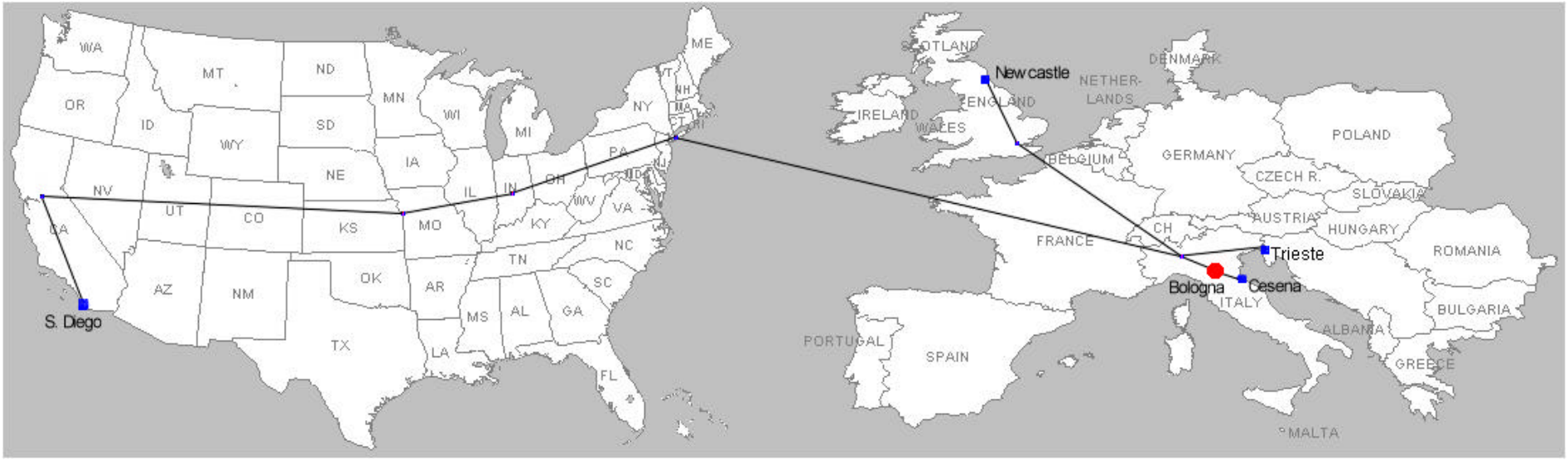
C2LD
un meccanismo per garantire “responsiveness” dei servizi Web

Vittorio Ghini, Fabio Panzieri, Marco Roccetti



Dipartimento di Scienze dell’Informazione
Università di Bologna

Il concetto di Web Server replicati localmente

- un servizio Web utilizza più Servers Web (detti repliche), tutti contenenti le stesse risorse Web, e collocati localmente in un cluster di workstation
- un router/gateway, il cui nome è quello che appare nell'URL del servizio richiesto, distribuisce le richieste provenienti dai browser assegnandole alle diverse repliche.

Questa tecnica

- Suddivide il carico di lavoro tra le repliche, ed aumenta il numero di richieste che il servizio Web può servire nell'unità di tempo, ma
- Non diminuisce il tempo per la trasmissione dei dati via rete dal server al browser.

Il concetto di Web Server replicati in Internet

- Distribuzione geografica delle repliche in Internet per diversificare i percorsi e differenziare i tempi di trasmissione.
- Scelta della replica più “conveniente” a cui passare la richiesta.
- Criteri di scelta di tipo round-robin o basate sul tempo di risposta percepito dall’utente (URT, User Response Time).
- Strategie di scelta basate tipicamente sul DNS per associare il nome dell’host (nell’URL) agli indirizzi IP delle repliche, che condividono lo stesso nome.

