



# **Analisi e confronto delle prestazioni degli OGC Web Services realizzati dai server geospaziali FOSS**

Laureando: Greta Pellegrini

Relatore: Prof. Massimo Rumor

Corso di laurea in Ingegneria Informatica

30 Settembre 2010

A.A. 2009/2010



*Ai miei genitori*



## Indice

1	Obiettivi del tirocinio.....	1
1.1	Problematica .....	1
1.2	Obiettivi del progetto.....	1
2	I server geospaziali e gli standard OGC .....	3
2.1	Gli standard OGC.....	5
2.1.1	WMS.....	5
2.1.2	WFS .....	8
2.1.3	WCS.....	8
2.1.4	SLD.....	8
2.2	I server geospaziali .....	10
2.2.1	GeoServer .....	10
2.2.2	MapServer.....	12
3	Progetto.....	15
3.1	Requisiti.....	15
3.2	Fasi del lavoro.....	15
4	Realizzazione del software di test .....	17
4.1	Ricerca degli strumenti .....	17
4.2	Realizzazione .....	19
5	Configurazione e analisi dei test.....	25
5.1	Configurazione delle macchine .....	25
5.1.1	Configurazione pc per l'esecuzione del software di testing.....	25
5.1.2	Configurazione pc per l'esecuzione dei server geospaziali .....	25
5.2	Configurazione dei server geospaziali.....	25
5.2.1	Configurazione di GeoServer .....	25
5.2.2	Configurazione di MapServer .....	26
5.3	Configurazione dei test .....	26
5.4	Analisi dei risultati .....	28
5.4.1	Prima macro sessione di test .....	28
5.4.2	Seconda macrosessione di test.....	63
6	Conclusioni.....	85
6.1	Risultati conseguiti .....	85
6.2	Problemi aperti.....	85
6.3	Sviluppi futuri .....	86

## Indice delle tabelle

Tabella 1 Parametri della richiesta GetCapabilities .....	5
Tabella 2 Parametri della sezione Service del Capabilities document .....	6
Tabella 3 Parametri richiesta GetMap .....	7
Tabella 4 Parametri della richiesta GetFeatureInfo.....	7
Tabella 5 Parametri della richiesta GetLegendGraphic .....	9

# 1 OBIETTIVI DEL TIROCINIO

---

## 1.1 PROBLEMATICHE

La problematica consiste nella progettazione e realizzazione di uno strumento per testare le performance e l'efficienza di differenti server geospaziali, utilizzati dall'azienda 3DGIS srl. Quest'ultima si occupa di progettazione e sviluppo di applicazioni GIS (Geographic Information System) e Web GIS con tecnologie FOSS (Free Open Source Software) e commerciali e architetture service oriented, secondo gli standard OGC.

In particolare, 3DGIS srl realizza web application per enti pubblici che operano nell'ambito dei sistemi informativi territoriali, cioè sistemi per la produzione e gestione di dati spaziali<sup>1</sup> dal mondo reale.

Tra le altre funzionalità, le suddette web application consentono di pubblicare dati cartografici sul web che possono quindi essere successivamente visualizzati, interrogati e manipolati da tutti, o solo dagli utenti che possiedono i permessi necessari.

Proprio il crescente numero di utenti che richiedono l'accesso ai dati, ha evidenziato sempre più l'esigenza di rendere gli applicativi realizzati da 3DGIS in grado di soddisfare le richieste, anche concorrenti, in tempi "accettabili" e di garantire la miglior risposta possibile.

Ecco allora la necessità di testare l'efficienza delle tecnologie utilizzate, in particolare dei server geospaziali coinvolti nella visualizzazione e manipolazione di dati spaziali attraverso la rete.

## 1.2 OBIETTIVI DEL PROGETTO

L'obiettivo del tirocinio è la realizzazione di un software per l'analisi delle performance di server geospaziali, che consenta di:

- determinare quale server geospaziale presenta le performance migliori;
- valutare se l'upgrade da una versione all'altra di uno stesso server geospaziale si rispecchia o meno in un miglioramento delle prestazioni;
- migliorare le prestazioni di un'applicazione, analizzandone il comportamento al variare del workload;
- analizzare la velocità, la scalabilità, e la stabilità di un prodotto prima di rilasciarlo;
- determinare quanti utenti concorrenti può gestire un'applicazione, prima che i tempi di risposta divengano inaccettabili;
- comprendere quali modifiche alla configurazione dei server o ai dati stessi possono migliorare le prestazioni.

---

<sup>1</sup> dato spaziale: rappresentazione di un oggetto o fenomeno del mondo reale.



# 2 I SERVER GEOSPAZIALI E GLI STANDARD OGC

---

Una web application è un'applicazione che consente ai suoi utenti (client) di accedere alle funzionalità che essa offre attraverso un web browser.

Nel caso specifico, le applicazioni Web GIS gestiscono banche dati cartografiche e consentono agli utenti di:

- consultare mappe statiche;
- effettuare l'interrogazione dinamica delle mappe;
- effettuare interrogazioni e analisi spaziali on-line di dati geografici;
- accedere a dati GIS attraverso Internet.

La struttura tipica di un'applicazione Web GIS è rappresentata da un'**architettura service oriented (SOA) e three tier**.

L'OASIS (Organization for the Advancement of Structured Information Standards) definisce la SOA come:

*“Un paradigma per l'organizzazione e l'utilizzazione di risorse distribuite che possono essere sotto il controllo di domini di proprietà differenti”.*

Si tratta in sostanza di un modello architetturale adatto a supportare l'uso dei web services. Questi ultimi sono applicazioni che possono essere pubblicate, reperite ed invocate sul web e che consentono, attraverso l'uso di standard, l'interoperabilità tra elaboratori diversi che interagiscono all'interno di una rete.

I web services dispongono di un'interfaccia che espone le operazioni attraverso cui applicazioni esterne possono interagire con il web service stesso. In particolare, per attivare tali operazioni, è necessario l'invio di una richiesta formattata secondo lo standard XML (eXtensible Markup Language) e trasportata, in genere, tramite il protocollo HTTP.

I web service utilizzano, oltre ad HTTP, altre tecnologie web come:

- SOAP (Simple Object Access Protocol), un protocollo per lo scambio di messaggi tra componenti software;
- UDDI (Universal Description Discovery and Integration), un registro che fornisce informazioni sui web service, più precisamente UDDI ha due tipi di clienti:
  - fornitori: coloro che vogliono pubblicare un servizio e la sua interfaccia;
  - clienti: coloro che vogliono ottenere un determinato servizio.

Un registro UDDI è quindi costituito da informazioni:

- white pages: contatti dell'azienda che offre uno o più servizi;
- yellow pages: classificazione dei servizi;
- green pages: informazioni tecniche sui web service;

e da operazioni:

- publish: per pubblicare un web service;
  - find: per reperire un web service;
  - bind: per connettersi e interagire con un web service.
- WSDL (Web Services Definition Language): un linguaggio XML che descrive cosa può fare un servizio e come può essere invocato.

Come tutte le architetture *three tier* l'architettura su cui si basano le applicazioni web GIS è costituita da tre strati, o livelli, sotto elencati a partire dal più basso:

1. **data tier:** si occupa della gestione dei dati (nel caso specifico dati spaziali e alfanumerici). Esistono diverse alternative, tra queste:
  - 1.1 gestione di tutti i dati tramite file (in genere file *shp* che archiviano le geometrie degli oggetti);
  - 1.2 gestione dei dati spaziali su file e gestione dei dati alfanumerici su DBMS;
  - 1.3 gestione di tutti i dati attraverso un DBMS con estensione spaziale che permette di visualizzare, manipolare e interrogare gli stessi con operazioni scritte in SQL. Tra i database spaziali disponibili sul mercato: Oracle Spatial, PostgreSQL con estensione PostGIS.
2. **business tier:** elabora le informazioni e i dati spaziali del livello data tier. Le componenti principali di questo livello sono i **server geospaziali** e i **web server**. I primi (le cui caratteristiche e funzionalità saranno discusse nel paragrafo 2.2) gestiscono e pubblicano dati geospaziali come ad esempio GeoServer, MapServer, Deegree e ArcGIS Server. I secondi, attraverso l'utilizzo di software dedicato, gestiscono le richieste HTTP dei client e possono restituire file di qualsiasi tipo. Tra questi *Apache HTTP Server* sviluppato dalla Apache Software Foundation, *Internet Information Services* di Microsoft e *Sun One* di Sun Microsystems.
3. **presentation (o client) tier:** gestisce l'interfaccia tra il sistema e l'utente, trasforma quindi i risultati ottenuti dalle elaborazioni effettuate a livello business tier in un formato "comprensibile" all'utente.

Ogni livello fornisce servizi al livello superiore secondo le linee generali del paradigma client server e può essere modificato indipendentemente dagli altri.

## 2.1 GLI STANDARD OGC

L'Open Geospatial Consortium (OGC) è un'organizzazione internazionale no-profit costituita da circa 400 enti tra aziende, agenzie di governo e università che collaborano nello sviluppo di standard GIS per l'informazione geospaziale. Tra questi standard, quelli che definiscono gli OGC Web Services (OWS) sono di particolare interesse per lo sviluppo di SOA, in quanto specificano la struttura dell'interfaccia dei web services che garantisce l'interoperabilità tra sistemi diversi.

Tre sono i principali OGC Web Services per l'accesso all'informazione georeferenziata: Web Map Service (WMS), Web Feature Service (WFS) e Web Coverage Service (WCS).

### 2.1.1 WMS

Come definito dall'Open Geospatial Consortium, lo standard WMS fornisce un'interfaccia HTTP per la richiesta di mappe distribuite su una o più sorgenti dati.

L'interfaccia WMS è caratterizzata dalle seguenti operazioni:

- **GetCapabilities:** restituisce una lista delle operazioni messe a disposizione dal servizio WMS più informazioni specifiche sulle mappe visualizzabili; sotto sono riportati i parametri di tale richiesta.

Parametro	Richiesto/Opzionale	Descrizione
VERSION=version	O	versione richiesta
SERVICE=WMS	R	tipo di servizio
REQUEST=GetCapabilities	R	nome della richiesta
UPDATESEQUENCE=string	O	numero di sequenza o stringa per il controllo della cache

**Tabella 1: Parametri della richiesta GetCapabilities**

Esempio di una richiesta GetCapabilities:

```
http://hostname:port/geoserver/ows?service=wms&version=1.1.1&request=GetCapabilities
```

Il "Capabilities document" restituito è un documento XML lungo e complesso, ma molto importante. Le sue componenti più significative sono le seguenti:

- **Service:** sezione che contiene informazioni base del servizio WMS, alcune di queste sono obbligatorie, altre no. Vediamole nella tabella che segue:

Informazione	Richiesta/Opzionale	Descrizione
Title	R	titolo del servizio
Name	R	OGC:WMS
OnLineResource	R	utilizzato, ad esempio, per specificare il sito web del fornitore del servizio
Abstract	O	descrizione del servizio
Keyword List	O	lista di parole chiave per facilitare la ricerca
Contact Information	O	informazioni per contattare la società che sviluppa il servizio
Fees	O	di solito impostato a 'none'
Access Constraints	O	vincoli di accesso, di solito impostati a 'none'

**Tabella 2 Parametri della sezione Service del Capabilities document**

- **Request:** sezione che descrive le operazioni riconosciute da WMS e i parametri disponibili per ognuna di queste;
- **Layer:** sezione che elenca i layer che possono essere richiesti nella forma: "namespace:layer". Per ogni layer sono poi specificati titolo, nome, stili e altri metadati.
- **GetMap:** restituisce un'immagine (una mappa creata a partire da dati spaziali) oppure un'eccezione. Di seguito sono riportati i parametri di tale richiesta:

Parametro	Richiesto /Opzionale	Descrizione
VERSION=version	R	versione richiesta
REQUEST=GetMap	R	nome richiesta
LAYERS=layer_list	R	uno o più layer separati da virgole (opzionale se è presente il parametro SLD)
STYLES=style_list	R	uno o più rendering style separati da virgole per i layer specificati (opzionale se è presente il parametro SLD)
SRS=namespace:identifier	R	Spatial Reference System
BBOX=minx,miny,maxx,maxy	R	Coordinate della bounding box (in basso a sinistra, in alto a destra)
WIDTH=output_width	R	larghezza dell'immagine in pixel
HEIGHT=output_height	R	Altezza dell'immagine in pixel
FORMAT=output_format	R	formato output della mappa
TRANSPARENT=TRUE FALSE	O	trasparenza dello sfondo della mappa (default=FALSE)

BGCOLOR=color_value	O	colore dello sfondo in esadecimale (default=0xFFFFFFFF)
EXCEPTIONS=exception_format	O	formato in cui devono essere restituite le eccezioni (default=SE_XML)
TIME=time	O	valore temporale per il layer specificato
ELEVATION	O	Valore dell'altezza del layer
SLD=styled_layer_descriptor_URL	O	URL dell'SLD (solo per il WMS che supporta SLD)
WFS=web_feature_service_URL	O	URL di un WFS che restituisce le features da simbolizzare con l'SLD specificato nel parametro sopra

**Tabella 3 Parametri richiesta GetMap**

La risposta a una richiesta GetMap valida è una mappa con le caratteristiche specificate dai parametri della richiesta. Una richiesta non valida invece restituisce un'eccezione.

- **GetFeatureInfo:** è un'operazione disponibile solo per quei layer che hanno l'attributo "queryable" impostato a 1 (true). Tale operazione fornisce informazioni su un punto preciso di una mappa (visualizzata in seguito a una richiesta GetMap) scelto dall'utente con un click. Poiché il protocollo WMS è "stateless", oltre a trasmettere le informazioni sul punto scelto dall'utente, sui layers da indagare e sul formato in cui la risposta alla richiesta dovrà essere restituita, l'operazione GetFeatureInfo ritrasmette anche la maggior parte dei parametri della richiesta GetMap precedente (bounding box, SRS, width, height,..). Di seguito sono riportati i parametri di tale richiesta:

Parametro	Richiesto/ Opzionale	Descrizione
VERSION=version	R	versione richiesta
<map_request_copy>	R	alcuni parametri della GetMap request che ha generato la mappa che si sta interrogando
REQUEST=GetFeatureInfo	R	nome della richiesta
QUERY_LAYERS=layer_list	R	lista dei layers da interrogare
INFO_FORMAT=output_format	O	formato output
FEATURE_COUNT=number	O	Numero di features per cui si desidera ricevere informazioni (default=1)
X=pixel_column	R	Coordinate in pixel della feature
Y=pixel_row	R	Coordinate in pixel della feature
EXCEPTIONS=exception_format	O	Formato output delle eccezioni
Vendor-specific parameters	O	Altri parametri opzionali

**Tabella 4 Parametri della richiesta GetFeatureInfo**

Altre operazioni opzionali sono disponibili solo per i servizi WMS che supportano lo standard SLD e sono descritte nel paragrafo 2.1.4.

### 2.1.2 WFS

Il WFS è uno standard che permette di ottenere informazioni sulle feature<sup>2</sup> di una mappa e di manipolarle attraverso operazioni di INSERT, UPDATE, DELETE, LOCK, QUERY e DISCOVERY.

Le operazioni presentate da WFS sono:

- **GetCapabilities:** descrive i tipi di feature disponibili e le operazioni che possono essere effettuate su di esse;
- **DescribeFeatureType:** restituisce informazioni sulle caratteristiche di una particolare feature tra quelle disponibili;
- **GetFeature:** restituisce una o più istanze di feature selezionandole attraverso determinati parametri;
- **GetGmlObject** (solo nella versione 1.1.0): permette il recupero di feature attraverso XLink (specifica che delinea metodi per creare collegamenti tra risorse in documenti XML, sia interne che esterne al documento);
- **Transaction:** gestisce una richiesta di transazione che può essere composta dalle seguenti operazioni: create, update, delete. Queste ultime permettono rispettivamente di creare, modificare e eliminare feature;
- **LockFeature:** blocca la richiesta di una o più istanze di un particolare tipo di feature per la durata di una transazione.

### 2.1.3 WCS

Tale servizio implementa le operazioni GetCapabilities, DescribeFeatureType e GetFeature dello standard WFS ma per i dati in formato raster.

### 2.1.4 SLD

Il servizio WMS non consente all'utente di definire stili personalizzati per la tematizzazione di una mappa. Ciò che permette di fare ciò è un'estensione di WMS, cioè la codifica SLD (Styled Layer Descriptor). Quest'ultima definisce la struttura del documento XML che l'utente deve predisporre per associare un particolare stile ad un layer.

