



UNIVERSITÀ
DI CAMERINO

Archiviazione dello stato della sessione di un'applicazione ASP.NET con Redis vs SQL Server

Studente

Toma Matteo – Mtr. 116781

Relatore

Fausto Marcantoni

Correlatore

Roberto Gagliardi

2. OBIETTIVI

Tecnologie come ASP.NET sono molto efficaci nel costruire pagine web, talvolta però è necessario poter salvare lo stato della sessione di un'applicazione per non perdere i dati.

Creazione dell'applicazione su ASP.NET



Memorizzazione dei dati dell'applicazione



Memorizzare lo stato della sessione dell'applicazione

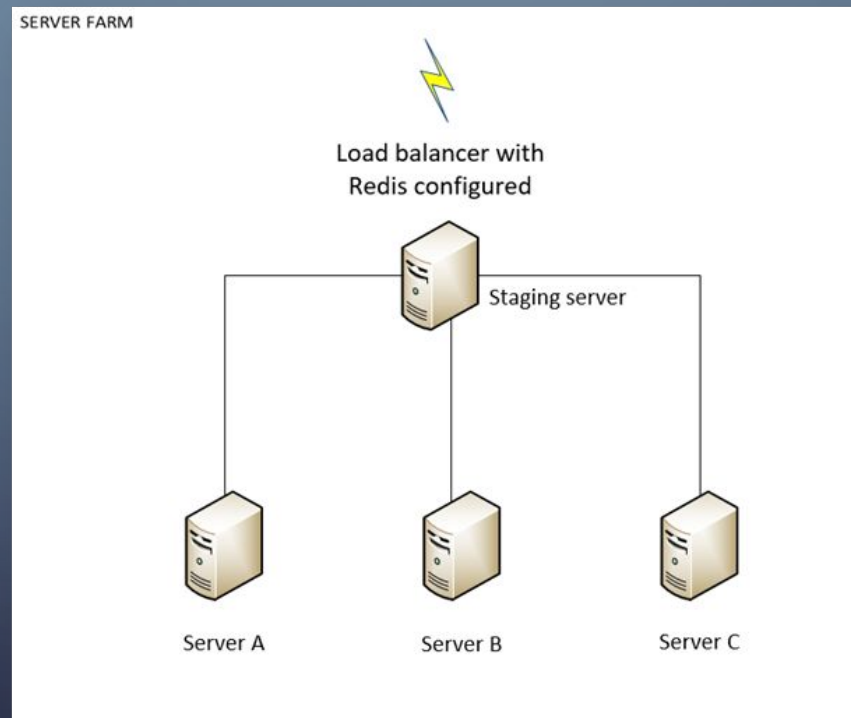


Testare le prestazioni di SQL Server e Redis

3. CREAZIONE DI UNA SERVER FARM

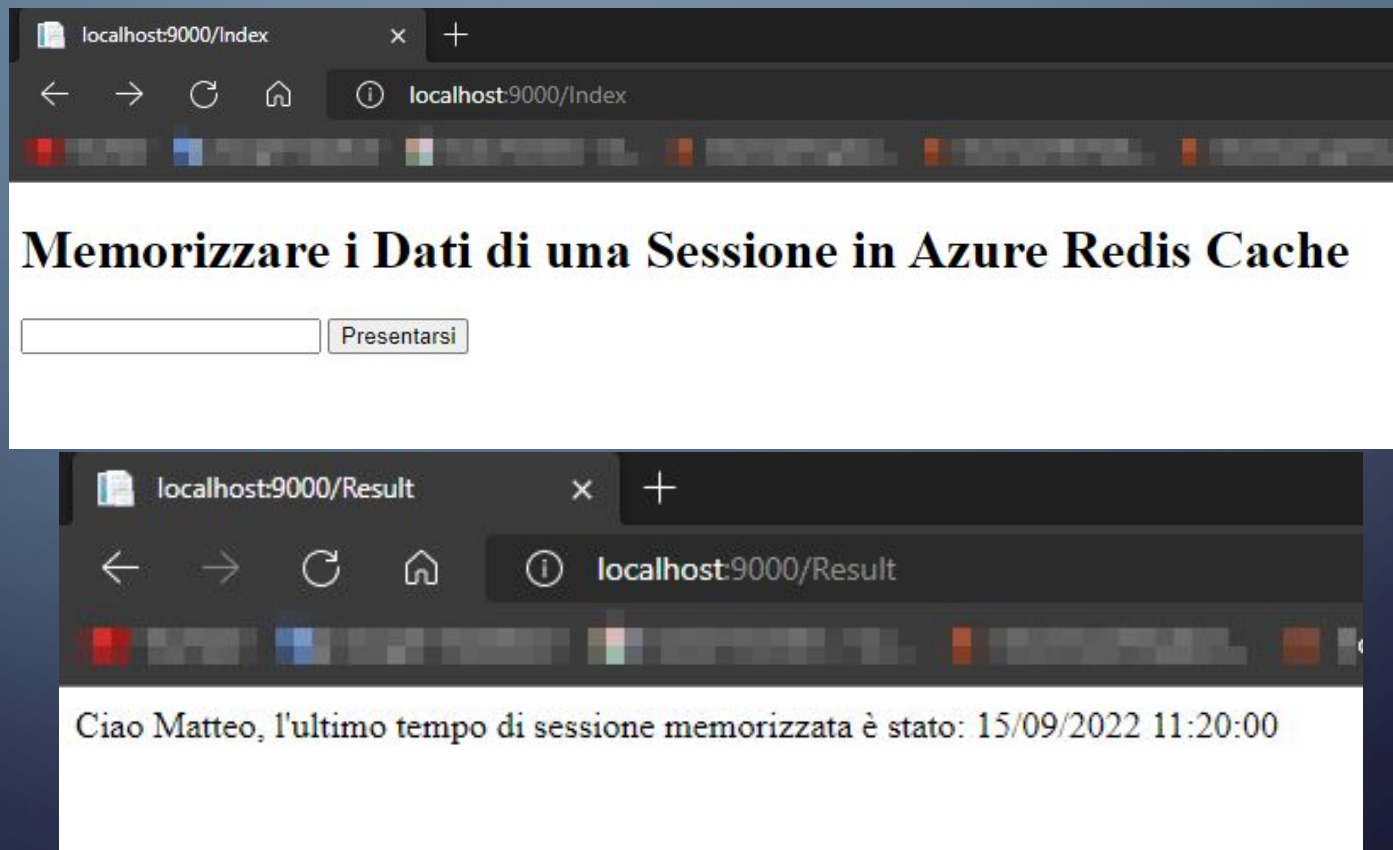
Per poter memorizzare lo stato di una sessione, e i dati dell'applicazione, serve una server farm. Si può utilizzare lo strumento IIS (Internet Information Services (IIS)) di Microsoft.