Ma anche con questa tecnica

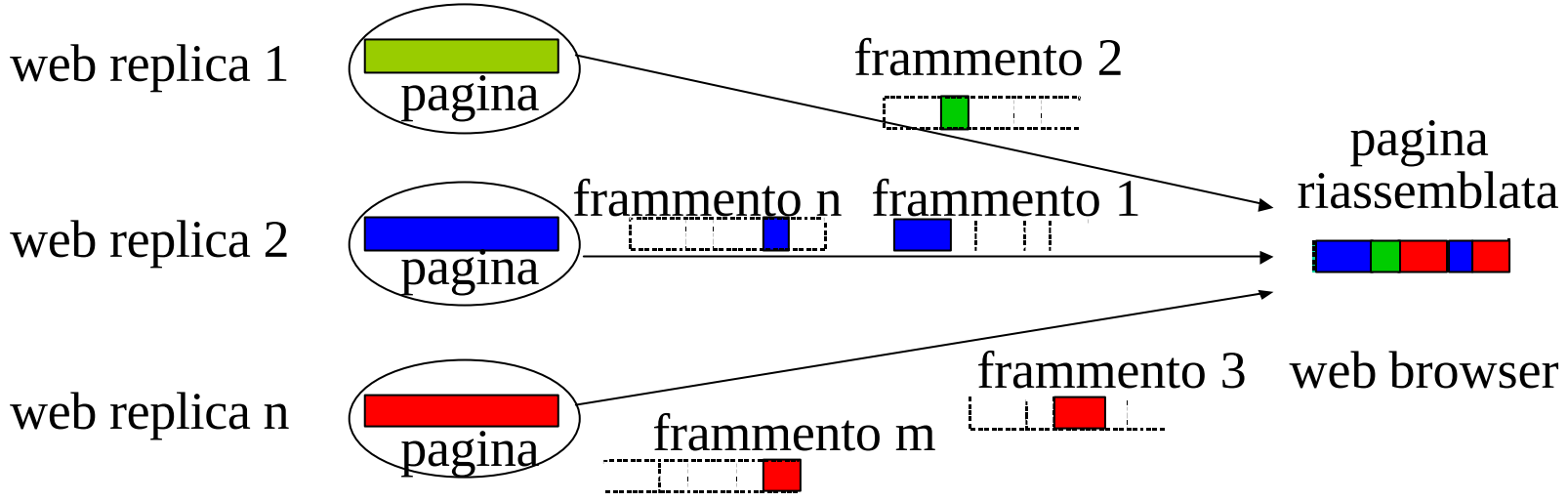
- Ogni richiesta di un browser è servita da una sola replica.
- Dopo tale assegnamento un decadimento delle prestazioni della replica scelta o un partizionamento in rete peggiorano o impediscono la fruizione del servizio nonostante la replicazione.

Il meccanismo C^2LD
Client-Centered Load Distribution

- è un nuovo approccio per la fruizione di servizi Web,
- fornisce disponibilità e prontezza di risposta del servizio,
- è basato sulla replicazione dei server Web in Internet,
- coinvolge tutte le repliche del server nel downloading
- richiedendo ad ogni replica frammenti della pagina richiesta
- utilizzando maggiormente le repliche che forniscono i migliori tempi di risposta all'utente
- viene implementato esclusivamente sul client
- utilizza lo standard HTTP 1.1
- non richiede modifiche sui server Web replicati

Approccio C^2LD: Richieste di Frammenti

- C^2LD (Client-Centered Load Distribution) implementa, sul client,
- l'accesso ad un servizio Web replicato geograficamente.
- La richiesta del browser per una pagina non è assegnata ad una sola replica, ma suddivisa in sottorichieste per frammenti della pagina.
- Ad ogni replica viene richiesto un primo diverso frammento.
- Quando una replica restituisce il frammento richiesto, le viene richiesto un altro frammento, e cosi via fino ad ottenere la pagina.
- I frammenti vengono riassemblati, e la pagina restituita al browser



Approccio C^2LD: Adattività

- C^2LD si adatta dinamicamente alle condizioni della rete e delle repliche, richiedendo frammenti più grandi alle repliche più veloci.
- il controllo delle prestazioni si basa sui tempi di risposta alle sottorichieste, ed avviene periodicamente con periodo S.
- al termine della fase di monitoraggio il meccanismo :
  - seleziona le repliche a cui inviare altre sottorichieste,
  - calcola la dimensione del frammento da chiedere a ciascuna replica,
  - e invia le sottorichieste, una per ogni replica selezionata.
- la dimensione del frammento dipende dalle prestazioni della replica, ed è calcolata in modo che, se la replica mantiene le prestazioni del precedente intervallo, la risposta si completerà circa all'istante del prossimo monitoraggio.
- Le successive sottorichieste usano dati aggiornati sulle prestazioni.

Implementazione di C^2LD

C^2LD opera in 3 fasi:

1) acquisizione URL e richiesta indirizzi IP delle repliche del server al DNS,

2) richiesta HTTP (tipo HEAD) a tutte le repliche contemporaneamente,
   - per ottenere informazioni sulle dimensioni della risorsa richiesta, e
   - per una prima valutazione delle prestazioni delle repliche,

3) sequenza di richieste HTTP di tipo GET alle repliche
   - per ottenere i frammenti dalle diverse repliche,
   - ottenuto un frammento da una replica *i* se ne chiede un altro, di dimensioni proporzionali al Data Rate della replica *i*,
   - fino al completo downloading della pagina, o fino al superamento di un tempo massimo (USD, User Specified Deadline) entro cui l'utente voleva ricevere la pagina.

Protezione dai Fault

Si ha un Fault per il servizio web, quando un partizionamento di rete o un deterioramento delle prestazioni dei server repliche impedisce di effettuare interamente il downloading della risorsa richiesta entro il tempo specificato dall'utente.