Esistono due modi per definire uno stile:

1. Colorare tutte le features della mappa nello stesso modo, indipendentemente dai valori assunti dagli attributi delle features;
2. Definire come colorare le features in base ai valori assunti dagli attributi delle stesse (per esempio in una mappa stradale si può definire uno stile che analizza l'attributo 'tipo-strada' di una mappa e colora le strade con tipo-strada 'statale')

---

<sup>2</sup> Feature: unità fondamentale dell'informazione geografica, astrazione di un fenomeno del mondo reale. Una feature possiede: identificatore, tipo, attributi, geometria, relazioni con altre features,... Esempi: strada statale, edificio, lago.

in giallo, le strade con tipo-strada 'autostrada' in verde, le strade con tipo-strada 'superstrada' in blu). Per conoscere gli attributi delle feature si usa l'operazione DescribeFeatureType del servizio WFS (descritta del paragrafo 2.1.2).

Un documento SLD è quindi costituito da una serie di regole (RULE) che definiscono come simboleggiare una particolare feature.

Operazioni:

- **GetLegendGraphic:** restituisce una legenda che specifica come le differenti features sono rappresentate nella mappa.

Di seguito sono riportati i parametri disponibili per l'operazione GetLegendGraphic:

Parametro	Richiesto/Opzionale	Descrizione
VERSION	R	versione
REQUEST	R	deve essere GetLegendGraphic
LAYER	R	layer di cui è richiesta la legenda
STYLE	O	stile del layer specificato nel parametro layer di cui si vuole ottenere la legenda, se non è specificato viene restituita la legenda dello stile di default
RULE	O	Nome della regola definita all'interno dello stile di cui si vuole visualizzare la legenda. Se non è specificata alcuna regola, la legenda includerà tutte le regole definite nello stile specificato (o nello stile di default se non specificato)
SCALE	O	Se non è specificato il parametro RULE, SCALE esclude dalla legenda alcune regole considerate "out-of-scope"
SLD	O	Definisce il riferimento a un SLD esterno
SLD_BODY	O	Definisce il documento SLD direttamente all'interno della richiesta HTTP
FORMAT	O	Formato in cui verrà restituita la legenda
WIDTH	O	Larghezza della legenda in pixel
HEIGHT	O	Altezza della legenda in pixel
EXCEPTIONS	O	Formato in cui devono essere restituite eventuali eccezioni

**Tabella 5 Parametri della richiesta GetLegendGraphic**

- **DescribeLayer:** per definire uno stile, l'utente necessita di informazioni sui layer e sulle feature da simbolizzare. L'operazione DescribeLayer è stata progettata proprio per questo, restituisce infatti un documento XML (eXtensible Markup Language) contenente informazioni sui layer specificati nella richiesta.
- **GetStyles:** permette di avere informazioni sugli stili definiti dall'utente. La risposta alla richiesta è un SLD document oppure un'eccezione nel caso di richiesta invalida;
- **PutStyles:** operazione utilizzata per memorizzare stili e layer definiti dall'utente.

## 2.2 I SERVER GEOSPAZIALI

Sono stati presi in considerazione i server geospaziali FOSS che implementano gli OWS, in particolare WMS, WFS, WCS ed SLD. Tra questi sono stati selezionati GeoServer e MapServer in quanto attualmente i più utilizzati.

### 2.2.1 GEOSERVER

OWS supportati da GeoServer (v 2.0.2):

- WMS: 1.1.1
- WFS: 1.0.0 e 1.1.0
- WCS: 1.0.0 e 1.1.0

È un software open source (rilasciato sotto licenza GNU), Java-based, che permette ai suoi utenti di visualizzare, modificare e condividere dati geospaziali attraverso l'utilizzo di protocolli definiti dall' OGC.

GeoServer è nato nel 2001 da "The Open Planning Project" (TOPP), un progetto con sede a New York ideato per creare una suite di strumenti che rendessero l'attività del governo più trasparente.

Con GeoServer nacque anche il progetto **GeoTools**, una libreria Java sviluppata per consentire a GeoServer di lavorare con vari formati vettoriali e geDBMS.

GeoTools toolkit è tuttora sfruttato da GeoServer ed è caratterizzato da un'architettura modulare che permette di aggiungere funzionalità in modo agevole, è rilasciato sotto GNU Lesser General Public Licence (LGPL) e contiene metodi standard compliant per la manipolazione di dati spaziali.

Contestualmente allo sviluppo di GeoServer infatti, OGC stava lavorando allo standard **Web Feature Service** (WFS), un protocollo per rendere i dati spaziali direttamente disponibili sul web usando **GML** (Geographic Markup Language, la grammatica XML definita sempre dall'OGC per specificare le caratteristiche di oggetti geografici).

Altri progetti che nacquero proprio per rafforzare le funzionalità di GeoServer sono:

- **Refractions Research**, che realizzò **PostGIS**, un database spaziale che abilita GeoServer alla connessione ai database.
- **MetaCarta**, ideò invece **Open Layers**, una libreria Java Script open source. Tale libreria è tuttora integrata in GeoServer e rilasciata sotto la licenza BSD-style; è in continuo sviluppo e rende la generazione delle mappe e la visualizzazione delle stesse nei moderni web browser, facile e veloce.

La prima versione di GeoServer (1.0) è stata rilasciata nell'ottobre del 2001. SourceForge.net (al momento il più grande sito di sviluppo e controllo di software open source), in quel mese, registrò circa 500 download del pacchetto GeoServer.

Poco dopo il rilascio della prima versione, che implementava solo la specifica WFS definita da OGC, un programmatore di un'azienda commerciale spagnola sviluppò il supporto per la specifica **Web Map Service** (WMS), un protocollo per la creazione e visualizzazione di mappe create a partire da dati spaziali.

Successivamente altri progetti hanno contribuito ad incrementare le funzionalità di questo server geospaziale. Nel 2005 la compagnia italiana GeoSolutions sviluppò il supporto per lo standard **Web Coverage Services** (WCS) e per la *grafica raster* che rappresenta le immagini descrivendole come una griglia di pixel opportunamente colorati (contrapposta alla *grafica vettoriale* che invece descrive le immagini attraverso

primitive geometriche che definiscono punti, linee e poligoni ai quali possono essere attribuiti colori e sfumature).

Nell'agosto del 2007 è stata poi rilasciata la versione 1.5.3 e sono stati registrati approssimativamente 8500 download. Al crescere di questi ultimi e dei progetti sviluppati intorno a GeoServer, corrispondeva la crescita del numero di utenti e sviluppatori.

Il successo di GeoServer è dovuto proprio al fatto che, essendo un progetto open source, la sua gestione è sempre stata delegata a tutta la comunità (formata da soggetti provenienti da diverse organizzazioni) e mai a una singola organizzazione.

La versione corrente è la 2.0.2, rilasciata il 24 Maggio 2010 ed è già disponibile la versione 2.1 Beta (4 Settembre 2010).

Lo sviluppo di GeoServer continua per rendere i dati spaziali accessibili a tutti. In questo senso, sta lavorando con Google per rendere i propri dati accessibili e ricercabili attraverso Google Map. Presto, ricercare dati spaziali diventerà quindi facile come cercare una pagina web su Google.

### Funzionalità di GeoServer

Una volta installato GeoServer si può accedere tramite browser alla sua Web Administration Interface, un tool web based che permette di configurare GeoServer e di provvedere alla gestione dei dati spaziali.

Loggandosi (con nome utente *admin* e password *geoserver*, definiti e modificabili dal file *users.properties* contenuto nella cartella *security* della GeoServer data directory, o direttamente dalla sezione *security* della Web Administration Interface) infatti è possibile accedere alle seguenti sezioni:

- **Server:** contiene informazioni sullo stato del server, parametri di configurazione e informazioni utili per il debugging;
- **Services:** permette la configurazione dei servizi supportati da GeoServer (WCS, WMS e WFS);
- **Data:** è la sezione più importante della Web Administration Interface e permette di aggiungere, modificare e cancellare dati spaziali. Tale sezione è a sua volta composta da:
  - **Workspaces:** un workspace è un contenitore di layer simili. È costituito da un nome e un Namespace URI;
  - **Stores:** sezione che consente di definire la sorgente dei dati (di seguito data source). Esistono quattro tipi di store, a ognuno dei quali corrispondono diversi formati di data source supportati da Geoserver. In particolare per i dati in formato raster uno store può essere solo un file (di tipo ArcGrid, GeoTIFF, Gtopo30, ImageMosaic o WorldImage), mentre per i dati in formato vettoriale uno store può essere un file (ESRI shapefile o un properties file), un database (PostGIS) o un server (Web Feature Server).
  - **Layers:** questa sezione permette di visualizzare, modificare, cancellare layer esistenti e di crearne di nuovi;
  - **Layer Groups:** permette di raggruppare più layer al fine di poterli richiamare con un nome univoco, anziché doverli specificare tutti;

- **Styles:** in questa sezione si possono definire le caratteristiche del rendering dei dati spaziali (colori, opacità, trasparenze, e altri attributi di visualizzazione). In GeoServer gli styles sono definiti usando SLD.
- **Security:** consente di aggiungere nuovi utenti e di modificare le credenziali di quelli già esistenti;
- **Demos:** contiene link a pagine utili per l'utente riguardo GeoServer e le sue caratteristiche;
- **Layer Preview:** permette di visualizzare i layer in diversi formati output (KML, GIF, JPEG, SVG, PNG, TIFF, PDF, OpenLayers).

Attraverso il "filtering" GeoServer permette inoltre di individuare le features che soddisfano una particolare condizione. Il filtering può essere usato ad esempio per associare una particolare simbolizzazione ad una mappa o a "filtrare" i risultati di una richiesta. GeoServer accetta filtri specificati in tre differenti linguaggi: OGC Filter encoding specification v 1.0, OGC Filter encoding specification v 1.1, CQL (Catalog Query Language).

Oltre a supportare gli standard OGC WMS, WFS, WFS-Transactional e WCS, GeoServer implementa anche il servizio WFS Versioning. Questo protocollo non è previsto dagli standard OGC ma è progettato per essere compatibile, permette di gestire uno storico delle modifiche e di ripristinare l'ambiente a uno stato precedente.

## 2.2.2 MAPSERVER

### OWS supportati da MapServer (v. 5.6.5):

- WMS 1.0.0, 1.1.0 e 1.1.1;
- WFS 1.0.0, 1.1.0;
- WCS 1.0.0 e 1.1.0.

MapServer è un ambiente di sviluppo multiplatforma scritto in C per la rappresentazione di dati geografici (rilasciato sotto licenza MIT-style license).

È nato nel 1994 dal ForNet project dell'università del Minnesota ed è stato originariamente sviluppato dalla stessa in collaborazione con la NASA e il Dipartimento di Risorse Naturali del Minnesota. Inizialmente MapServer funzionava solo su sistemi Unix based. Le fasi principali dello sviluppo di MapServer sono le seguenti:

- 1995-1996: introduzione librerie *GD* e *shapelib* per supportare la creazione delle immagini, le prime organizzazioni cominciano ad utilizzare MapServer e nasce la prima user community;
- 1997: sviluppo supporto per la gestione dei dati in formato raster;
- 1999: MapServer diventa un progetto OpenSource;
- 2000: anche i sistemi Windows possono finalmente lavorare con MapServer;
- 2001: implementazione dello standard WMS e supporto per l'interazione con database PostGIS;
- 2004: implementazione dello standard WCS, sviluppo protocollo FastCGI (per migliorare le performance del server);
- 2007: introduzione di nuove librerie.

Oggi MapServer è un progetto OSGeo (fondazione per lo sviluppo di software geospaziale open source), mantenuto da un crescente numero di sviluppatori (circa venti al momento) distribuiti in tutto il mondo, supportato da un gruppo eterogeneo di organizzazioni e amministrato dal MapServer Project Steering Committee di OSGeo costituito da sviluppatori e contribuenti.

Le caratteristiche principali di MapServer sono le seguenti:

- supporto per visualizzare e interrogare un'ampia varietà di formati raster, vettoriali e DBMS;
- supporto per i linguaggi di scripting e gli ambienti di sviluppo più popolari (PHP, Python, Perl, Ruby, Java, Net)
- possibilità di operare su molteplici sistemi operativi (Windows, Linux, Mac OS X, ecc. ).

A differenza di GeoServer, MapServer non dispone di una GUI per l'utente per facilitare la gestione dei dati. Quest'ultima avviene invece attraverso la creazione di file con estensione "map".

Un'applicazione MapServer consiste infatti in:

- un **Map file**: file di configurazione strutturato con estensione .map (il cuore di MapServer) che definisce le caratteristiche dell'area della mappa e comunica al MapServer program la sorgente dei dati;
- **pagine HTML**: utilizzate da MapServer per visualizzare le risposte alle richieste dell'utente;
- **MapServer CGI (Common Gateway Interface)**: file binario o eseguibile che riceve richieste e restituisce immagini, dati, eccezioni. I programmi CGI, sono 'stateless', cioè ogni volta che ricevono una richiesta da un'applicazione, la considerano sempre come nuova, non ricordano nulla delle richieste ricevute in precedenza (anche se identiche a quella appena ricevuta) e devono quindi rielaborare tutte le informazioni di contesto ricevute (layer, stili, formato richiesto, ecc..).
- **HTTP Server**: gestisce le richieste HTTP dei client.

Il cuore di MapServer è quindi costituito dai Map file che definiscono relazioni tra oggetti, la sorgente dei dati e le opzioni di visualizzazione degli stessi (colori, stili, proiezioni,..). Tale file ha una struttura gerarchica, all'apice della quale c'è l'oggetto MAP.

Struttura tipica di un map file:

```
MAP
.      LAYER
.      .      CLASS
.      .      .      STYLE
.      .      .      END STYLE
.      .      END CLASS
.      END LAYER
.      LAYER
.      ...
.      END LAYER
END MAP
```

Come GeoServer, MapServer può gestire una grande varietà di dati sia in formato vettoriale che raster.

I formati di dati vettoriali supportati da Mapserver sono: Shapefiles, Microstation Design Files (DGN), GeoTIFF images, ESRI ArcInfo Coverages, US Census TIGER, PostGIS, ESRI ArcSDE, MySQL. I formati raster supportati sono invece: TIFF/GeoTIFF, GIF, PNG, ERDAS Imagine, JPEG;

L'interoperabilità è fondamentale per le applicazioni che distribuiscono e condividono dati su Internet. Per questo MapServer, che gestisce la pubblicazione di dati cartografici sulla rete, supporta numerosi standard OGC, in particolare: WMS, WFS, WFS Filter Encoding, WCS e l'estensione WMS per l'SLD.

### **Principali differenze tra GeoServer e MapServer**

Le principali differenze tra GeoServer e MapServer sono le seguenti:

1. MapServer utilizza la tecnologia CGI, mentre GeoServer utilizza JEE (Java Enterprise Edition);
2. MapServer è rilasciato sotto licenza MIT-style license mentre GeoServer sotto licenza GNU;
3. MapServer è un progetto più maturo di GeoServer, il primo è nato negli anni novanta, il secondo nel 2001;
4. GeoServer è scritto principalmente in Java, mentre MapServer in C;
5. GeoServer ha una Web Administration Interface che facilita la gestione dei dati, MapServer invece non ne è dotato;
6. GeoServer supporta WFS-Transactional mentre MapServer no.

# 3 PROGETTO

---

## 3.1 REQUISITI

Lo strumento realizzato dovrà risolvere la problematica presentata, in particolare dovrà ricevere in input una o più richieste HTTP e produrre in output informazioni sulle performance (tempi di risposta sia per richieste singole che concorrenti) dei server geospaziali chiamati a processare le richieste HTTP specificate.

Nella fase di ricerca dello strumento di benchmarking si dovrà tener conto dei seguenti vincoli:

- si desidera utilizzare software FOSS;
- il software finale dovrà funzionare su sistemi operativi Windows e Unix;
- il software finale realizzato dovrà consentire la configurazione veloce dei test.

## 3.2 FASI DEL LAVORO

- **Definizione dell'ambiente di testing:** definire il contesto del progetto, acquisire conoscenza degli strumenti disponibili e dei vincoli.
- **Realizzazione dello strumento di testing:** sul mercato esistono già benchmarking tools, si tratterà quindi di scegliere il più opportuno e di costruire uno strumento che lo sfrutti in grado di analizzare e confrontare i risultati che restituisce, al fine di soddisfare le esigenze oggetto della problematica.
- **Definizione e configurazione dell'ambiente di testing:**
  - definizione della struttura dei test;
  - pianificazione dei test: devono essere tali da simulare e rispecchiare la variabilità di richieste di utenti reali;
  - acquisizione e configurazione delle risorse necessarie per l'esecuzione dei test.
- **Implementazione e esecuzione dei test pianificati:** sviluppo, esecuzione e monitoraggio dei test pianificati;
- **Analisi dei risultati:** valutazione dei risultati conseguiti.