Struttura della server farm



4. Visualizzazione dell'applicazione

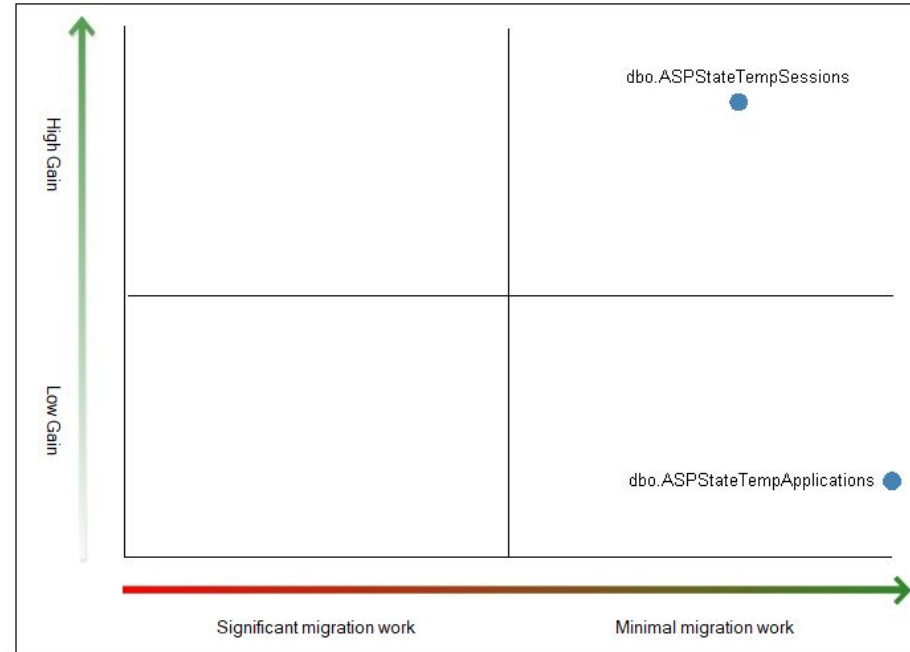
Aprendo ora un qualsiasi Web Browser e andando all'indirizzo locale dello staging server, cioè localhost:9000/Index, verrà mostrato il contenuto della pagina:



6.1 Test delle prestazioni con SQL SERVER

Ottimizzazione della memoria
nel carico di lavoro.

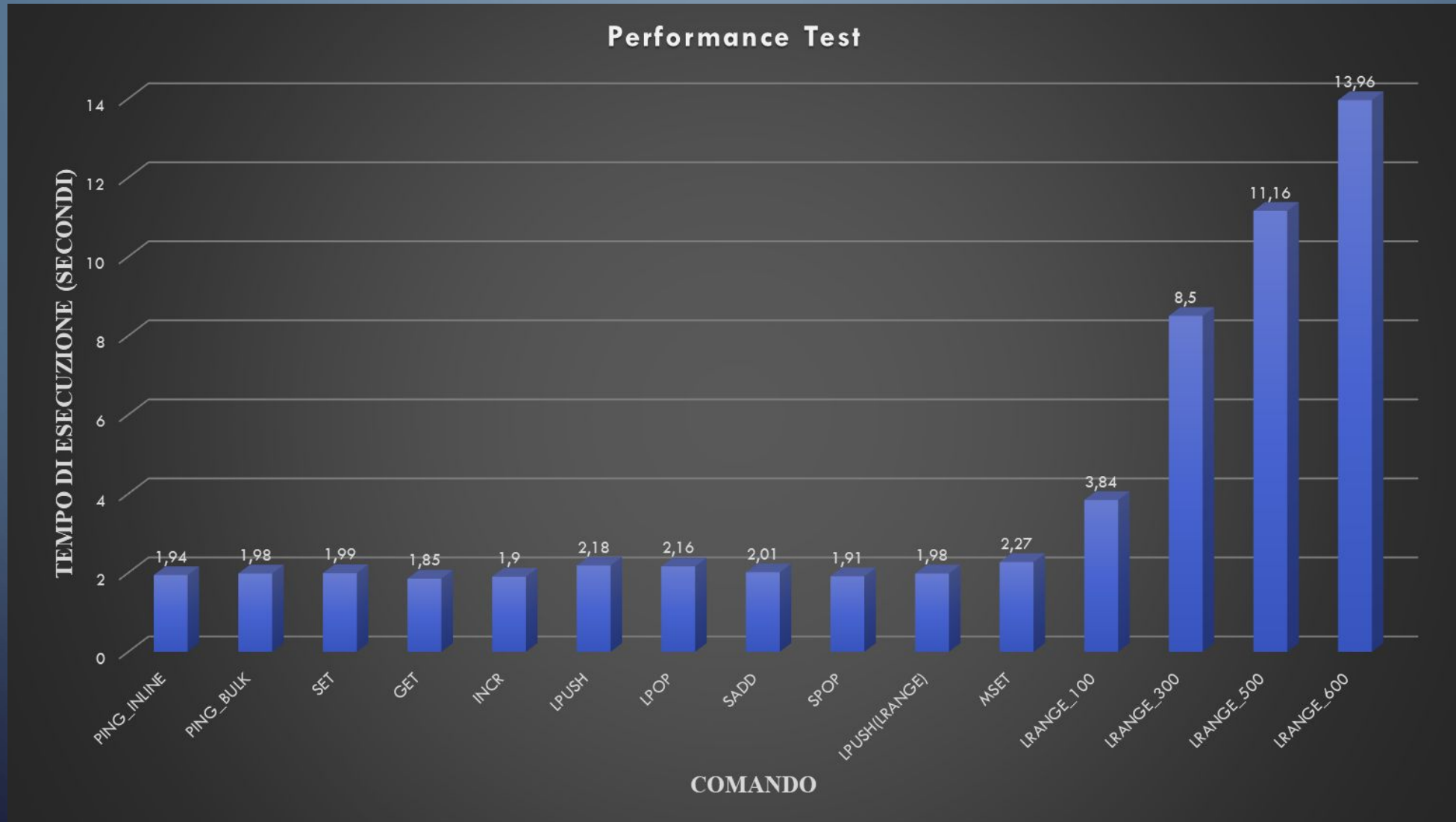
Select number of Tables:
5
10
15
20
25
30