Il meccanismo C^2LD protegge il servizio Web dai Fault in questo modo:

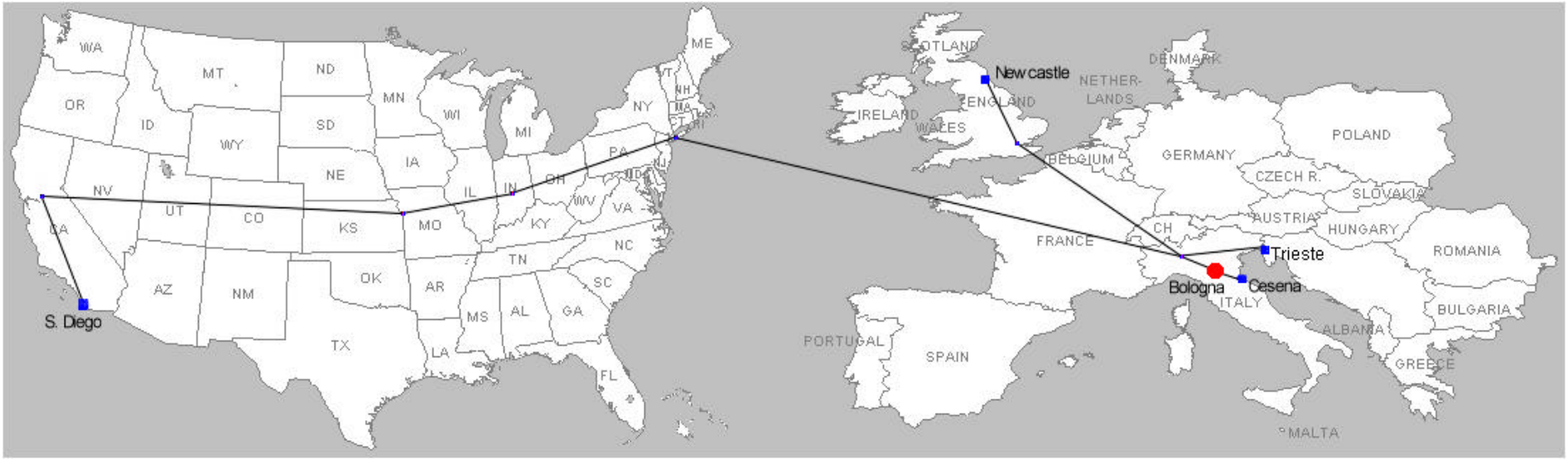
- Quando si chiede un frammento ad una replica *i*, parte un timeout.
- Se il timeout scade prima che la replica abbia consegnato il frammento,
  - si assume che la replica sia *non disponibile*.
  - e il frammento non arrivato viene richiesto ad un'altra replica.

Le repliche non disponibili vengono periodicamente testate per verificare se continuano ad essere non disponibili.

Scenario Sperimentale

Il meccanismo C^2LD è stato testato mediante numerosi esperimenti comparativi, circa 60mila download di files di diversa dimensione.

Gli esperimenti hanno coinvolto 4 repliche server collocate a Cesena, Trieste, Newcastle e San Diego. Il client è stato collocato a Bologna.



In questo scenario, un servizio Web replicato può essere implementato usando una delle 11 possibili combinazioni di due o più tra le 4 repliche esistenti. Abbiamo considerato tutti questi 11 servizi Web, per valutare il meccanismo al variare del numero e della latenza delle repliche.

Modalità degli Esperimenti

Si è voluto confrontare il comportamento del meccanismo C^2LD con il meccanismo HTTP standard di download dei documenti, cioè la GET di HTTP.

Ogni esperimento ha effettuato il download di un certo file, utilizzando prima il meccanismo C^2LD su ciascuno degli 11 possibili servizi Web replicati e poi, per confronto, la modalità GET standard del protocollo HTTP 1.1 su ciascuna delle repliche separatamente.

In particolare, per effettuare le sperimentazioni, ogni replica è stata dotata di files con dimensioni tra 3 Kbytes e 1 Mbytes, ed il periodo di monitoraggio S è stato variato tra 50 millisecondi e 10 secondi.

Metriche

Due metriche sono particolarmente indicate:

i) la percentuale di richieste terminate con successo, cioè quando si è riuscito a rendere disponibile il documento entro il tempo previsto;

ii) il tempo (URT, User Response Time) impiegato per il download.

Con queste metriche è stato analizzato il comportamento del meccanismo al variare di:

1) numero di repliche del servizio Web,
2) Data Rate medio che ogni replica (e la rete) fornisce,
3) periodo di monitoraggio S adottato dal meccanismo C^2LD,
4) la dimensione del documento richiesto.

Risultati: Disponibilità del servizio Web

La tabella riporta la **percentuale delle richieste per cui è stato impossibile effettuare il download in tempo utile**. In seconda colonna la percentuale di fault (0%) del meccanismo C^2LD. Nelle colonne dalla terza alla sesta le percentuali di fault ottenute con il meccanismo HTTP standard di download GET, applicato alle repliche separatamente.

I valori riportati sono la media tra tutti gli esperimenti effettuati.