# 4 REALIZZAZIONE DEL SOFTWARE DI TEST

---

## 4.1 RICERCA DEGLI STRUMENTI

### Strumenti di benchmarking

Tra i benchmarking tools free presenti sul mercato sono stati presi in considerazione i seguenti:

- **JMeter**: è un'applicazione desktop open source scritta in Java, sviluppata dalla Apache Software Foundation che permette di effettuare test di carico delle proprie applicazioni. Inizialmente JMeter fu progettato per testare le applicazioni Web ma, successivamente, è stato esteso e attualmente permette di testare qualsiasi tipologia di software.

Con questo strumento è possibile simulare la concorrenza attraverso l'uso di threads permettendo quindi di misurare la performance al variare del carico. I risultati prodotti da JMeter forniscono valori statistici come la media, la mediana, la varianza e il throughput.

È sufficiente scaricare JMeter e avviarlo per cominciare a pianificare i test, tuttavia quest'ultima attività può risultare piuttosto laboriosa anche se facilitata da un'interfaccia utente.

- **ab - Apache HTTP server benchmarking tool**: è un comando di estensione per il Server HTTP Apache che permette di simulare richieste HTTP e di valutare, tra le altre metriche, il tempo che il server impiega a servire la richiesta, la quantità di dati trasferiti, il numero di richieste servite al secondo.

Anche *ab* consente di simulare un workload elevato e richieste multiple concorrenti ma non dispone di un'interfaccia utente.

Per lanciare *ab* è sufficiente digitare da riga di comando:

```
ab [opzioni] http[s]://[hostname[:port]]/path
```



richiesta HTTP da inviare al server

Per realizzare lo strumento di benchmarking evaluation, è stato scelto *ab* in quanto permette di automatizzare e velocizzare la fase di definizione dei test da effettuare. JMeter, pur essendo una valida alternativa come strumento per il benchmarking e pur disponendo di un'interfaccia grafica, rende la fase di configurazione dei test gravosa e laboriosa per l'utente. Ciò è in contrasto con i requisiti del progetto che impongono che la fase di configurazione dei test debba essere semplice, intuitiva e veloce.

Di seguito sono riportate le opzioni di *ab* disponibili nella versione corrente (2.3):

- **-A** *auth-username:password* credenziali per autenticarsi al server
- **-b** *window-size* dimensione in byte del buffer TCP di invio/ricezione
- **-c** *concurrency* numero di richieste da eseguire per volta (default=1)
- **-C** *cookie-name=value* aggiunge cookie alla richiesta, cioè coppie nella forma "nome:valore" che verranno inviate al server
- **-d** non visualizza la tabella che mostra la percentuale di richieste servite e il relativo tempo in ms

- **-e** *csv-file* scrive in un file csv la percentuale di richieste servite (da 1% a 100%) e il tempo in ms che il server impiega per servire questa percentuale di richieste
- **-f** *protocol* specifica i protocolli crittografici per la comunicazione (SSL/TSL)
- **-g** *gnuplot-file* scrive tutto l'output in un file TSV (Tab Separate Values)
- **-h** visualizza informazioni sulle opzioni disponibili per il comando *ab*
- **-H** *custom-header* aggiunge intestazioni alla richiesta
- **-i** effettua richieste HEAD anziché GET
- **-k** abilita HTTP keep alive
- **-n** *requests* numero di richieste da effettuare in una sessione di benchmark
- **-p** *POST-file* file che contiene dati da inviare attraverso una richiesta HTTP POST (da impostare insieme all'opzione **-T**)
- **-P** *proxy-auth-username:password* credenziali per l'autenticazione a un eventuale proxy di rete
- **-q** elimina i messaggi verbosi restituiti da *ab* quando il numero di richieste è superiore a 150
- **-r** non termina l'esecuzione dei test se riceve un errore dal socket
- **-s** usa il protocollo HTTPS anziché HTTP (opzione ancora sperimentale)
- **-S** non visualizza la media e la deviazione standard nei risultati restituiti
- **-t** *timelimit* impone un limite alla durata del test (in secondi)
- **-T** *content-type* header da usare per metodi PUT e POST
- **-u** *PUT-file* nome del file contenete dati da inviare attraverso una richiesta HTTP PUT
- **-v** *verbosity* abilita messaggi verbosi come warning e info
- **-V** per visualizzare la versione di *ab* che si sta utilizzando
- **-w** visualizza i risultati in una tabella HTML
- **-x** *<table>-attributes* attributi da usare nella tabella di risultati che verrà visualizzata
- **-X** *proxy[:port]* specifica la porta del server proxy da usare per la richiesta
- **-y** *<tr>-attributes* intestazione da usare nelle righe nella tabella di risultati che verrà visualizzata
- **-z** *<td>-attributes* stringa da usare come attributo per i dati della tabella che verrà restituita
- **-Z** *ciphersuite* specifica la suite per la cifratura (SSL/TSL)

## 4.2 REALIZZAZIONE

Visto il poco tempo a disposizione si è scelto un linguaggio versatile e con tipizzazione debole, ma non per questo poco performante, quale è PHP.

Il software, realizzato appunto in PHP, è stato strutturato in modo tale da:

- generare richieste HTTP con le caratteristiche definite dall'utente;
- passare tali richieste ad *ab* a cui è a sua volta delegato il compito di calcolare i tempi impiegati dai server geospaziali a processare tali richieste e altre metriche;
- permettere all'utente di confrontare il comportamento dei server geospaziali testati confrontando le metriche misurate da *ab*.

### Funzionamento del software realizzato:

Il software realizzato prevede la previa predisposizione di un file di configurazione xml che definisce i test da effettuare. Il file di configurazione ha una struttura composta dai seguenti tag:

- **configuration**: elemento radice;
- **ID** : identificatore della sessione di test ;
- **description** : breve descrizione della sessione di test;
- **updateSequence**: può essere true o false. Si imposta a true per innescare un meccanismo che permette di fare in modo che due richieste consecutive identiche non vengano considerate tali dal server che deve soddisfarle (il server non potrà quindi accedere alla cache per soddisfare la richiesta, ma dovrà riprocessarla);
- **servers**: contiene il nome dei server che si vogliono testare e relativo URL (nel caso specifico i test analizzati sono GeoServer e MapServer ma il software è progettato per poterne gestire anche più di due);
- **testOptions**: contiene un tag per ogni opzione di *ab* disponibile nella versione corrente come per esempio il numero di richieste da effettuare e il livello di concorrenza;
- **tests**: contiene una o più sezioni **test**. Ognuna di queste è composta dai seguenti tag: id (identificatore del test), nome (breve descrizione del test, facoltativa), metodo (GET o POST), parentId (va inserito solo nel caso in cui si voglia effettuare un test simile ad un altro definito nello stesso file di configurazione) e infine i parametri veri e propri della richiesta HTTP da inviare al server geospaziale.

Di seguito è riportato il contenuto di un file di configurazione che esemplifica quanto sopra.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <ID>PostGIS_DefaultStyle_richieste_multiple</ID>
  <description>default style; all, 10000, 1000 feature
</description>
  <servers>
    <server>
      <name>geoserver</name>
      <url>http://192.168.42.3:8100/geoserver/wms?</url>
    </server>
  </servers>
</configuration>
```

```

    <server>
      <name>mapserver</name>
      <url>http://192.168.42.3/cgi-bin/mapserv?map=
        maps/edifici.map&#38;
      </url>
    </server>
  </servers>
  <testOptions>
    <usernamePassword></usernamePassword>
    <windowSize></windowSize>
    <concurrencyLevel>5</concurrencyLevel>
    <cookie></cookie>
    <protocol></protocol>
    <customHeader></customHeader>
    <numberOfRequests>50</numberOfRequests>
    <postFile></postFile>
    <cookie></cookie>
    <proxy></proxy>
    <timeLimit></timeLimit>
    <contentType></contentType>
    <putFile></putFile>
    <verbosity></verbosity>
    <tableAttributes></tableAttributes>
    <proxyServer></proxyServer>
    <trAttributes></trAttributes>
    <tdAttributes></tdAttributes>
    <ciphersuite></ciphersuite>
    <optionWithoutValue></optionWithoutValue>
  </testOptions>
  <tests>
    <test>
      <id>1</id>
      <name></name>
      <method>get</method>
      <parameters>
        <parameter>
          <name>service</name>
          <value>WMS</value>
        </parameter>
        <parameter>
          <name>version</name>
          <value>1.1.0</value>
        </parameter>
        <parameter>
          <name>request</name>
          <value>GetMap</value>
        </parameter>
        <parameter>
          <name>layers</name>
          <value>test:edifici</value>
        </parameter>
        <parameter>
          <name>styles</name>
          <value></value>
        </parameter>
        <parameter>
          <name>bbox</name>
          <value>2351630.45,4846423.31,
            2369777.78,4860692.46
          </value>
        </parameter>
      </parameters>
    </test>
  </tests>

```

```

        <parameter>
            <name>width</name>
            <value>512</value>
        </parameter>
        <parameter>
            <name>height</name>
            <value>402</value>
        </parameter>
        <parameter>
            <name>srs</name>
            <value>EPSG:3004</value>
        </parameter>
        <parameter>
            <name>format</name>
            <value>image/png</value>
        </parameter>
    </parameters>
</test>

<test>
    <id>2</id>
    <name>10000 feature, scala 1:38900</name>
    <parentId>1</parentId>
    <parameters>
        <parameter>
            <name>bbox</name>
            <value>2357063.45,4853164.66,2362474.25,
            4857269.82
            </value>
        </parameter>
    </parameters>
</test>
<test>
    <id>3</id>
    <name>1000 features, scala 1:8500</name>
    <parentId>1</parentId>
    <parameters>
        <parameter>
            <name>width</name>
            <value>600</value>
        </parameter>
        <parameter>
            <name>height</name>
            <value>500</value>
        </parameter>
    </parameters>
</test>
</tests>
</configuration>

```

Nel file di configurazione sopra riportato come esempio è definito un primo test con id 1 che specifica i parametri di una richiesta WMS. Nei test con id 2 e 3 è specificato anche il tag *parentId* e solo pochi parametri che chiamiamo nuovi parametri; ciò significa che si vuole effettuare un test che si differenzia da quello con *id* indicato dal *parentId* solo nei nuovi parametri specificati. Nell'esempio riportato quindi:

- il secondo test erediterà tutti i parametri del primo test eccetto il parametro *bounding box*;

- il terzo test erediterà tutti i parametri del primo eccetto i parametri *width* e *height*.

Il software riceve in input il nome del file di configurazione specificato dall'utente (è possibile specificare anche più file di configurazione), lo analizza e converte i test definiti nel file, in veri e propri oggetti Test che rappresentano le richieste da inviare ai server geospaziali.

La classe Test che costruisce queste richieste, realizza quindi il meccanismo di ereditarietà sopra spiegato attraverso il metodo *inheritFrom(id)*.

In particolare, ogni oggetto Test è rappresentato dalle seguenti variabili:

- *xmlNode*: permette di riferirsi a uno dei tests definiti all'interno del file di configurazione;
- *id*: deve essere univoco tra i test definiti nello stesso file ed è obbligatorio;
- *nome*: breve descrizione del test, opzionale;
- *metodo*: può essere GET o POST;
- *parentId*: id di un altro test. Variabile diversa da null solo per gli oggetti Test che ereditano parametri da un altro test;
- *parametri*: sono coppie 'nome valore' (per esempio quelli definiti per ogni richiesta nelle tabelle 1,2,3,4 e 5) che andranno appese all'URL specificato nella sezione *servers* per definire la richiesta HTTP da effettuare.

Il metodo più rilevante della classe Test è *getRequestParameters()*. Questo metodo restituisce una stringa del tipo

