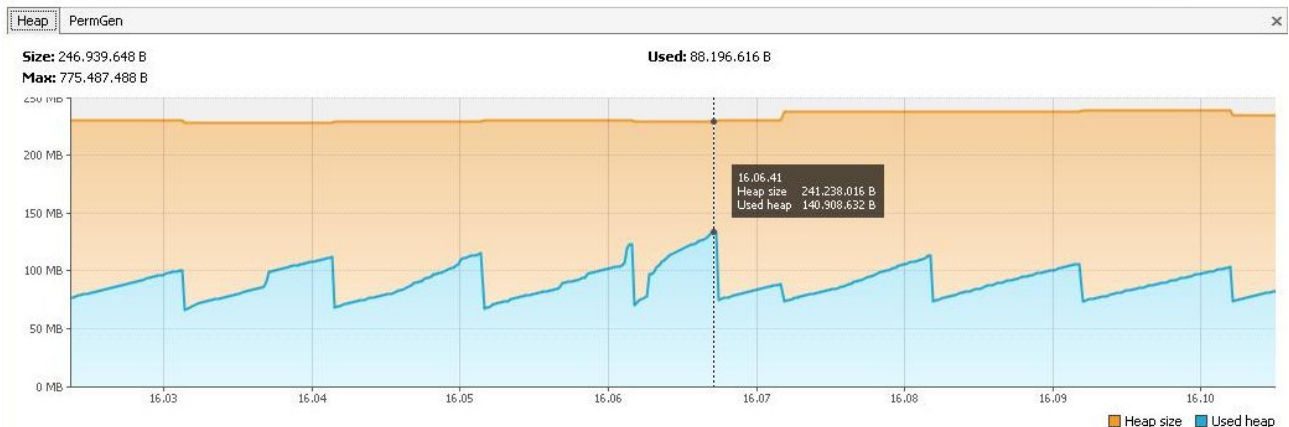
Una volta avviato lo sviluppo, l'utilizzo di risorse da parte di una webapp non può essere stabilito in maniera assoluta, dal momento che ogni progetto può avere le sue specifiche esigenze, determinando un maggiore o minore impiego di risorse. Pertanto l'analisi successiva è frutto di deduzioni empiriche, prendendo a modello alcune delle applicazioni più impegnative sviluppate finora con il wizard.

### Occupazione su disco per singola app (es. AmorePhilos):

- Database: 10 MB
- Applicazione(CDA/CMA/FILES): 85 MB

### Occupazione RAM per singola app:

Dalle osservazioni fatte nel paragrafo precedente, ci aspetteremmo che n applicazioni occupassero n\*20MB di ram. In realtà, dal momento che gli accessi alle applicazioni sono limitati ai soli sviluppatori, l'impiego di memoria è di gran lunga inferiore alle aspettative. Nel nostro esempio ci sono 19 applicazioni deployate sotto lo stesso Tomcat e cinque utenti che navigano tre web applications. Se dovessimo fidarci della stima per eccesso appena effettuata, dovremmo aspettarci che il sistema utilizzi mediamente 460 MB di memoria [20MB\*19 + 80 MB Tomcat ]. In realtà, in dieci minuti di navigazione, il picco massimo di HeapM occupata è stato di 140 MB e di 95MB per la PermanentG, per un totale di impiego massimo di 235MB di RAM: di fatto la metà dell'utilizzo stimato.



Resta tuttavia importante mantenere un limite superiore di impiego di memoria abbastanza alto per singola applicazione, così da non rischiare blocchi della JVM per la saturazione della memoria dedicata. Lo schema successivo può essere un ottimo riferimento per il dimensionamento dell'architettura ospitante il Wizard Kaistar.

## Utilizzo delle risorse: esempio

- Database: 40 MB
- Applicazione (CDA/CMA/FILES): 150 MB
- RAM per singola applicazione(CDA/CMA): 25MB
- Numero massimo di applicazioni per Tomcat:
  - ( RAM dedicata alla JVM[Heap + NonHeap] - utilizzo medio di memoria da parte del Tomcat privo di applicazioni ) / 25

**Il numero totali di applicazioni che è possibile ospitare su una singola istanza di Tomcat con 512 Mb di RAM ad esso dedicata è pari a 17,28 applicazioni.**