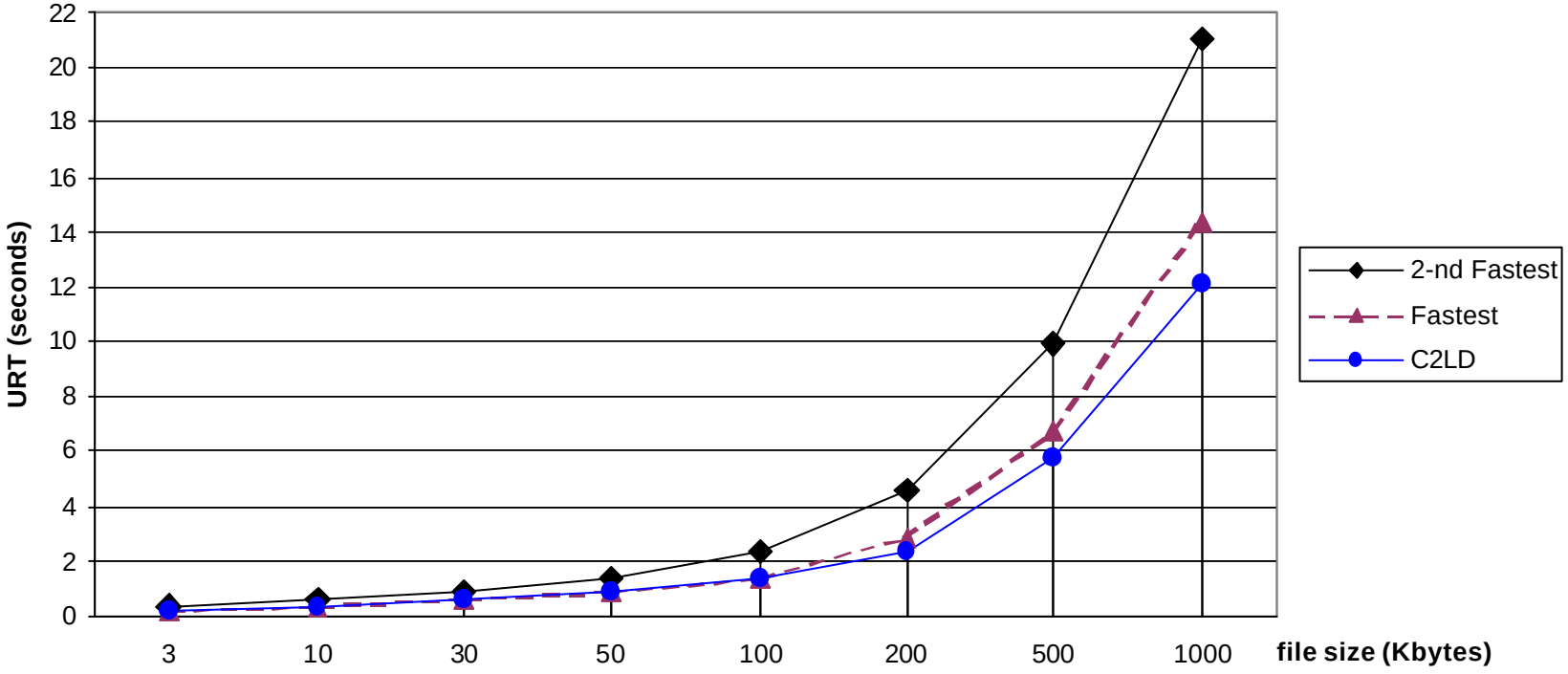
Aggregando i dati, si ottiene una percentuale di errore dello 0% per C^2LD, contro circa 1% per il meccanismo standard.

files Kbytes	C^2LD	S. Diego (GET)	Newcastle (GET)	Trieste (GET)	Cesena (GET)
50	0 %	0.3 %	0 %	0 %	1.65 %
100	0 %	0.25 %	0 %	0 %	5 %
200	0 %	0.2 %	0 %	0 %	4 %
500	0 %	0.3 %	0 %	0 %	1 %
1000	0 %	0.45 %	0 %	0 %	2.5 %

Risultati: URT al variare della dimensione dei files

La figura riporta lo **URT medio tra tutte le prove, al variare della dimensione del file richiesto**. La linea blu rappresenta lo URT ottenuto col meccanismo C^2LD. La linea tratteggiata rappresenta lo URT ottenuto col meccanismo standard dalla replica che "a posteriori" si è rivelata più veloce, cioè il limite teorico per i servizi web replicati.

Si noti che C^2LD è migliore per files maggiori o uguali a 50 Kbytes.



Risultati: URT al variare del numero di repliche

La tabella riporta **il miglioramento percentuale dello URT con C^2LD rispetto alla più veloce delle repliche**, considerando la media tra tutti gli esperimenti al crescere del numero delle repliche.

Il miglioramento introdotto da C^2LD aumenta al crescere del numero delle repliche.