```
nomeParametro=valoreParametro&nomeParametro=valoreParametro&  
...
```

che costituisce parte della richiesta HTTP da inviare al server.

La classe *AbObject* appende a tale stringa l'url dei server da testare e costruisce il comando *ab* che sarà del tipo '*ab [opzioni] HTTPrequest*'. Per fare ciò analizza le sezioni *updateSequence*, *servers* e *testOptions* del file di configurazione.

In particolare se nel file di configurazione, *updateSequence* è impostato a *true*, la classe *AbObject* provvede ad appendere alla richiesta HTTP un parametro del tipo "UPDATESEQUENCE=numeroCasuale". Ciò consente di fare in modo che il server a cui è destinata la richiesta non possa recuperare velocemente dalla cache i dati da restituire; aggiungendo questo parametro casuale infatti si fa in modo che una richiesta sia sempre differente da quella precedente, impedendo a eventuali accessi alla cache di influenzare le performance.

Il software richiama poi il tool *ab* e i risultati restituiti da *ab* vengono catturati dall'oggetto *FinalResult*.

Quest'ultimo infatti memorizza in variabili le metriche misurate da *ab* (time per request, transfer rate,..) e delega all'oggetto *CsvResult* la gestione del file csv restituito da *ab* (un file contenente la percentuale di richieste servite e il tempo impiegato dal server per rispondere).

Attraverso la classe *GnuData* viene poi generato un file contenente, per ogni richiesta effettuata, l'id del test cui la richiesta si riferisce e i tempi impiegati a soddisfare la richiesta dai server considerati. Questo file sarà poi analizzato da *gnuplot* (un software per la creazione di grafici) che produrrà in output un istogramma rappresentativo dei risultati conseguiti.

Il software crea infine un file HTML (chiamato output.html) che per ogni server geospaziale oggetto dei test e per ogni test effettuato riporta:

- la richiesta HTTP effettuata;
- la risposta restituita (una mappa se si tratta di una richiesta GetMap, una legenda se si tratta di una richiesta GetLegendGraphic,..);
- i parametri della richiesta;
- il livello di concorrenza;
- il tempo impiegato per eseguire i test;
- il numero di richieste completate con successo;
- il numero di richieste fallite;
- il numero di errori riscontrati;
- il numero di byte trasferiti;
- la media del numero di richieste servite in un secondo;
- il tempo medio impiegato per soddisfare una richiesta;
- il transfer rate (KB/s);
- la percentuale di richieste servite (da 1 a 100) con relativo tempo in ms (solo nel caso in cui nel file di configurazione sia stato specificato un numero di richieste maggiore di 1);

Oltre al file HTML sopra specificato, per ogni sessione di test definita in un file di configurazione viene creata una cartella, con il nome specificato nella sezione ID del configuration file, contenete:

- il file csv restituito da *ab* per ogni test definito nel file di configurazione;
- una cartella con le risposte alla richiesta HTTP inviata al server (principalmente immagini);
- un file di testo readme.txt contenente una breve descrizione dei test effettuati, corrispondente al testo inserito nella sezione description del configuration file;
- un'immagine .png che mostra in forma grafica i risultati ottenuti (tali grafici saranno riportati nel paragrafo 5.4-Analisi dei risultati).

In questo modo è possibile tenere traccia dei test effettuati e dei risultati conseguiti per ogni test.



# 5 CONFIGURAZIONE E ANALISI DEI TEST

---

## 5.1 CONFIGURAZIONE DELLE MACCHINE

È stato utilizzato un pc per l'esecuzione del software di testing ed un pc su cui sono stati eseguiti i server geospaziali. Le macchine erano connesse ad una rete Ethernet 10/100 Mbps.

### 5.1.1 CONFIGURAZIONE PC PER L'ESECUZIONE DEL SOFTWARE DI TESTING

Processore: Intel Pentium M 1,7 GHz;

RAM: 1 GB

Sistema operativo: Ubuntu Linux 9.10 32-bit (kernel linux 2.6.31)

### 5.1.2 CONFIGURAZIONE PC PER L'ESECUZIONE DEI SERVER GEOSPAZIALI

Processore: Intel Core 2 Duo E8400 a 3 GHz;

RAM: 4 GB

Sistema operativo: Ubuntu Linux 9.10 64-bit (kernel linux 2.6.31)

## 5.2 CONFIGURAZIONE DEI SERVER GEOSPAZIALI

### 5.2.1 CONFIGURAZIONE DI GEOSERVER

Il servlet container scelto per l'esecuzione di GeoServer è Apache Tomcat (v. 6.0.26 rilasciato sotto licenza Apache Software License).

La versione di Java Development Kit (JDK) utilizzata è la 1.6.

Per migliorare le performance di GeoServer si è provveduto a:

- impostare alcuni valori della Java Virtual Machine (JVM);
- installare alcune librerie native come *Java Advanced Imaging API (JAI)*, una libreria per il supporto alla manipolazione delle immagini;
- Eliminare i dati di default esistenti che potrebbero appesantire la web application;
- modificare alcuni parametri attraverso l'interfaccia di amministrazione per ottimizzare la generazione delle immagini e la connessione con il geodbms.

Il processo di configurazione dei servizi e dei dati per i test è stato semplice perché è avvenuto attraverso l'interfaccia di amministrazione.

Per quanto riguarda i dati, sono stati creati due layer, uno con data source shapefile, l'altro con data source database PostGIS (versione di PostGIS utilizzata nel progetto 1.5.1 con estensione PostgreSQL v. 8.4).

Per il primo è stato sufficiente aggiungere uno store a quelli esistenti, specificando la cartella contenente i file con estensione .shp, .dbf, .shx e .prj. Questi file costituiscono la sorgente dei dati. Più precisamente, il file con estensione .shp archivia le geometrie degli oggetti, il file con estensione .dbf contiene informazioni sugli attributi degli oggetti, il file con estensione .shx contiene l'indice delle geometrie degli oggetti e infine il file con estensione .prj fornisce informazioni sul sistema di coordinate da utilizzare.

Per il secondo (layer con data source PostGIS) è stato aggiunto uno store specificando il nome del database a cui connettersi, l'username e la password per accedere al database e il nome dell'host che ospita il database .

## 5.2.2 CONFIGURAZIONE DI MAPSERVER

Essendo disponibile il pacchetto Ubuntu di MapServer, l'installazione è stata semplice in quanto ha evitato una lunga e laboriosa configurazione del prodotto stesso.

Per l'esecuzione di MapServer è stato utilizzato il web server Apache HTTP Server (v. 2.2.12) .

La configurazione dei dati e dei servizi è stata più laboriosa rispetto a quella di GeoServer a causa della mancanza di un'interfaccia di amministrazione. È stato infatti necessario configurare manualmente un map file contenente la definizione degli stessi layer visti per GeoServer.

Per migliorare le prestazioni di MapServer si è provveduto inoltre a installare ed eseguire MapServer con fastCGI, un'estensione dell'interfaccia CGI per ottimizzare le performance del server nel gestire più richieste contemporaneamente.

Si precisa inoltre che, per migliorare ulteriormente le performance di entrambi i server geospaziali, è stata aumentata la variabile sharedBuffer di PostgreSQL che specifica la quantità massima di memoria che PostgreSQL può usare per memorizzare richieste di accesso al database in attesa di essere processate.

## 5.3 CONFIGURAZIONE DEI TEST

Di seguito saranno presentate sedici sessioni di test che richiedono la visualizzazione di una mappa che rappresenta l'edificato di un comune, nel dettaglio:

*Prima macrosessione di test: data source PostgreSql/PostGIS*

1. Richiesta GetMap con stile di default;
2. Richiesta GetMap con la specifica di un SLD con una regola (la richiesta effettuata comprende la specifica di un SLD che contiene una regola che impone di colorare tutte le feature, edifici nella mappa considerata, di verde);
3. Richiesta GetMap con la specifica di un SLD con dieci regole (la richiesta comprende la specifica di un file SLD che contiene dieci regole; in particolare l'SLD specifica colori diversi a seconda del valore dell'attributo "area" delle feature rappresentate);
4. Richiesta GetLegendGraphic.

Ripetizione di queste quattro richieste effettuando 30 richieste al server (con livello di concorrenza 10)

- *Seconda macrosessione di test: data source shapefile*  
Identica alla sessione precedente ma con data source shapefile

Le due macrosezioni di test sopra presentate sono state ripetute più volte; confrontando i risultati ottenuti non sono state rilevate discrepanze significative nel comportamento dei server.

**Nota:**

Nell'SLD specificato al punto 3 sono state definite dieci regole, in particolare:

1. Colorare di blu le feature con area compresa tra 0 e 49 m<sup>2</sup>;
2. Colorare di rosso le feature con area compresa tra 50 e 999 m<sup>2</sup>;
3. Colorare di verde le feature con area compresa tra 1000 e 1499 m<sup>2</sup>;
4. Colorare di azzurro le feature con area compresa tra 1500 e 1999 m<sup>2</sup>;
5. Colorare di magenta le feature con area compresa tra 2000 e 2999 m<sup>2</sup>;
6. Colorare di fucsia le feature con area compresa tra 3000 e 4999 m<sup>2</sup>;
7. Colorare di salmone le feature con area compresa tra 5000 e 6999 m<sup>2</sup>;
8. Colorare di viola le feature con area compresa tra 7000 e 7999 m<sup>2</sup>;
9. Colorare di grigio le feature con area compresa tra 8000 e 9999 m<sup>2</sup>;
10. Colorare di marrone le feature con area compresa tra 10000 e 20000 m<sup>2</sup>;

## 5.4 ANALISI DEI RISULTATI

### 5.4.1 PRIMA MACRO SESSIONE DI TEST

Nelle tabelle seguenti sono riportati i risultati restituiti da *ab* per ogni richiesta effettuata. In ogni sessione di test, il test con ID 1 visualizza 10000 features della mappa, il test con ID 2 visualizza 1000 features della mappa, il test con ID 3 visualizza 100 features della mappa, infine il test con ID 4 visualizza 10 features della mappa.

L'unico parametro che cambia nella richiesta HTTP tra test con ID diverso è quindi il Bounding Box, gli altri sono gli stessi del test con ID 1.

In particolare, il bounding box è:

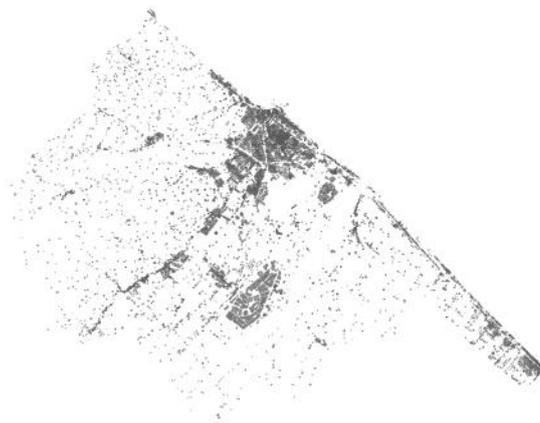
- 2357063.45,4853164.66,2362474.25,4857269.82 per i test con ID 1
- 2359316.97,4854816.46,2360499.28,4855713.47 per i test con ID 2
- 2359521.94,4855341.36,2359869.67,4855605.19 per i test con ID 3
- 2359418.07,4855938.09,2359560.49,4856046.16 per i test con ID 4

e specifica di visualizzare solo le feature della mappa che sono contenute (o parzialmente contenute) nella regione della mappa delimitata dalle coordinate riportate.

Le immagini restituite sono riportate solo nella prima sessione di test in quanto Geoserver e MapServer sono stati configurati in modo tale da restituire immagini pressoché identiche provvedendo per entrambi all'attivazione dell'antialiasing (di default per GeoServer, specificato nel map file per MapServer) e all'impostazione del formato dell'immagine da restituire (png a 8 bit).

Nelle tabelle che seguiranno, il parametro "Number of Request" specifica il numero di richieste inviate al server una di seguito all'altra, pari a 3 per le prime quattro richieste.