Le tabelle in alto a
destra sono quelle a
cui bisognerebbe
dare la priorità per
ottimizzare l'uso
della memoria.

Sforzo decrescente di ottimizzazione della memoria

5. Test delle prestazioni con Redis



5.1 Qual è il provider più prestante

Anche in letteratura, sono state studiate le performance di altre tecnologie, aumentando il numero degli oggetti nell'applicazione i risultati sono i seguenti:

#	Provider	Remarks
1	Redis	Overall the strongest performer winning 4 of 6 tests.
2	In-Proc	Good performer, but not suitable for web farms.
3	Session State Service	Very solid performer.
4	Couchbase	Another solid performer.
5	SQL Server	Best memory manager, but still a sledge hammer.
6	MongoDb	Poor performance under high load, possible memory issues.
7	RavenDb	The provider has a memory leak.

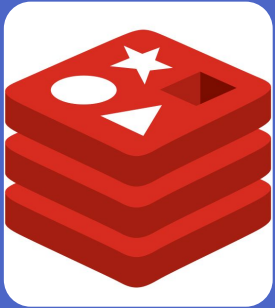
#	Provider	Server Response Time (ms)	Throughput (RPM)	Response Time (Avg.)(ms)	Response Time (Max)(ms)	Memory Management Hourly Growth	Processor Time (avg)
2	In-Proc	31	7630	401.56	25,002	0.58%	26%
3	Session State Service	33	7590	493.29	26,583	1.74%	29%
5	SQL Server 2012	34	7520	627.05	29,780	-0.26%	23%
1	Redis	33	7640	384.37	23,065	1.38%	20%
6	MongoDb	35	7060	1877.74	120,185	3.57%	25%
4	Couchbase	35	7590	508.99	27,615	1.48%	27%
7	RavenDb	1220	0.267	7584.38	41,512	91.5%	35%

Fonte: <https://www.slideshare.net/devopsguys/best-performing-aspnet-session-state-providers>

Come si evince dalla seconda figura, le migliori prestazioni, espresse in millisecond, si ottengono su Redis.

7. Conclusioni e Sviluppi futuri

Sviluppi futuri



Più semplice salvare lo stato della sessione
Maggiori prestazioni in termini di richieste
Database non relazionale



Interfaccia grafica
Semplice uso
Più lento

Aumentare il cluster di dati da memorizzare nella sessione, andando a studiare come cambiano le performance all'aumentare del numero di oggetti, che andamento ha il grafico (esponenziale, lineare o di altro tipo).

The background is a dark blue gradient. It is decorated with white, stylized circuit board traces. These traces are located in the corners and along the sides, featuring small circles at various points, resembling solder points or vias. The traces are more prominent on the left and right sides, with some extending towards the top and bottom edges.

GRAZIE PER
L'ATTENZIONE