Numero di Repliche	miglioramento percentuale medio dell' URT di C^2LD rispetto alla replica più veloce
2	4%
3	17.2%
4	21.5%

Conclusioni

- C^2LD è un nuovo approccio per la progettazione di servizi Web, affidabili e veloci, basati su repliche del server distribuite geograficamente in Internet.
- è implementato sul client e non richiede modifiche sui server replicati.
- fa si che tutte le repliche concorrano al downloading di un file per un client, ciascuna secondo le sue possibilità.
- I risultati sperimentali indicano che C^2LD è estremamente efficace, soprattutto quando applicato a files di grandi dimensioni.
- In particolare C^2LD migliora le prestazioni del meccanismo standard di downloading, perchè:
  - migliora notevolmente la resistenza ai fault,
  - diminuisce lo URT dal 4% al 21% al variare del numero di repliche

Sviluppi Futuri

Gli sviluppi futuri del lavoro riguarderanno quattro diverse direzioni:

1) valutazione del'overhead introdotto dal meccanismo C^2LD sia in rete che nei server,

2) miglioramento del meccanismo con anticipo della richiesta rispetto alla terminazione della risposta precedente,

3) collocazione del meccanismo in un proxy server,

4) confronto con altri meccanismi di replicazione, come quelli forniti da cluster locali di workstation

Ulteriori dettagli

seguono 7 lucidi non utilizzati,

da usare per eventuali richieste di precisazioni

- Fase 1,
- Fase 2,
- Fase 3,
- Risultati: miglioramento percentuale dell’URT al variare della dimensione dei files
- Risultati: URT al variare del periodo di monitoraggio S
- Risultati: URT al variare del tipo delle repliche
- Condizioni della rete durante le sperimentazioni

Fase 1:
acquisizione URL e richiesta indirizzi IP delle repliche

- il browser invoca la GET HTTP con l'URL della risorsa richiesta,
- C^2LD intercetta la chiamata, e interroga il DNS con l'hostname dell'URL della risorsa,
- il DNS restituisce gli indirizzi IP di tutte le repliche

Fase 2: Richiesta HTTP di tipo HEAD

- C^2LD invia, contemporaneamente, ad ogni replica la richiesta HTTP tipo HEAD (richiesta informazioni su un URL) con l'URL richiesto.
- C^2LD riceve la prima risposta alla HEAD da una certa replica *i*, la più veloce a rispondere, ed effettua le seguenti operazioni:
  - memorizza la dimensione della pagina richiesta,
  - calcola il Data Rate della replica *i* per la risposta ricevuta,
  - calcola la dimensione del primo frammento da chiedere alla replica *i*,
  - invia alla replica i la richiesta HTTP di tipo GET per l'URL richiesto, aggiungendo l'opzione RANGE per ottenere il solo frammento.
- Le risposte alle HEAD provenienti dalle altre repliche verranno trattate in modo analogo alla prima ricevuta.