**Mappa originale interrogata** (contenente circa 19000 feature):



## Prima sessione di test - Caratteristiche principali:

Tipo di richiesta:	GetMap
Data source:	PostGIS
Formato output:	png 8
Style:	default

### Richiesta HTTP effettuata:

- Per GeoServer:  
`http://192.168.42.3:8080/geoserver/wms?service=WMS&version=1.1.0&request=GetMap&layers=test:edifici&styles=&bbox=2357063.45,4853164.66,2362474.25,4857269.82width=512&height=512&srs=EPSG:3004&format=image/png&UPDATESEQUENCE=5085`
- Per MapServer:  
`http://192.168.42.3/cgi-bin/mapserv?map=maps/edifici.map&service=WMS&version=1.1.0&request=GetMap&layers=test:edifici&styles=&bbox=2357063.45,4853164.66,2362474.25,4857269.82width=512&height=512&srs=EPSG:3004&format=image/png&UPDATESEQUENCE=5085`

Si noti che, ovviamente, nelle due richieste HTTP presentate sopra cambia solo l'intestazione della richiesta HTTP che rappresenta l'URL del server geospaziale. Nei test che seguiranno, per brevità sarà riportata solo la richiesta HTTP inviata a GeoServer.

### Test 1 - Test - Mappa restituita:



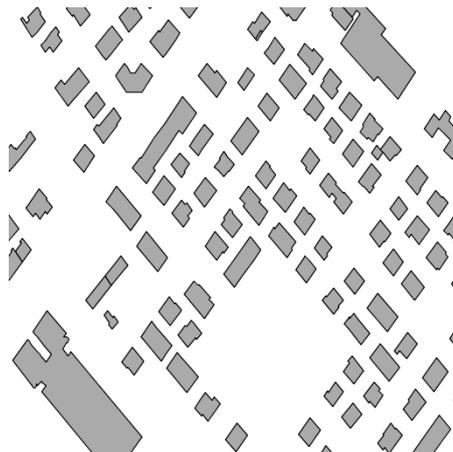
Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	1	1
Time taken for tests:	1.308 seconds	1.501 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	274587 bytes	159753 bytes
Requests per seconds:	2.29 [#/sec] (mean)	2.00 [#/sec] (mean)
Time per request:	435.969 [ms] (mean) 435.969 [ms] (mean, across all concurrent requests)	500.486 [ms] (mean) 500.486 [ms] (mean, across all concurrent requests)
Transfer rate:	205.02 [Kbytes/sec] received	103.90 [Kbytes/sec] received

## Test 2 - Test - Mappa restituita:



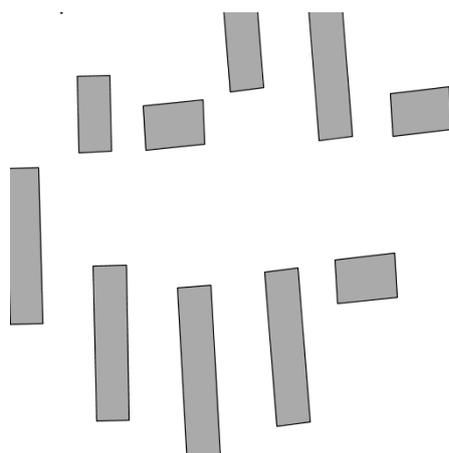
Server name	GeoServer	MapServer
<b>Test id:</b>	2	2
<b>Concurrency level:</b>	1	1
<b>Time taken for tests:</b>	0.538 seconds	0.479 seconds
<b>Complete requests:</b>	3	3
<b>Failed Requests:</b>	0	0
<b>Write errors:</b>	0	0
<b>Total transferred:</b>	402927 bytes	272103 bytes
<b>Requests per seconds:</b>	5.58 [#/sec] (mean)	6.27 [#/sec] (mean)
<b>Time per request:</b>	179.345 [ms] (mean) 179.345 [ms] (mean, across all concurrent requests)	159.558 [ms] (mean) 159.558 [ms] (mean, across all concurrent requests)
<b>Transfer rate:</b>	731.33 [Kbytes/sec] received	555.13 [Kbytes/sec] received

### Test 3 - Test - Mappa restituita:



Server name	GeoServer	MapServer
Test id:	3	3
Concurrency level:	1	1
Time taken for tests:	0.414 seconds	0.313 seconds
Complete	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	228924 bytes	198102 bytes
Requests per seconds:	7.24 [#/sec] (mean)	9.58 [#/sec] (mean)
Time per request:	138.033 [ms] (mean) 138.033 [ms] (mean, across all concurrent requests)	104.366 [ms] (mean) 104.366 [ms] (mean, across all concurrent requests)
Transfer rate:	539.87 [Kbytes/sec] received	617.88 [Kbytes/sec] received

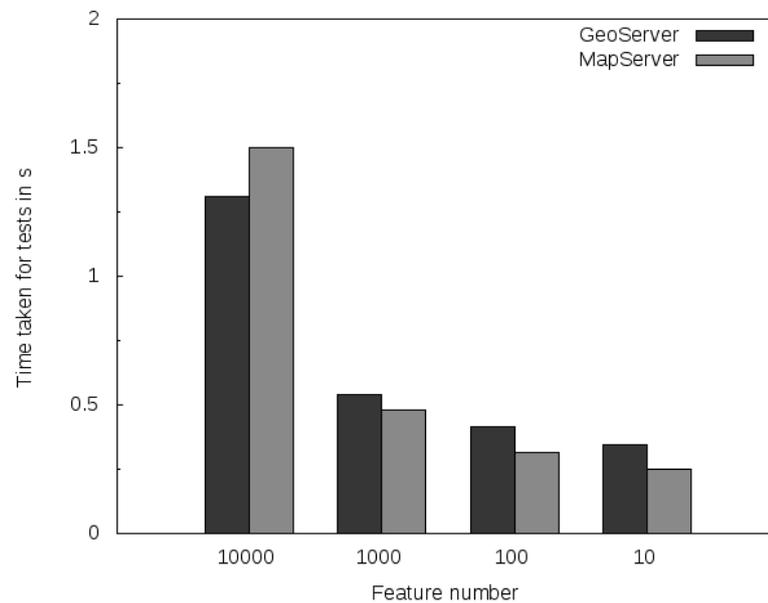
**Test 4 - Test - Mappa restituita:**



Server name	GeoServer	MapServer
Test id:	4	4
Concurrency level:	1	1
Time taken for tests:	0.345 seconds	0.251 seconds
Complete	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	60801 bytes	63507 bytes
Requests per seconds:	8.70 [#/sec] (mean)	11.93 [#/sec] (mean)
Time per request:	114.932 [ms] (mean) 114.932 [ms] (mean, across all concurrent requests)	83.812 [ms] (mean) 83.812 [ms] (mean, across all concurrent requests)
Transfer rate:	172.21 [Kbytes/sec] received	246.66 [Kbytes/sec] received

In tutti i grafici che seguiranno, la metrica scelta per il confronto delle prestazioni dei server è "Time taken for tests" cioè il tempo necessario per effettuare ogni test.

Per la prima sessione di test presentata, il grafico risultante è il seguente:



Dal grafico si può dedurre che richiedendo una mappa con data source PostGIS, GeoServer presenta un response time migliore di MapServer solo quando il numero di feature richieste è elevato.

## **Seconda sessione di test - Caratteristiche principali:**

Tipo di richiesta:	GetMap
Data source:	PostGIS
Formato output:	png 8
Style:	file SLD che definisce il colore della mappa con una regola

### **Richiesta HTTP effettuata:**

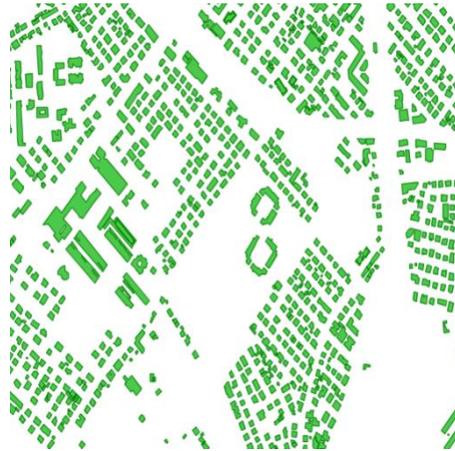
[http://192.168.42.3:8080/geoserver/wms?service=WMS&version=1.1.0&request=GetMap&layers=test:edifici&styles=&bbox=2357063.4483333323,4853164.6627976205,2362474.2530952375,4857269.81684524&width=512&height=512&sld=http://192.168.42.3/sld/edifici\\_one\\_rule.sld&srs=EPSG:3004&format=image/png&UPDATESEQUENCE=9579](http://192.168.42.3:8080/geoserver/wms?service=WMS&version=1.1.0&request=GetMap&layers=test:edifici&styles=&bbox=2357063.4483333323,4853164.6627976205,2362474.2530952375,4857269.81684524&width=512&height=512&sld=http://192.168.42.3/sld/edifici_one_rule.sld&srs=EPSG:3004&format=image/png&UPDATESEQUENCE=9579)

## Test 1 - Test - Mappa restituita:



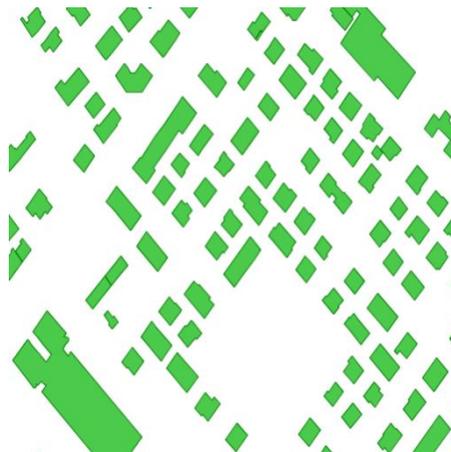
Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	1	1
Time taken for tests:	1.451 seconds	1.668 seconds
Complete	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	602055 bytes	130167 bytes
Requests per seconds:	2.07 [#/sec] (mean)	1.80 [#/sec] (mean)
Time per request:	483.687 [ms] (mean) 483.687 [ms] (mean, across all concurrent requests)	555.840 [ms] (mean) 555.840 [ms] (mean, across all concurrent requests)
Transfer rate:	405.18 [Kbytes/sec] received	76.23 [Kbytes/sec] received

## Test 2 - Test - Mappa restituita:



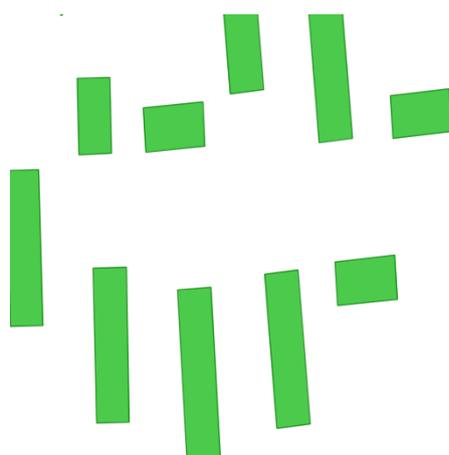
Server name	GeoServer	MapServer
Test id:	2	2
Concurrency level:	1	1
Time taken for tests:	0.554 seconds	0.717 seconds
Complete	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	664431 bytes	143973 bytes
Requests per seconds:	5.41 [#/sec] (mean)	4.18 [#/sec] (mean)
Time per request:	184.694 [ms] (mean) 184.694 [ms] (mean, across all concurrent requests)	239.070 [ms] (mean) 239.070 [ms] (mean, across all concurrent requests)
Transfer rate:	1171.05 [Kbytes/sec] received	196.04 [Kbytes/sec] received

### Test 3 - Test - Mappa restituita:

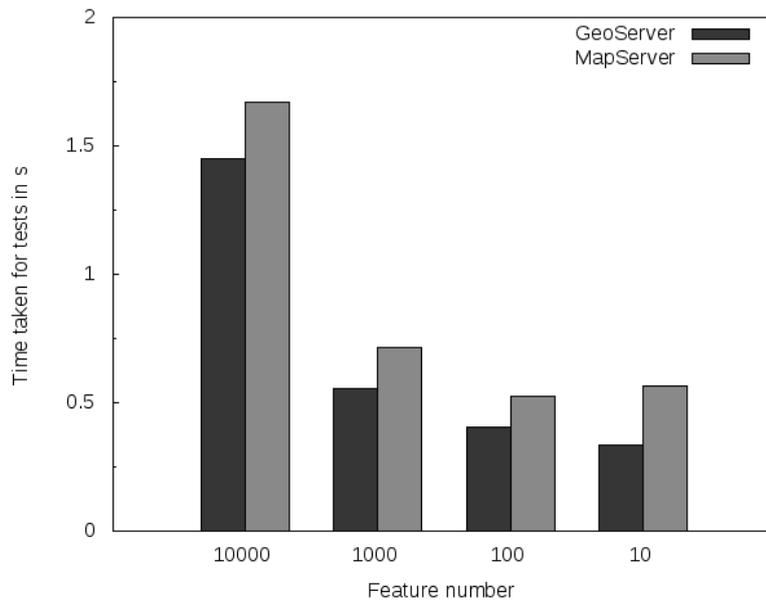


Server name	GeoServer	MapServer
Test id:	3	3
Concurrency level:	1	1
Time taken for tests:	0.403 seconds	0.523 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	328254 bytes	83004 bytes
Requests per seconds:	7.45 [# /sec] (mean)	5.74 [# /sec] (mean)
Time per request:	134.173 [ms] (mean)	174.267 [ms] (mean)
Transfer rate:	796.39 [Kbytes/sec] received	155.05 [Kbytes/sec] received

#### Test 4 - Test - Mappa restituita:



Server name	GeoServer	MapServer
Test id:	4	4
Concurrency level:	1	1
Time taken for tests:	0.336 seconds	0.566 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	72087 bytes	20947 bytes
Requests per	8.93 [#/sec] (mean)	5.30 [#/sec] (mean)
Time per request:	111.961 [ms] (mean) 111.961 [ms] (mean, across all concurrent requests)	188.729 [ms] (mean) 188.729 [ms] (mean, across all concurrent requests)
Transfer rate:	209.59 [Kbytes/sec] received	36.13 [Kbytes/sec] received



Dal grafico si evince che con richieste che coinvolgono SLD esterni, GeoServer si comporta meglio di MapServer, ciò è probabilmente dovuto al fatto che GeoServer, a differenza di MapServer, supporta nativamente lo standard SLD (il supporto all'estensione SLD per MapServer è stato aggiunto infatti solo a partire dalla versione 4.2)

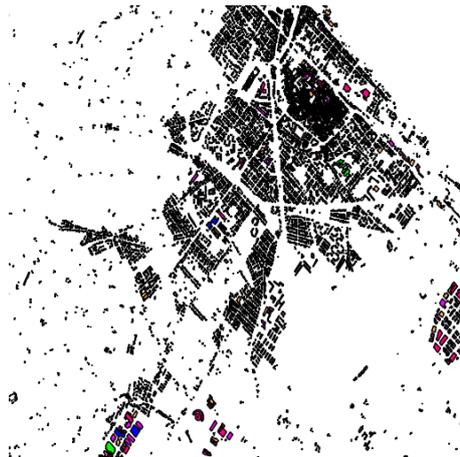
### **Terza sessione di test - Caratteristiche principali:**

Tipo di richiesta:	GetMap
Data source:	PostGIS table
Formato output:	image/png
Style:	file SLD che definisce i colori delle feature della mappa con dieci regole

### **Richiesta http effettuata**

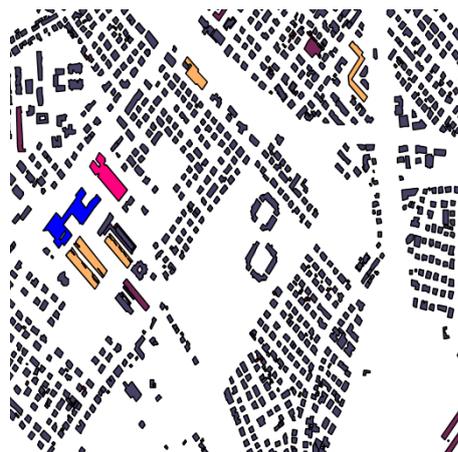
[http://192.168.42.3:8080/geoserver/wms?service=WMS&version=1.1.0&request=GetMap&layers=test:edifici&styles=&bbox=2357063.4483333323,4853164.6627976205,2362474.2530952375,4857269.81684524&width=512&height=512&sld=http://192.168.42.3/sld/edifici\\_ten\\_rules.sld&srs=EPSG:3004&format=image/png&UPDATESEQUENCE=6573](http://192.168.42.3:8080/geoserver/wms?service=WMS&version=1.1.0&request=GetMap&layers=test:edifici&styles=&bbox=2357063.4483333323,4853164.6627976205,2362474.2530952375,4857269.81684524&width=512&height=512&sld=http://192.168.42.3/sld/edifici_ten_rules.sld&srs=EPSG:3004&format=image/png&UPDATESEQUENCE=6573)

**Test 1 - Test - Mappa restituita:**



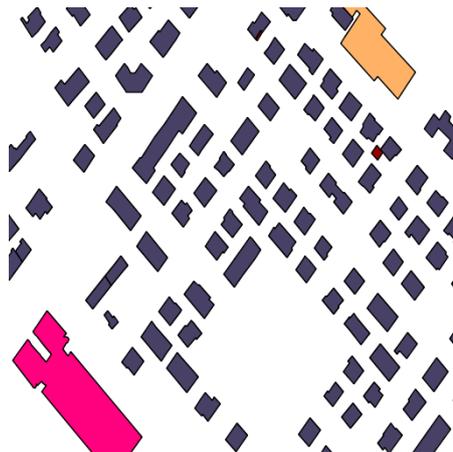
Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	1	1
Time taken for tests:	1.703 seconds	2.416 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	294570 bytes	136095 bytes
Requests per seconds:	1.76 [#/sec] (mean)	1.24 [#/sec] (mean)
Time per request:	567.832 [ms] (mean) 567.832 [ms] (mean, across all concurrent requests)	805.378 [ms] (mean) 805.378 [ms] (mean, across all concurrent requests)
Transfer rate:	168.87 [Kbytes/sec] received	55.01 [Kbytes/sec] received

Test 2 - Test - Mappa restituita:



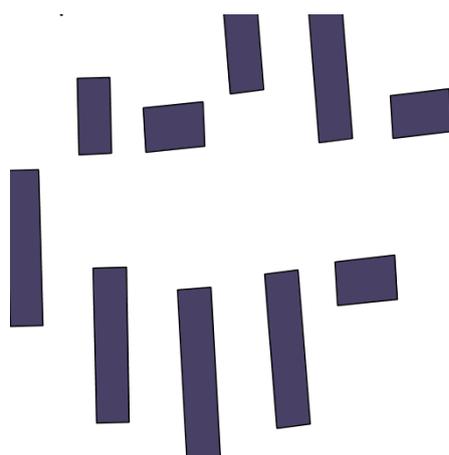
Server name	GeoServer	MapServer
Test id:	2	2
Concurrency level:	1	1
Time taken for tests:	0.570 seconds	0.833 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	490245 bytes	151689 bytes
Requests per seconds:	5.26 [#/sec] (mean)	3.60 [#/sec] (mean)
Time per request:	190.058 [ms] (mean) 190.058 [ms] (mean, across all concurrent requests)	277.514 [ms] (mean) 277.514 [ms] (mean, across all concurrent requests)
Transfer rate:	839.67 [Kbytes/sec] received	177.93 [Kbytes/sec] received

Test 3 - Test - Mappa restituita:

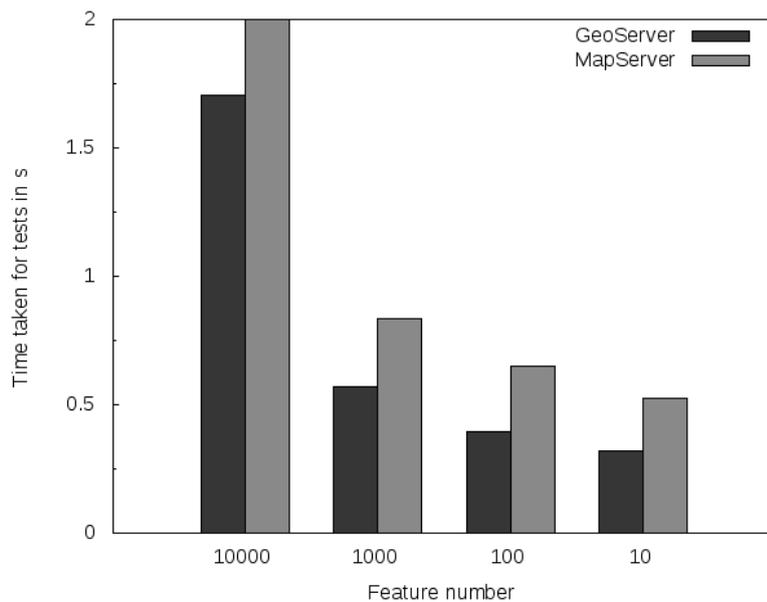


Server name	GeoServer	MapServer
<b>Test id:</b>	3	3
<b>Concurrency level:</b>	1	1
<b>Time taken for tests:</b>	0.393 seconds	0.648 seconds
<b>Complete requests:</b>	3	3
<b>Failed Requests:</b>	0	0
<b>Write errors:</b>	0	0
<b>Total transferred:</b>	308283 bytes	85092 bytes
<b>Requests per seconds:</b>	7.63 [# /sec] (mean)	4.63 [# /sec] (mean)
<b>Time per request:</b>	131.026 [ms] (mean) 131.026 [ms] (mean, across all concurrent requests)	215.902 [ms] (mean) 215.902 [ms] (mean, across all concurrent requests)
<b>Transfer rate:</b>	765.90 [Kbytes/sec] received	128.30 [Kbytes/sec] received

#### Test 4 - Test - Mappa restituita:



Server name	GeoServer	MapServer
Test id:	4	4
Concurrency level:	1	1
Time taken for tests:	0.318 seconds	0.523 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	65778 bytes	20871 bytes
Requests per seconds:	9.44 [#/sec] (mean)	5.74 [#/sec] (mean)
Time per request:	105.943 [ms] (mean) 105.943 [ms] (mean, across all concurrent requests)	174.298 [ms] (mean) 174.298 [ms] (mean, across all concurrent requests)
Transfer rate:	202.11 [Kbytes/sec] received	38.98 [Kbytes/sec] received



Ancora una volta GeoServer si mostra più performante di MapServer nella gestione di mappe con data source PostGIS e con la specifica di un file SLD, si conferma pertanto il comportamento osservato nel test precedente.

### Quarta sessione di test - Caratteristiche principali:

Tipo di richiesta:	GetLegendGraphic
Data source:	PostGIS
Formato output:	png 8