Fase 3: Richiesta HTTP di tipo GET

nel momento in cui C^2LD riceve un frammento da una replica *i*,

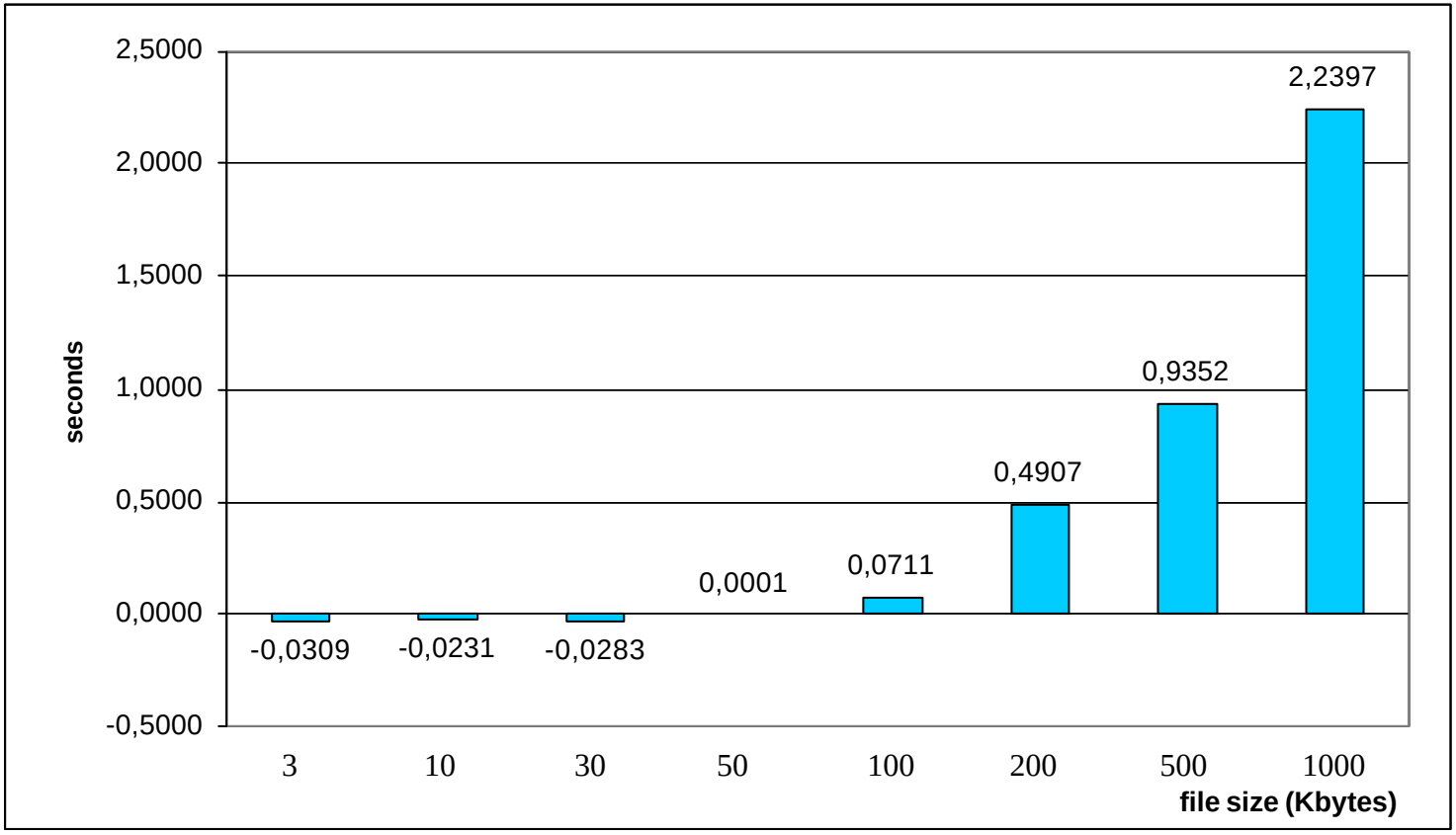
- C^2LD calcola il Data Rate DR della replica i per quel frammento, ovvero il rapporto tra la dimensione del frammento e il tempo impiegato per ottenerlo,
- calcola il tempo residuo S* fino al prossimo istante di monitoraggio,
- calcola la dimensione del prossimo frammento da chiedere alla replica *i*, moltiplicando il Data Rate per il tempo residuo ($DR \cdot S^*$),
- invia alla replica *i* una nuova richiesta HTTP di tipo GET, con l'opzione RANGE per ottenere il nuovo frammento.

Si prosegue chiedendo frammenti fino a che:

- tutta la pagina è stata ottenuta, o
- si è superato un tempo massimo (USD, User Specified Deadline) entro cui l'utente voleva ricevere la pagina.

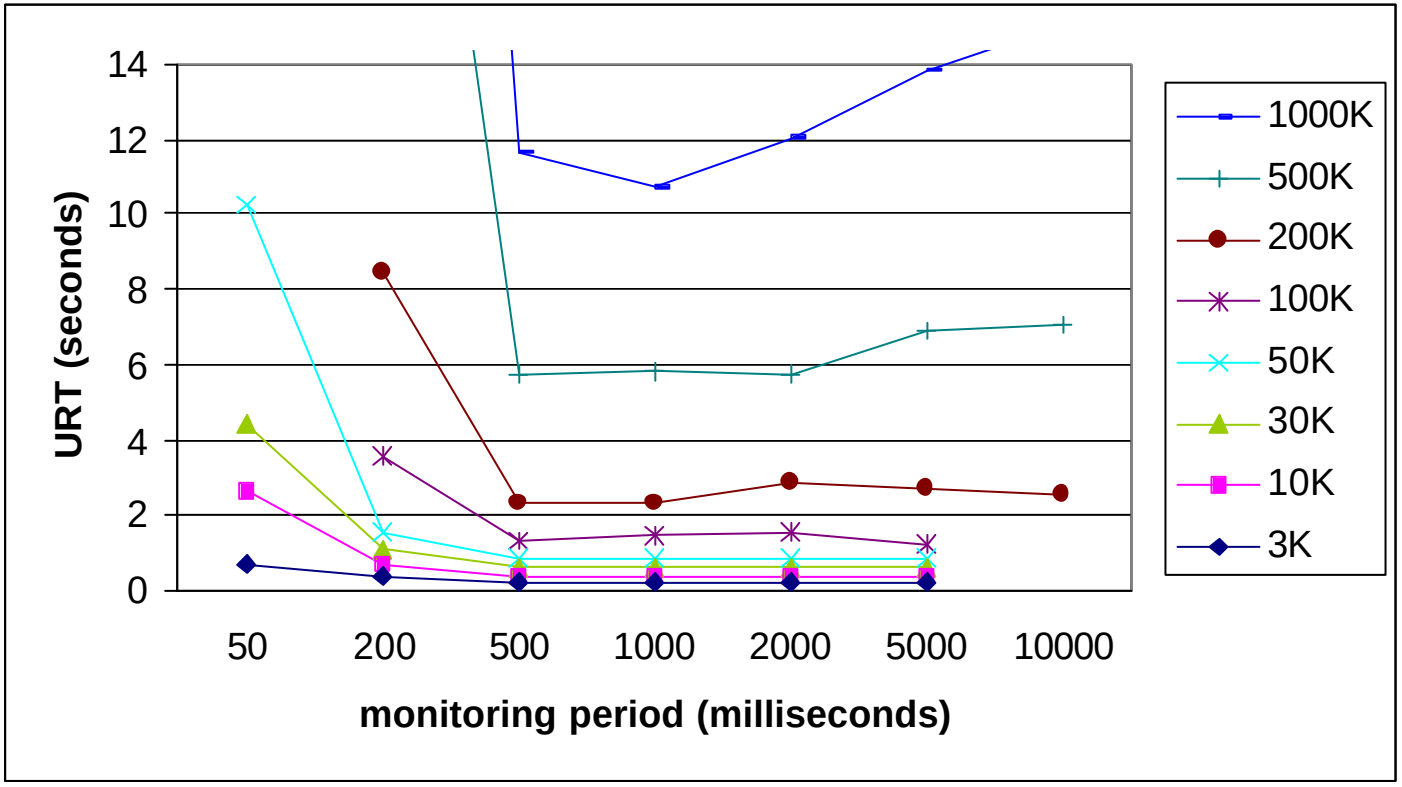
Risultati: URT al variare della dimensione dei files

La figura riporta **il miglioramento dello URT con C²LD rispetto alla più veloce delle repliche**, cioè rispetto alla massima performances "teorica" di un servizio Web replicato, qualora fosse possibile conoscere a priori la replica più veloce. C²LD è migliore per files maggiori o uguali a 50 Kbytes.



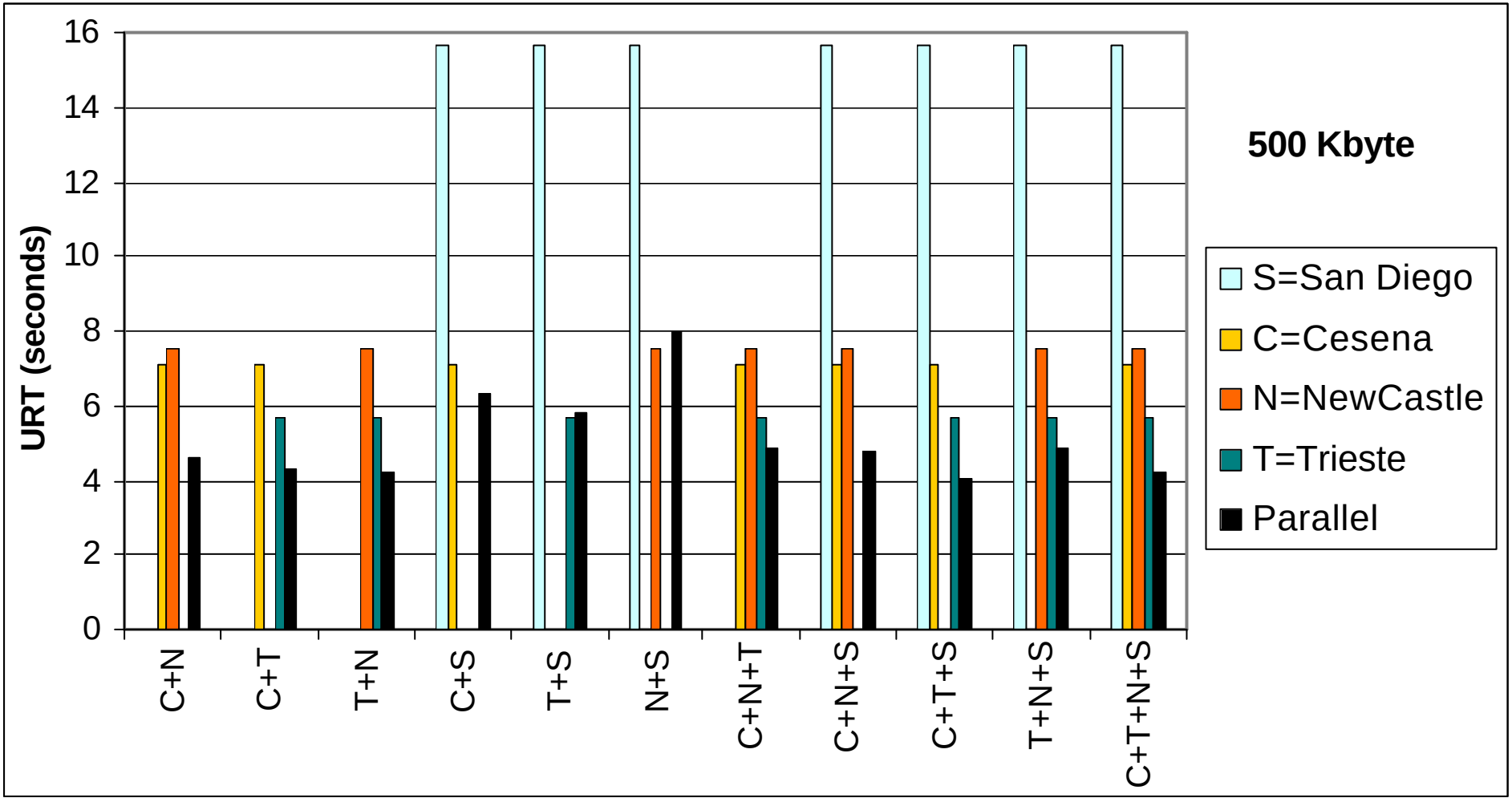
Risultati: URT al variare del periodo di monitoraggio S