Style:	file SLD che definisce i colori delle feature della mappa con dieci regole

### **Richiesta http effettuata:**

http://192.168.42.3:8080/geoserver/wms?service=WMS&version=1.1.0&request=GetLegendGraphic&layer=test:edifici&sld=http://192.168.42.3/sld/edifici\_ten\_rules.sld&format=image/png&UPDATESEQUENCE=9589

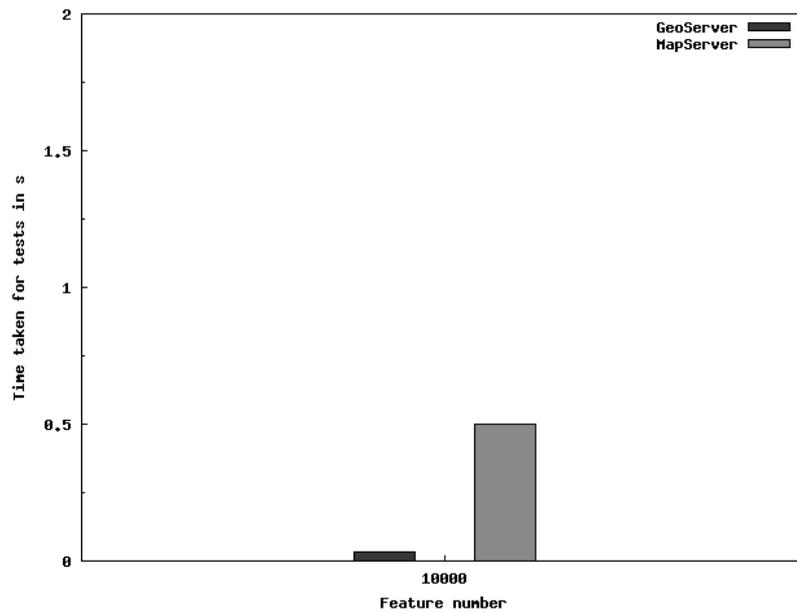
### **Legenda restituita:**



**Nota:** ad ogni colore corrisponde il numero della regola associata definita nel file .sld

Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	1	1
Time taken for tests:	0.034 seconds	0.501 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	4356 bytes	4431 bytes
Requests per seconds:	88.80 [#/sec] (mean)	5.98 [#/sec] (mean)
Time per request:	11.262 [ms] (mean) 11.262 [ms] (mean, across all concurrent requests)	167.150 [ms] (mean) 167.150 [ms] (mean, across all concurrent requests)
Transfer rate:	125.91 [Kbytes/sec] received	8.63 [Kbytes/sec] received

Si noti che per la richiesta GetLegendGraphic è stato fatto un solo test in quanto tale richiesta non prevede il parametro bounding box.



GeoServer impiega molto meno tempo a restituire la legenda.

Le quattro sessioni di test sopra riportate sono state ripetute inviando però al server 30 richieste, 10 alla volta (nel file di configurazione sono state quindi specificate le opzioni “numberOfRequests” e “concurrencyLevel” impostate rispettivamente a 30 e 10).

**Quinta sessione di test - Caratteristiche principali:**

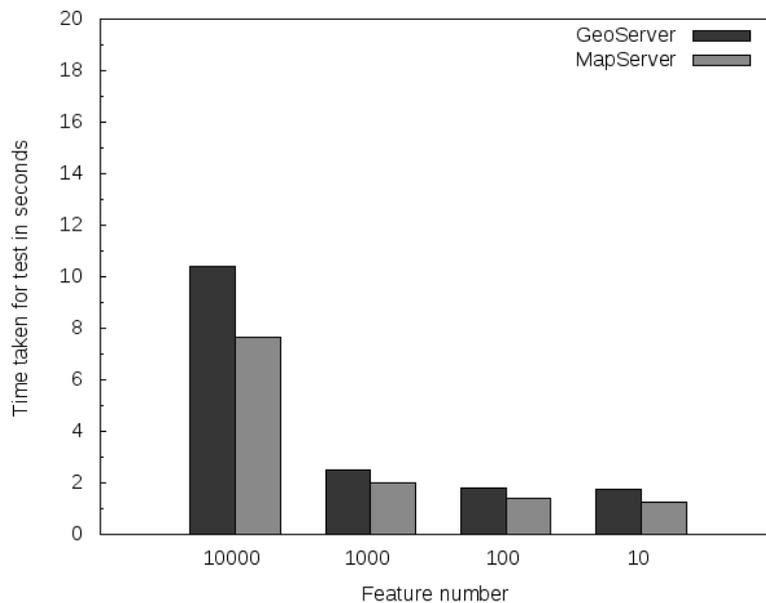
Tipo di richiesta:	GetMap
Data source:	PostGIS table
Formato output:	png 8
Style:	default
Numero di richieste:	30
Livello di concorrenza:	10

Server name	GeoServer	MapServer
<b>Test id:</b>	1	1
<b>Concurrency level:</b>	10	10
<b>Time taken for tests:</b>	10.391 seconds	7.630 seconds
<b>Complete requests:</b>	30	30
<b>Failed Requests:</b>	0	0
<b>Write errors:</b>	0	0
<b>Total transferred:</b>	2745870 bytes	1597530 bytes
<b>Requests per seconds:</b>	2.89 [#/sec] (mean)	3.93 [#/sec] (mean)
<b>Time per request:</b>	3463.503 [ms] (mean) 346.350 [ms] (mean, across all concurrent requests)	2543.452 [ms] (mean) 254.345 [ms] (mean, across all concurrent requests)
<b>Transfer rate:</b>	258.07 [Kbytes/sec] received	204.46 [Kbytes/sec] received

<b>Server name</b>	GeoServer	MapServer
<b>Test id:</b>	2	2
<b>Concurrency level:</b>	10	10
<b>Time taken for tests:</b>	2.479 seconds	2.001 seconds
<b>Complete requests:</b>	30	30
<b>Failed Requests:</b>	0	0
<b>Write errors:</b>	0	0
<b>Total transferred:</b>	4029270 bytes	2721030 bytes
<b>Requests per seconds:</b>	12.10 [#/sec] (mean)	14.99 [#/sec] (mean)
<b>Time per request:</b>	826.286 [ms] (mean) 82.629 [ms] (mean, across all concurrent requests)	666.983 [ms] (mean) 66.698 [ms] (mean, across all concurrent requests)
<b>Transfer rate:</b>	1587.36 [Kbytes/sec] received	1328.00 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	3	3
Concurrency level:	10	10
Time taken for tests:	1.794 seconds	1.389 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	2289240 bytes	1981020 bytes
Requests per seconds:	16.72 [#/sec] (mean)	21.60 [#/sec] (mean)
Time per request:	598.055 [ms] (mean) 59.806 [ms] (mean, across all concurrent requests)	462.963 [ms] (mean) 46.296 [ms] (mean, across all concurrent requests)
Transfer rate:	1246.03 [Kbytes/sec] received	1392.90 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	4	4
Concurrency level:	10	10
Time taken for tests:	1.739 seconds	1.259 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	608010 bytes	635645 bytes
Requests per seconds:	17.25 [#/sec] (mean)	23.84 [#/sec] (mean)
Time per request:	579.625 [ms] (mean) 57.962 [ms] (mean, across all concurrent requests)	419.505 [ms] (mean) 41.951 [ms] (mean, across all concurrent requests)
Transfer rate:	341.46 [Kbytes/sec] received	493.24 [Kbytes/sec] received



MapServer presenta tempi di risposta migliori quando i server sono chiamati a processare richieste multiple contemporaneamente.

### Sesta sessione di test - Caratteristiche principali:

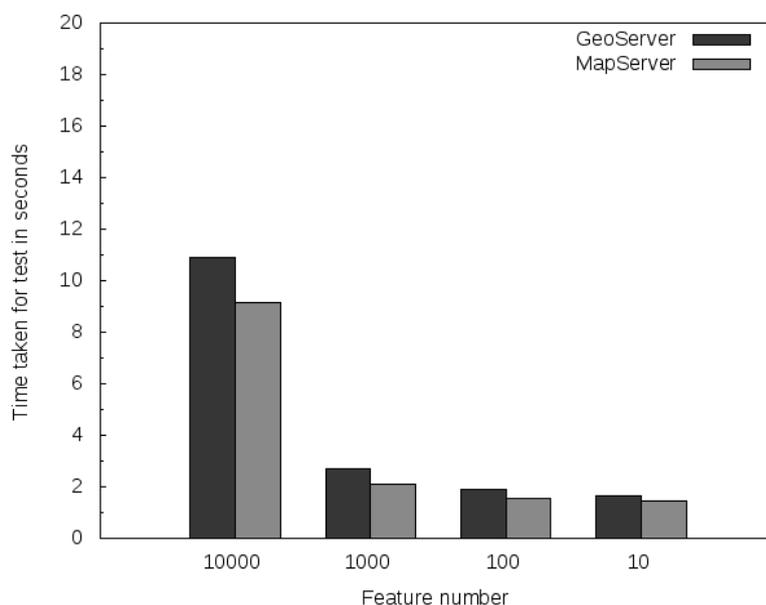
Tipo di richiesta:	GetMap
Data source:	PostGIS
Formato output:	png 8
Style:	file SLD che definisce il colore della mappa con una regola
Numero di richieste:	30
Livello di concorrenza:	10

Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	10	10
Time taken for tests:	10.913 seconds	9.169 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	6020550 bytes	1301831 bytes
Requests per seconds:	2.75 [# /sec] (mean)	3.27 [# /sec] (mean)
Time per request:	3637.822 [ms] (mean) 363.782 [ms] (mean, across all concurrent requests)	3056.425 [ms] (mean) 305.643 [ms] (mean, across all concurrent requests)
Transfer rate:	538.73 [Kbytes/sec] received	138.65 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	2	2
Concurrency level:	10	10
Time taken for tests:	2.706 seconds	2.109 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	6644310 bytes	1439730 bytes
Requests per seconds:	11.09 [#/sec] (mean)	14.22 [#/sec] (mean)
Time per request:	901.886 [ms] (mean) 90.189 [ms] (mean, across all concurrent requests)	703.081 [ms] (mean) 70.308 [ms] (mean, across all concurrent requests)
Transfer rate:	2398.15 [Kbytes/sec] received	666.58 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	3	3
Concurrency level:	10	10
Time taken for tests:	1.920 seconds	1.542 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	3282540 bytes	830477 bytes
Requests per seconds:	15.63 [#/sec] (mean)	19.45 [#/sec] (mean)
Time per request:	639.939 [ms] (mean) 63.994 [ms] (mean, across all concurrent requests)	514.029 [ms] (mean) 51.403 [ms] (mean, across all concurrent requests)
Transfer rate:	1669.75 [Kbytes/sec] received	525.92 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	4	4
Concurrency level:	10	10
Time taken for tests:	1.652 seconds	1.448 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	720870 bytes	209734 bytes
Requests per seconds:	18.16 [#/sec] (mean)	20.72 [#/sec] (mean)
Time per request:	550.685 [ms] (mean) 55.069 [ms] (mean, across all concurrent requests)	482.677 [ms] (mean) 48.268 [ms] (mean, across all concurrent requests)
Transfer rate:	426.12 [Kbytes/sec] received	141.45 [Kbytes/sec] received



Nel gestire richieste multiple concorrenti MapServer mostra un comportamento migliore di GeoServer anche nella gestione degli SLD.

### Settima sessione di test – Caratteristiche principali:

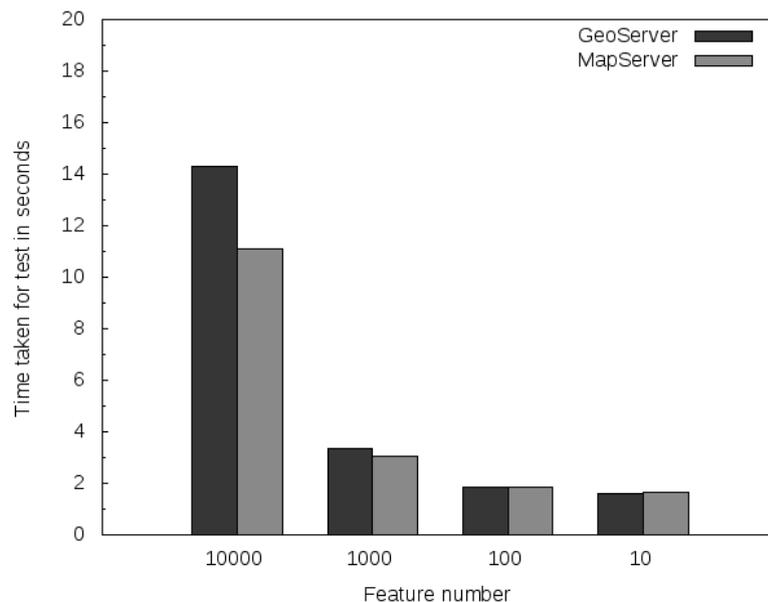
Tipo di richiesta:	GetMap
Data source:	PostGIS table
Formato output:	png 8
Style:	file SLD che definisce i colori delle feature della mappa con 10 regole
Numero di richieste:	30
Livello di concorrenza:	10

Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	10	10
Time taken for tests:	14.278 seconds	11.100 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	2945675 bytes	1360996 bytes
Requests per seconds:	2.10 [#/sec] (mean)	2.70 [#/sec] (mean)
Time per request:	4759.380 [ms] (mean) 475.938 [ms] (mean, across all concurrent requests)	3700.056 [ms] (mean) 370.006 [ms] (mean, across all concurrent requests)
Transfer rate:	201.47 [Kbytes/sec] received	119.74 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	2	2
Concurrency level:	10	10
Time taken for tests:	3.360 seconds	3.046 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	4902450 bytes	1516890 bytes
Requests per seconds:	8.93 [#/sec] (mean)	9.85 [#/sec] (mean)
Time per request:	1120.015 [ms] (mean) 112.001 [ms] (mean, across all concurrent requests)	1015.346 [ms] (mean) 101.535 [ms] (mean, across all concurrent requests)
Transfer rate:	1424.85 [Kbytes/sec] received	486.32 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	3	3
Concurrency level:	10	10
Time taken for tests:	1.855 seconds	1.843 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	3082830 bytes	851288 bytes
Requests per seconds:	16.17 [#/sec] (mean)	16.28 [#/sec] (mean)
Time per request:	618.438 [ms] (mean) 61.844 [ms] (mean, across all concurrent requests)	614.436 [ms] (mean) 61.444 [ms] (mean, across all concurrent requests)
Transfer rate:	1622.68 [Kbytes/sec] received	451.00 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	4	4
Concurrency level:	10	10
Time taken for tests:	1.603 seconds	1.635 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	657780 bytes	209172 bytes
Requests per seconds:	18.71 [#/sec] (mean)	18.35 [#/sec] (mean)
Time per request:	534.453 [ms] (mean) 53.445 [ms] (mean, across all concurrent requests)	544.973 [ms] (mean) 54.497 [ms] (mean, across all concurrent requests)
Transfer rate:	400.64 [Kbytes/sec] received	124.94 [Kbytes/sec] received

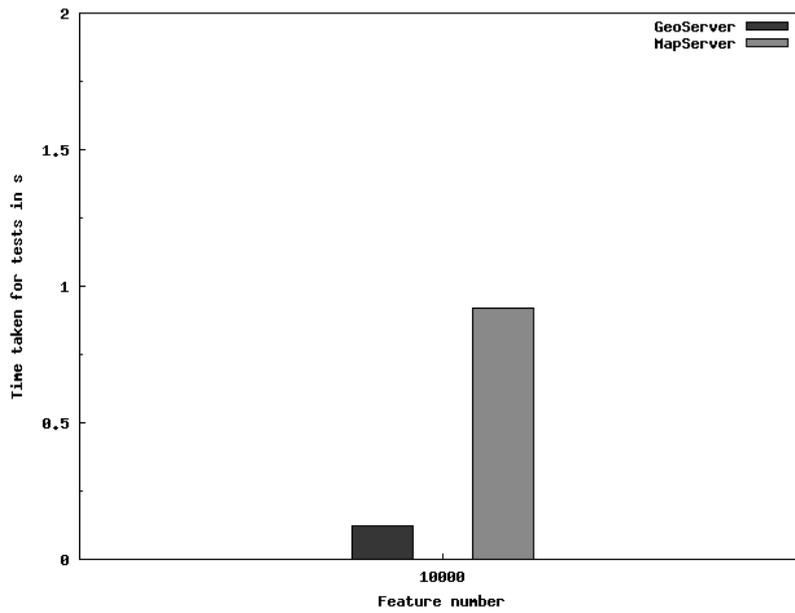


MapServer si mostra in questo caso più performante quando il numero di feature da visualizzare è piuttosto elevato. Al diminuire delle feature visualizzate, i tempi di risposta dei due server sono pressoché gli stessi.

### Ottava sessione di test – Caratteristiche principali:

Tipo di richiesta:	GetLegendGraphic
Data source:	PostGIS
Formato output:	png 8
Style:	file SLD che definisce i colori delle feature della mappa con dieci regole
Numero di richieste:	30
Livello di concorrenza:	10

Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	10	10
Time taken for tests:	0.135 seconds	1.060 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	43560 bytes	44882 bytes
Requests per seconds:	223.04 [# /sec] (mean)	28.31 [# /sec] (mean)
Time per request:	44.834 [ms] (mean) 4.483 [ms] (mean, across all concurrent requests)	353.216 [ms] (mean) 35.322 [ms] (mean, across all concurrent requests)
Transfer rate:	316.27 [Kbytes/sec] received	41.36 [Kbytes/sec] received



Come per le richieste singole, GeoServer si mostra più veloce nel servire la richiesta GetLegendGrafic.

## 5.4.2 SECONDA MACROSESSIONE DI TEST

Vediamo ora gli stessi test effettuati però con data source shapefile. Riportiamo di nuovo la richiesta http in quanto cambia il layer richiesto.

Anche qui valgono le considerazioni fatte sul parametro bounding box nella prima macro sessione di test.

### Nona sessione di test - Caratteristiche principali:

Tipo di richiesta:	GetMap
Data source:	shapefile
Formato output:	png 8 bit
Style:	default

### Richiesta HTTP effettuata:

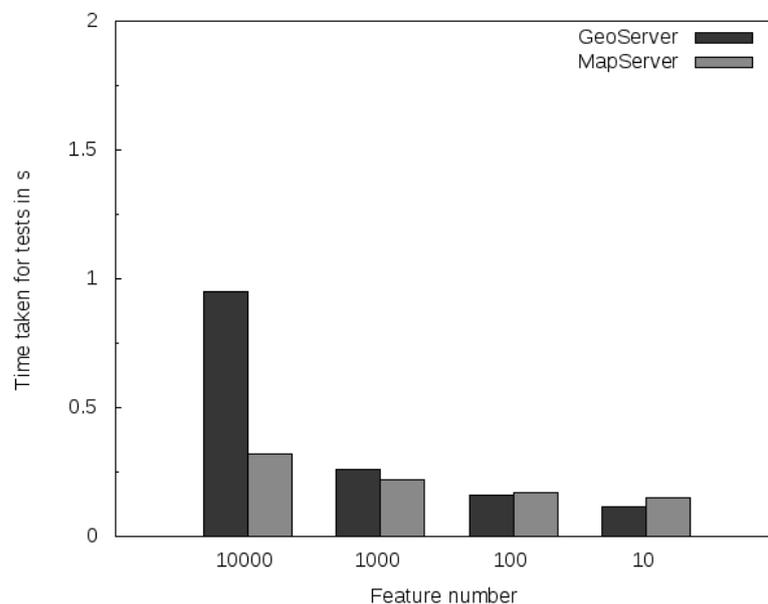
<http://192.168.42.3:8080/geoserver/wms?service=WMS&version=1.1.0&request=GetMap&layers=test:shapeEdifici&styles=&bbox=2357063.4483333323,4853164.6627976205,2362474.2530952375,4857269.81684524&width=512&height=512&srs=EPSG:3004&format=image/png&UPDATESEQUENCE=1014>

Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	1	1
Time taken for tests:	0.951 seconds	0.319 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	274416 bytes	162169 bytes
Requests per seconds:	3.16 [#/sec] (mean)	9.40 [#/sec] (mean)
Time per request:	316.834 [ms] (mean) 316.834 [ms] (mean, across all concurrent requests)	106.376 [ms] (mean) 106.376 [ms] (mean, across all concurrent requests)
Transfer rate:	281.94 [Kbytes/sec] received	496.25 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	2	2
Concurrency level:	1	1
Time taken for tests:	0.259 seconds	0.222 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	402927 bytes	280179 bytes
Requests per seconds:	11.58 [#/sec] (mean)	13.53 [#/sec] (mean)
Time per request:	86.352 [ms] (mean) 86.352 [ms] (mean, across all concurrent requests)	73.894 [ms] (mean) 73.894 [ms] (mean, across all concurrent requests)
Transfer rate:	1518.91 [Kbytes/sec] received	1234.25 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	3	3
Concurrency level:	1	1
Time taken for tests:	0.160 seconds	0.171 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	228924 bytes	198879 bytes
Requests per seconds:	18.75 [#/sec] (mean)	17.51 [#/sec] (mean)
Time per request:	53.335 [ms] (mean) 53.335 [ms] (mean, across all concurrent requests)	57.105 [ms] (mean) 57.105 [ms] (mean, across all concurrent requests)
Transfer rate:	1397.19 [Kbytes/sec] received	1133.69 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	4	4
Concurrency level:	1	1
Time taken for tests:	0.115 seconds	0.148 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	60801 bytes	64354 bytes
Requests per seconds:	26.08 [#/sec] (mean)	20.23 [#/sec] (mean)
Time per request:	38.346 [ms] (mean) 38.346 [ms] (mean, across all concurrent requests)	49.425 [ms] (mean) 49.425 [ms] (mean, across all concurrent requests)
Transfer rate:	516.15 [Kbytes/sec] received	423.84 [Kbytes/sec] received



Nella gestione degli shape, MapServer si mostra più performante solo quando il numero di feature da visualizzare è elevato (10000 e 1000).

### Decima sessione di test - Caratteristiche principali:

Tipo di richiesta:	GetMap
Data source:	shapefile
Formato output:	png 8
Style:	file SLD che definisce il colore della mappa

### Richiesta http effettuata:

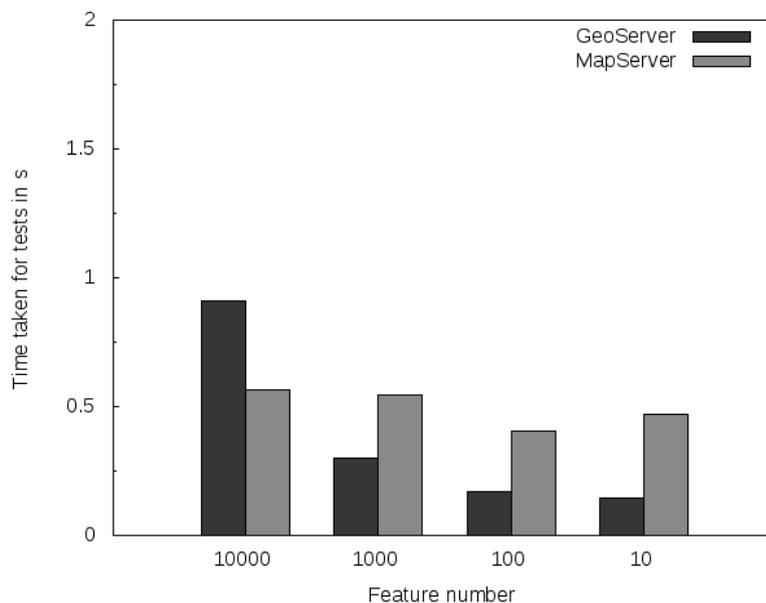
http://192.168.42.3:8080/geoserver/wms?service=WMS&version=1.1.0&request=GetMap&layers=test:shapeEdifici&styles=&bbox=2357063.45,4853164.66,2362474.25,4857269.82&width=512&height=512&sld=http://192.168.42.3/sld/edifici\_one\_rule.sld&srs=EPSG:3004&format=image/png&UPDATESEQUENCE=4778

Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	1	1
Time taken for tests:	0.912 seconds	0.564 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	603012 bytes	130190 bytes
Requests per seconds:	3.29 [#/sec] (mean)	5.32 [#/sec] (mean)
Time per request:	304.122 [ms] (mean) 304.122 [ms] (mean, across all concurrent requests)	188.103 [ms] (mean) 188.103 [ms] (mean, across all concurrent requests)
Transfer rate:	645.44 [Kbytes/sec] received	225.30 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	2	2
Concurrency level:	1	1
Time taken for tests:	0.300 seconds	0.546 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	664431 bytes	143990 bytes
Requests per seconds:	9.99 [#/sec] (mean)	5.49 [#/sec] (mean)
Time per request:	100.073 [ms] (mean) 100.073 [ms] (mean, across all concurrent requests)	182.067 [ms] (mean) 182.067 [ms] (mean, across all concurrent requests)
Transfer rate:	2161.28 [Kbytes/sec] received	257.44 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	3	3
Concurrency level:	1	1
Time taken for tests:	0.170 seconds	0.404 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	328254 bytes	83050 bytes
Requests per seconds:	17.68 [#/sec] (mean)	7.42 [#/sec] (mean)
Time per request:	56.545 [ms] (mean) 56.545 [ms] (mean, across all concurrent requests)	134.737 [ms] (mean) 134.737 [ms] (mean, across all concurrent requests)
Transfer rate:	1889.70 [Kbytes/sec] received	200.65 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	4	4
Concurrency level:	1	1
Time taken for tests:	0.147 seconds	0.468 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	72087 bytes	20969 bytes
Requests per seconds:	20.48 [#/sec] (mean)	6.41 [#/sec] (mean)
Time per request:	48.836 [ms] (mean) 48.836 [ms] (mean, across all concurrent requests)	156.056 [ms] (mean) 156.056 [ms] (mean, across all concurrent requests)
Transfer rate:	480.51 [Kbytes/sec] received	43.74 [Kbytes/sec] received



Anche in questo caso, MapServer risulta più veloce nel servire le richieste solo quando il numero di feature da visualizzare è elevato (nell'analogica sessione di test con data source PostGIS, MapServer si presentava meno performante di Geoserver anche in questo caso, quindi si comincia ad osservare un miglioramento delle prestazioni di Mapserver anche nella gestione di poche richieste quando si una come data source shapefile)

## Undicesima sessione di test - Caratteristiche principali:

Tipo di richiesta:	GetMap
Data source:	shapefile
Formato output:	png 8
Style:	file SLD che definisce i colori delle feature della mappa con dieci regole

### Richiesta http effettuata

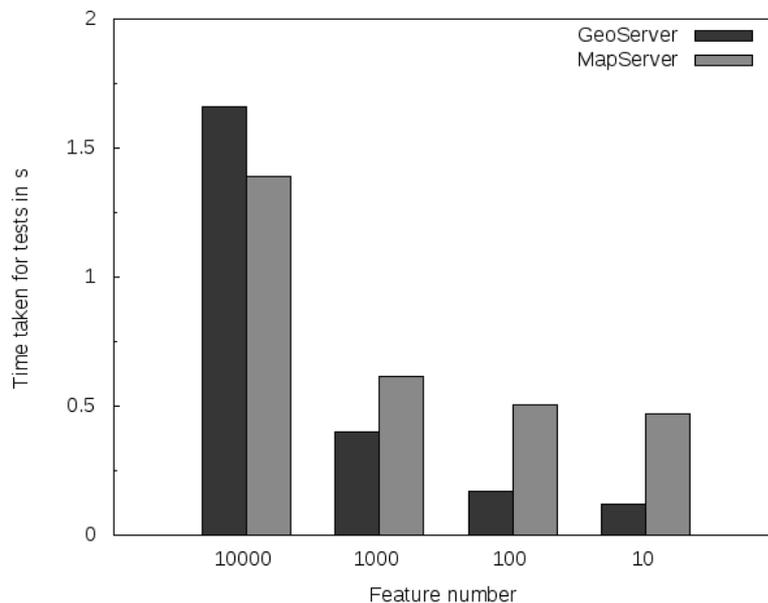
http://192.168.42.3:8080/geoserver/wms?service=WMS&version=1.1.0&request=GetMap&layers=test:shapeEdifici&styles=&bbox=2357063.4483333323,4853164.6627976205,2362474.2530952375,4857269.81684524&width=512&height=512&sld=http://192.168.42.3/sld/edifici\_ten\_rules.sld&srs=EPSG:3004&format=image/png&UPDATESEQUENCE=8645

Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	1	1
Time taken for tests:	1.661 seconds	1.388 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	294216 bytes	136141 bytes
Requests per seconds:	1.81 [#/sec] (mean)	2.16 [#/sec] (mean)
Time per request:	553.831 [ms] (mean) 553.831 [ms] (mean, across all concurrent requests)	462.502 [ms] (mean) 462.502 [ms] (mean, across all concurrent requests)
Transfer rate:	172.93 [Kbytes/sec] received	95.82 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	2	2
Concurrency level:	1	1
Time taken for tests:	0.399 seconds	0.613 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	490245 bytes	151706 bytes
Requests per seconds:	7.53 [#/sec] (mean)	4.89 [#/sec] (mean)
Time per request:	132.853 [ms] (mean) 132.853 [ms] (mean, across all concurrent requests)	204.411 [ms] (mean) 204.411 [ms] (mean, across all concurrent requests)
Transfer rate:	1201.21 [Kbytes/sec] received	241.59 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	3	3
Concurrency level:	1	1
Time taken for tests:	0.172 seconds	0.505 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	308283 bytes	85138 bytes
Requests per seconds:	17.40 [#/sec] (mean)	5.94 [#/sec] (mean)
Time per request:	57.477 [ms] (mean) 57.477 [ms] (mean, across all concurrent requests)	168.231 [ms] (mean) 168.231 [ms] (mean, across all concurrent requests)
Transfer rate:	1745.95 [Kbytes/sec] received	164.74 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	4	4
Concurrency level:	1	1
Time taken for tests:	0.118 seconds	0.470 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	65778 bytes	20893 bytes
Requests per seconds:	25.37 [#/sec] (mean)	6.38 [#/sec] (mean)
Time per request:	39.418 [ms] (mean) 39.418 [ms] (mean, across all concurrent requests)	156.824 [ms] (mean) 156.824 [ms] (mean, across all concurrent requests)
Transfer rate:	543.20 [Kbytes/sec] received	43.37 [Kbytes/sec] received



Anche in questa sessione di test, in modo analogo alla precedente, MapServer si mostra più performante solo quando il numero di feature da visualizzare è elevato.

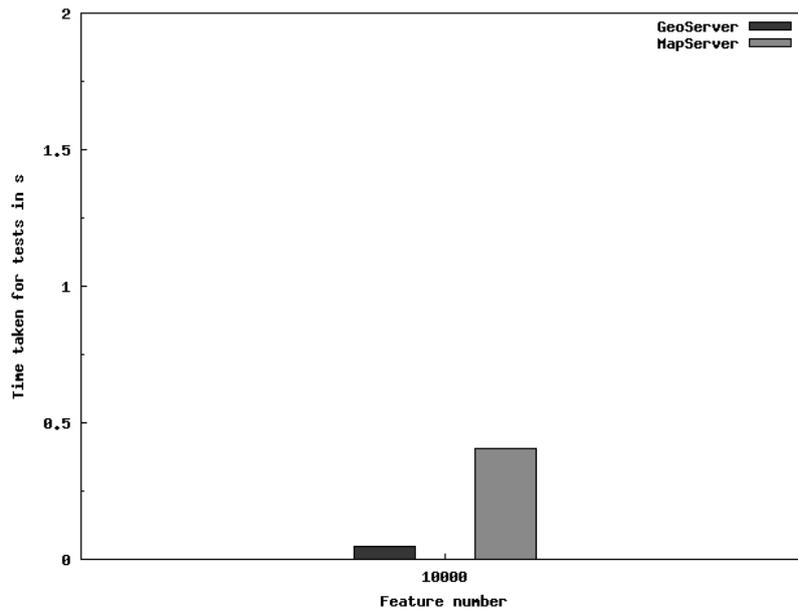
## Dodicesima sessione di test - Caratteristiche principali:

Tipo di richiesta:	GetLegendGraphic
Data source:	shapefile
Formato output:	png 8
Style:	file SLD che definisce i colori delle feature della mappa con dieci regole

### Richiesta http effettuata:

[http://192.168.42.3:8080/geoserver/wms?service=WMS&version=1.1.0&request=GetLegendGraphic&layer=test:shapeEdifici&sld=http://192.168.42.3/sld/edifici\\_ten\\_rules.sld&format=image/png&UPDATESEQUENCE=6969](http://192.168.42.3:8080/geoserver/wms?service=WMS&version=1.1.0&request=GetLegendGraphic&layer=test:shapeEdifici&sld=http://192.168.42.3/sld/edifici_ten_rules.sld&format=image/png&UPDATESEQUENCE=6969)

Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	1	1
Time taken for tests:	0.048 seconds	0.404 seconds
Complete requests:	3	3
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	4356 bytes	4497 bytes
Requests per seconds:	62.34 [#/sec] (mean)	7.43 [#/sec] (mean)
Time per request:	16.042 [ms] (mean) 16.042 [ms] (mean, across all concurrent requests)	134.645 [ms] (mean) 134.645 [ms] (mean, across all concurrent requests)
Transfer rate:	88.39 [Kbytes/sec] received	10.87 [Kbytes/sec] received



GeoServer conferma il comportamento dei test precedenti nel gestire la richiesta GetLegendGraphic.

Di seguito sono ripetute le quattro sessioni di test precedenti ma con un numero di richieste pari a 30 e livello di concorrenza 10.

### Tredicesima sessione di test - Caratteristiche principali:

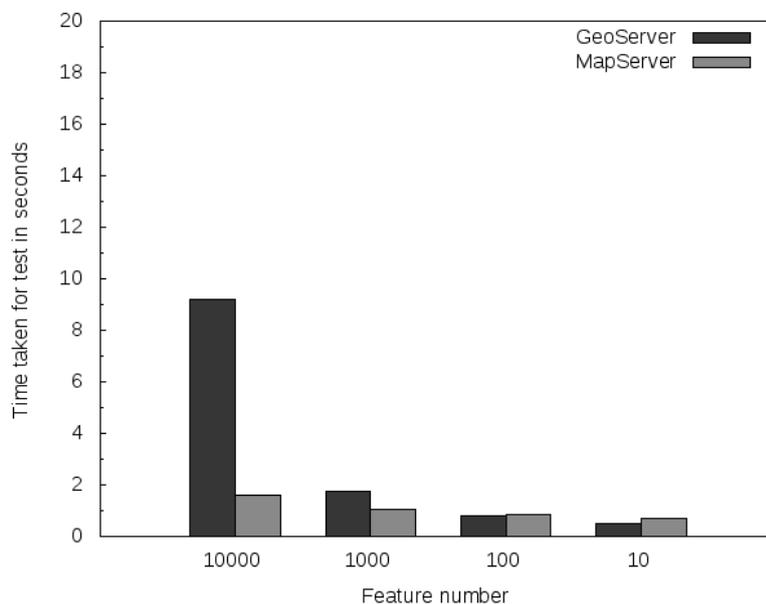
Tipo di richiesta:	GetMap
Data source:	shapefile
Formato output:	png 8
Style:	default
Numero di richieste:	30
Livello di concorrenza:	10

Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	10	10
Time taken for tests:	9.191 seconds	1.584 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	2744160 bytes	1621253 bytes
Requests per seconds:	3.26 [#/sec] (mean)	18.94 [#/sec] (mean)
Time per request:	3063.550 [ms] (mean) 306.355 [ms] (mean, across all concurrent requests)	527.893 [ms] (mean) 52.789 [ms] (mean, across all concurrent requests)
Transfer rate:	291.58 [Kbytes/sec] received	999.73 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	2	2
Concurrency level:	10	10
Time taken for tests:	1.757 seconds	1.062 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	4029270 bytes	2801790 bytes
Requests per seconds:	17.08 [#/sec] (mean)	28.26 [#/sec] (mean)
Time per request:	585.650 [ms] (mean) 58.565 [ms] (mean, across all concurrent requests)	353.898 [ms] (mean) 35.390 [ms] (mean, across all concurrent requests)
Transfer rate:	2239.58 [Kbytes/sec] received	2577.13 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	3	3
Concurrency level:	10	10
Time taken for tests:	0.797 seconds	0.853 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	2289240 bytes	1988790 bytes
Requests per seconds:	37.63 [#/sec] (mean)	35.17 [#/sec] (mean)
Time per request:	265.724 [ms] (mean) 26.572 [ms] (mean, across all concurrent requests)	284.328 [ms] (mean) 28.433 [ms] (mean, across all concurrent requests)
Transfer rate:	2804.40 [Kbytes/sec] received	2276.92 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	4	4
Concurrency level:	10	10
Time taken for tests:	0.479 seconds	0.697 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	608010 bytes	643494 bytes
Requests per seconds:	62.62 [#/sec] (mean)	43.03 [#/sec] (mean)
Time per request:	159.703 [ms] (mean) 15.970 [ms] (mean, across all concurrent requests)	232.378 [ms] (mean) 23.238 [ms] (mean, across all concurrent requests)
Transfer rate:	1239.30 [Kbytes/sec] received	901.42 [Kbytes/sec] received