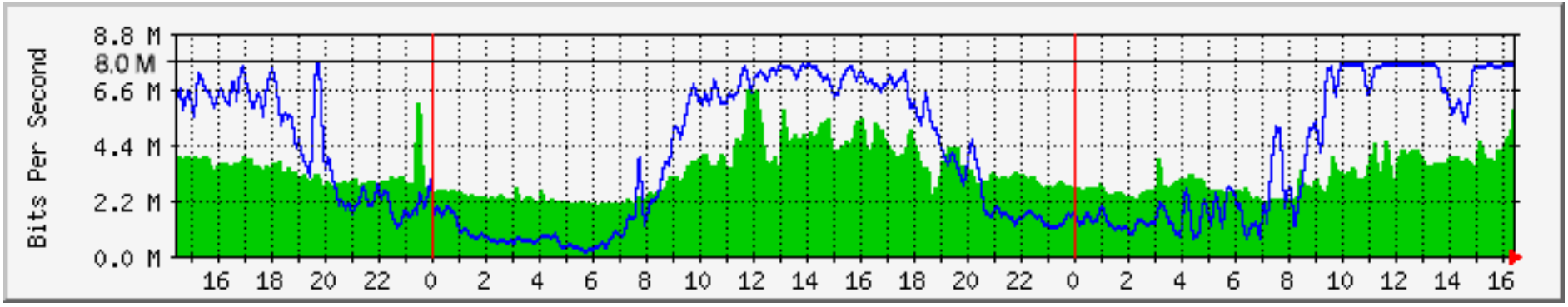
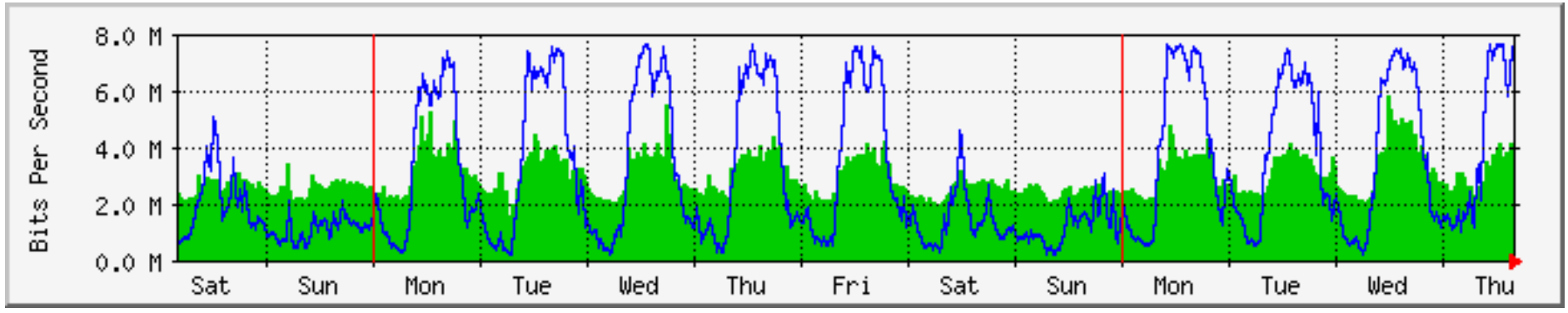
La figura evidenzia lo URT medio ottenuto col meccanismo C^2LD, per diverse dimensioni di files, al variare dell'intervallo di monitoraggio S considerato. Nelle condizioni illustrate, il periodo di monitoraggio che fornisce i risultati migliori in termini di URT appartiene all'intervallo 500-1000 msec.



Risultati: URT al variare del tipo di repliche

La figura compara lo URT di C^2LD e delle repliche che compongono il servizio web replicato (solo con file da 500 Kbytes).



Condizioni della rete

ping da Bologna verso:	Cesena	Trieste	Newcastle	S. Diego
RTT 90% dei pkt arrivati (msec)	107	95	160	450
RTT min (msec)	10	38	59	190
Lost Packet	2%-9%	0%	0%	0%-3%