Nella gestione di richieste multiple concorrenti, MapServer continua a presentare tempi di risposta migliori quando il numero di feature da visualizzare è elevato. Per poche feature da visualizzare, i tempi di risposta dei due server si equivalgono.

### Quattordicesima sessione di test - Caratteristiche principali:

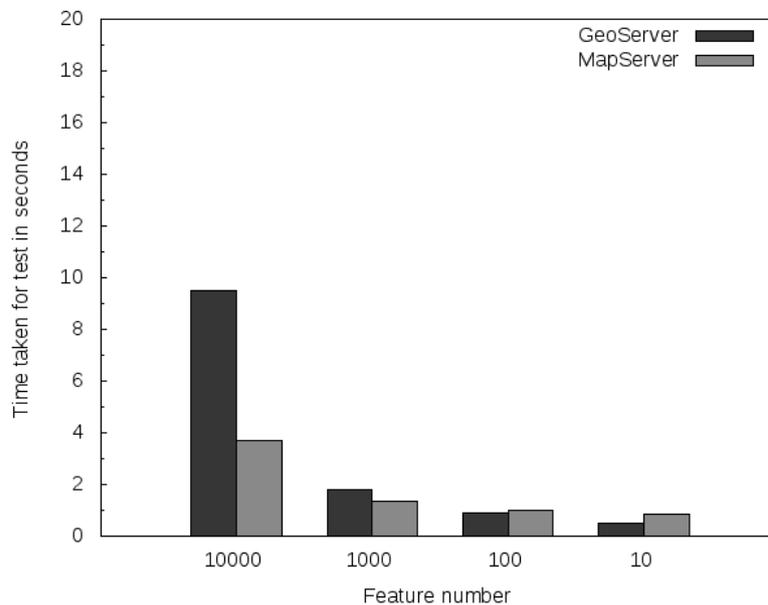
Tipo di richiesta:	GetMap
Data source:	shapefile
Formato output:	png 8
Style:	file SLD che definisce il colore della mappa con una regola
Numero di richieste:	30
Livello di concorrenza:	10

Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	10	10
Time taken for tests:	9.503 seconds	3.710 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	6030095 bytes	1302107 bytes
Requests per seconds:	3.16 [#/sec] (mean)	8.09 [#/sec] (mean)
Time per request:	3167.523 [ms] (mean) 316.752 [ms] (mean, across all concurrent requests)	1236.791 [ms] (mean) 123.679 [ms] (mean, across all concurrent requests)
Transfer rate:	619.70 [Kbytes/sec] received	342.71 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	2	2
Concurrency level:	10	10
Time taken for tests:	1.825 seconds	1.340 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	6644310 bytes	1439785 bytes
Requests per seconds:	16.44 [#/sec] (mean)	22.39 [#/sec] (mean)
Time per request:	608.262 [ms] (mean) 60.826 [ms] (mean, across all concurrent requests)	446.591 [ms] (mean) 44.659 [ms] (mean, across all concurrent requests)
Transfer rate:	3555.81 [Kbytes/sec] received	1049.46 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	3	3
Concurrency level:	10	10
Time taken for tests:	0.887 seconds	1.004 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	3282540 bytes	830477 bytes
Requests per seconds:	33.81 [#/sec] (mean)	29.89 [#/sec] (mean)
Time per request:	295.736 [ms] (mean) 29.574 [ms] (mean, across all concurrent requests)	334.528 [ms] (mean) 33.453 [ms] (mean, across all concurrent requests)
Transfer rate:	3613.14 [Kbytes/sec] received	808.12 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	4	4
Concurrency level:	10	10
Time taken for tests:	0.480 seconds	0.838 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	720870 bytes	209690 bytes
Requests per seconds:	62.44 [#/sec] (mean)	35.82 [#/sec] (mean)
Time per request:	160.145 [ms] (mean) 16.015 [ms] (mean, across all concurrent requests)	279.211 [ms] (mean) 27.921 [ms] (mean, across all concurrent requests)
Transfer rate:	1465.29 [Kbytes/sec] received	244.47 [Kbytes/sec] received



Analogamente al test precedente, MapServer si mostra più performante rispetto a Geoserver nel gestire richieste multiple concorrenti quando il numero di feature da visualizzare è consistente.

### Quindicesima sessione di test - Caratteristiche principali:

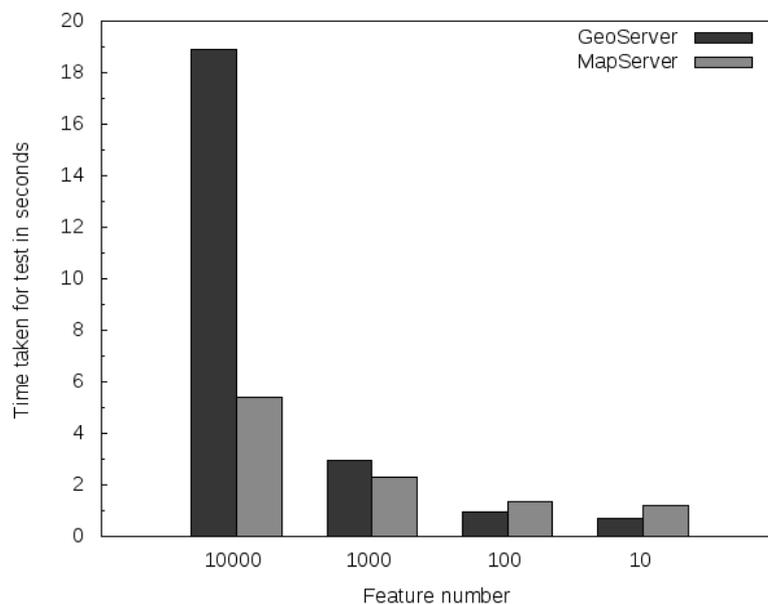
Tipo di richiesta:	GetMap
Data source:	shapefile
Formato output:	png 8
Style:	file SLD che definisce i colori delle feature della mappa con dieci regole
Numero di richieste:	30
Livello di concorrenza:	10

Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	10	10
Time taken for tests:	18.906 seconds	5.392 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	2942160 bytes	1361226 bytes
Requests per seconds:	1.59 [#/sec] (mean)	5.56 [#/sec] (mean)
Time per request:	6301.897 [ms] (mean) 630.190 [ms] (mean, across all concurrent requests)	1797.190 [ms] (mean) 179.719 [ms] (mean, across all concurrent requests)
Transfer rate:	151.98 [Kbytes/sec] received	246.56 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	2	2
Concurrency level:	10	10
Time taken for tests:	2.933 seconds	2.322 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	4902450 bytes	1516922 bytes
Requests per seconds:	10.23 [#/sec] (mean)	12.92 [#/sec] (mean)
Time per request:	977.637 [ms] (mean) 97.764 [ms] (mean, across all concurrent requests)	774.118 [ms] (mean) 77.412 [ms] (mean, across all concurrent requests)
Transfer rate:	1632.35 [Kbytes/sec] received	637.87 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	3	3
Concurrency level:	10	10
Time taken for tests:	0.964 seconds	1.348 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	3082830 bytes	851403 bytes
Requests per seconds:	31.11 [#/sec] (mean)	22.25 [#/sec] (mean)
Time per request:	321.444 [ms] (mean) 32.144 [ms] (mean, across all concurrent requests)	449.490 [ms] (mean) 44.949 [ms] (mean, across all concurrent requests)
Transfer rate:	3121.93 [Kbytes/sec] received	616.59 [Kbytes/sec] received

Server name	GeoServer	MapServer
Test id:	4	4
Concurrency level:	10	10
Time taken for tests:	0.677 seconds	1.188 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	665020 bytes	209172 bytes
Requests per seconds:	44.29 [#/sec] (mean)	25.25 [#/sec] (mean)
Time per request:	225.773 [ms] (mean) 22.577 [ms] (mean, across all concurrent requests)	396.005 [ms] (mean) 39.601 [ms] (mean, across all concurrent requests)
Transfer rate:	958.83 [Kbytes/sec] received	171.94 [Kbytes/sec] received

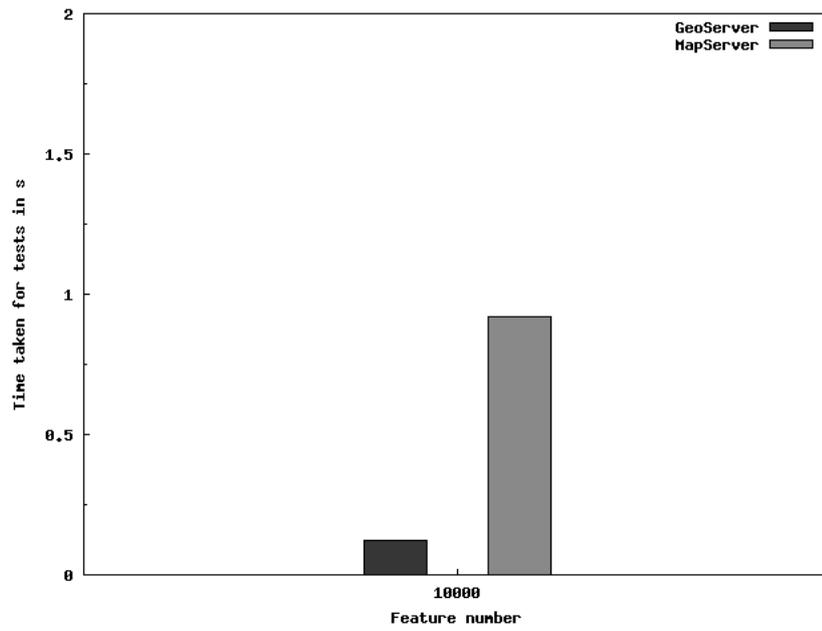


Comportamento analogo al test precedente.

### Sedicesima sessione di test – Caratteristiche principali:

Tipo di richiesta:	GetLegendGraphic
Data source:	shapefile
Formato output:	png 8
Style:	file SLD che definisce i colori delle feature della mappa con dieci regole
Numero di richieste:	30
Livello di concorrenza:	10

Server name	GeoServer	MapServer
Test id:	1	1
Concurrency level:	10	10
Time taken for tests:	0.123 seconds	0.922 seconds
Complete requests:	30	30
Failed Requests:	0	0
Write errors:	0	0
Total transferred:	43560 bytes	44882 bytes
Requests per seconds:	243.32 [#/sec] (mean)	32.55 [#/sec] (mean)
Time per request:	41.098 [ms] (mean) 4.110 [ms] (mean, across all concurrent requests)	307.255 [ms] (mean) 30.726 [ms] (mean, across all concurrent requests)
Transfer rate:	345.02 [Kbytes/sec] received	47.55 [Kbytes/sec] received



GeoServer conferma il comportamento mostrato finora con la richiesta GetLegendGraphic.

# 6 CONCLUSIONI

---

## 6.1 RISULTATI CONSEGUITI

Dalla prima macrosessione di test, che coinvolge test effettuati con data source PostGIS, si evince che:

- effettuando tre richieste alla volta, una di seguito all'altra, MapServer mostra performance migliori solo quando riceve richieste di mappe da tematizzare con lo stile di default. GeoServer sembrerebbe quindi più performante nella gestione degli SLD e di dati con data source PostGIS.
- effettuando invece 30 richieste, 10 alla volta, MapServer presenta tempi di risposta decisamente migliori rispetto a GeoServer, eccetto nel caso della richiesta GetLegendGraphic di importanza trascurabile.

Dalla seconda macrosessione di test, che coinvolge test effettuati con data source shapefile, si evince che:

- effettuando tre richieste alla volta, una di seguito all'altra, MapServer mostra performance migliori solo quando il numero di feature da restituire è elevato, sia quando per la tematizzazione della mappa si usa lo stile di default, sia quando per la tematizzazione della mappa si usa un file SLD definito dall'utente;
- effettuando invece 30 richieste, 10 alla volta, MapServer presenta tempi di risposta nettamente migliori rispetto a GeoServer quando il numero di feature da visualizzare è abbastanza alto (10000 e 1000).

In generale si può concludere quindi, che dall'analisi dei test qui discussi, MapServer mostra una migliore gestione della concorrenza, considerando anche che per GeoServer si sono rese necessarie configurazioni più complesse prima di osservare un miglioramento delle prestazioni in risposta a richieste multiple e concorrenti (si veda paragrafo 5.2.1).

Si tratta comunque di risultati che non possono essere facilmente generalizzati, in quanto condizionati dalle tecnologie utilizzate e dalle scelte di configurazione fatte.

## 6.2 PROBLEMI APERTI

Il software realizzato fornisce solo uno strumento per il benchmarking, realizzato per le esigenze specifiche di un'azienda e non un metodo generale e definitivo; testare le performance di una web application non è semplice, è difficile infatti riuscire a simulare con precisione il workload che una web application dovrà gestire, tener conto di tutte le variabili che possono compromettere o in qualche modo influenzare le performance, così com'è difficile riassumere e interpretare i risultati.

Lo strumento realizzato è quindi solo una base da cui partire che presenta molti margini di miglioramento. Nel corso del progetto si è cercato infatti di fare tutto il possibile per mettere entrambi i server geospaziali nelle condizioni di offrire le performance migliori

ma è necessario indagare ulteriormente su quali impostazioni di configurazione sia del server HTTP che dei server geospaziali possono influenzare le performance.

### **6.3 SVILUPPI FUTURI**

Nella problematica qui presentata, sono stati presi in considerazione le performance dei server geospaziali GeoServer e MapServer, sviluppi futuri possono riguardare l'analisi delle performance di altri server geospaziali come Deegree e ArcGIS Server.

Inoltre, nei test è stato coinvolto per motivi di tempo solo il servizio WMS (di principale interesse per il progetto), sviluppi futuri potrebbero pertanto prevedere l'estensione dei test ad altri Web Services come WFS.

Infine, la concorrenza simulata nei test ha coinvolto sempre la stessa richiesta, uno sviluppo interessante potrebbe riguardare la simulazione di più richieste concorrenti diverse tra loro, realizzabile attraverso la predisposizione di più pc che inviano richieste diverse simultaneamente ma che coinvolgono la stessa mappa.

## **RIFERIMENTI**

<http://www.w3.org>

<http://www.opengeospatial.org/standards>

<http://httpd.apache.org/docs/2.3/programs/ab.html>

<http://mapserver.org/>

<http://geoserver.org/>

<http://openlayers.org/>

## **ACRONIMI UTILIZZATI**

**CGI** - Common Gateway Interface

**ESRI** - Environmental Systems Research Institute

**FOSS** – Free Open Source Software

**GIS** - Geographic(al) Information System

**GML** - Geographic Markup Language

**HTTP** - HiperText Transfer protocol

**OASIS** - Organization for the Advancement of Structured Information Standards

**OGC** - Open Geospatial Consortium

**OWS** - OGC Web Services

**SLD** - Styled Layer Descriptor

**SOA** - Service Oriented Architecture

**SSL** - Secure Sockets Layers

**TSL** - Transport Layer security

**UDDI** - Universal Description Discovery and Integration

**WCS** - Web Coverage Service

**WFS** - Web Feature Service

**WMS** - Web Map Service

**WSDL** - Web Services Definition Language

**XML** - eXtensible Markup Language

## **RINGRAZIAMENTI**

Ringrazio innanzitutto coloro che mi hanno seguito durante l'attività di tirocinio e nella realizzazione di questo elaborato.

Un immenso grazie ai miei genitori e a mia sorella per essermi sempre stati vicini e per il loro immancabile aiuto in ogni momento del mio percorso universitario.

Ringrazio infine i miei parenti e i miei amici per il sostegno che mi hanno dato e per l'affetto che mi hanno sempre dimostrato.