



# LibreSpeed

Federico Dossena e contributors

<https://github.com/librespeed>

**10 ottobre 2019 – Workshop GARR 2019**

# Perchè?

- Manca uno Speedtest Libero e Open Source di qualità che funzioni nel browser
- Agli utenti serve un modo per testare la velocità della loro connessione ad Internet
- Alti costi di licenza delle soluzioni proprietarie come OOKLA
- Permettere agli ISP di rimanere in controllo dei loro dati invece di darli ad aziende esterne, e potenzialmente anche di fornire un servizio migliore
- Permettere a chi ha un sito web di aggiungere facilmente uno speedtest

# Requisiti

## Server:

- Apache2, IIS, nginx (testato dagli utenti anche su altre piattaforme)
- PHP 5.4 o superiore con OpenSSL (altri backend disponibili)
- MySQL, PostgreSQL, SQLite (opzionale)
- Molta banda!

## Client:

- Browser con supporto a XHR Level 2 e Web Workers (IE11, Firefox, Chrome, Edge, ...) Nessun plugin è richiesto.
- Fino a 5-600MB di RAM durante il test, a seconda della velocità della connessione

# Test di download/upload

- Trasferiscono un grosso blob di dati spazzatura dal server al client e viceversa
- Basato su XmlHttpRequest Level 2, niente WebSockets
- Stream paralleli
- Durata del test automatica in base alla velocità rilevata (oppure fissa)
- Velocità visibile in tempo reale durante il test
- Compensa l'aggiustamento iniziale della window TCP e i buffer di upload
- Misura quanto velocemente possono essere scaricati o caricati dei dati su siti che trasferiscono grandi quantità di dati come YouTube

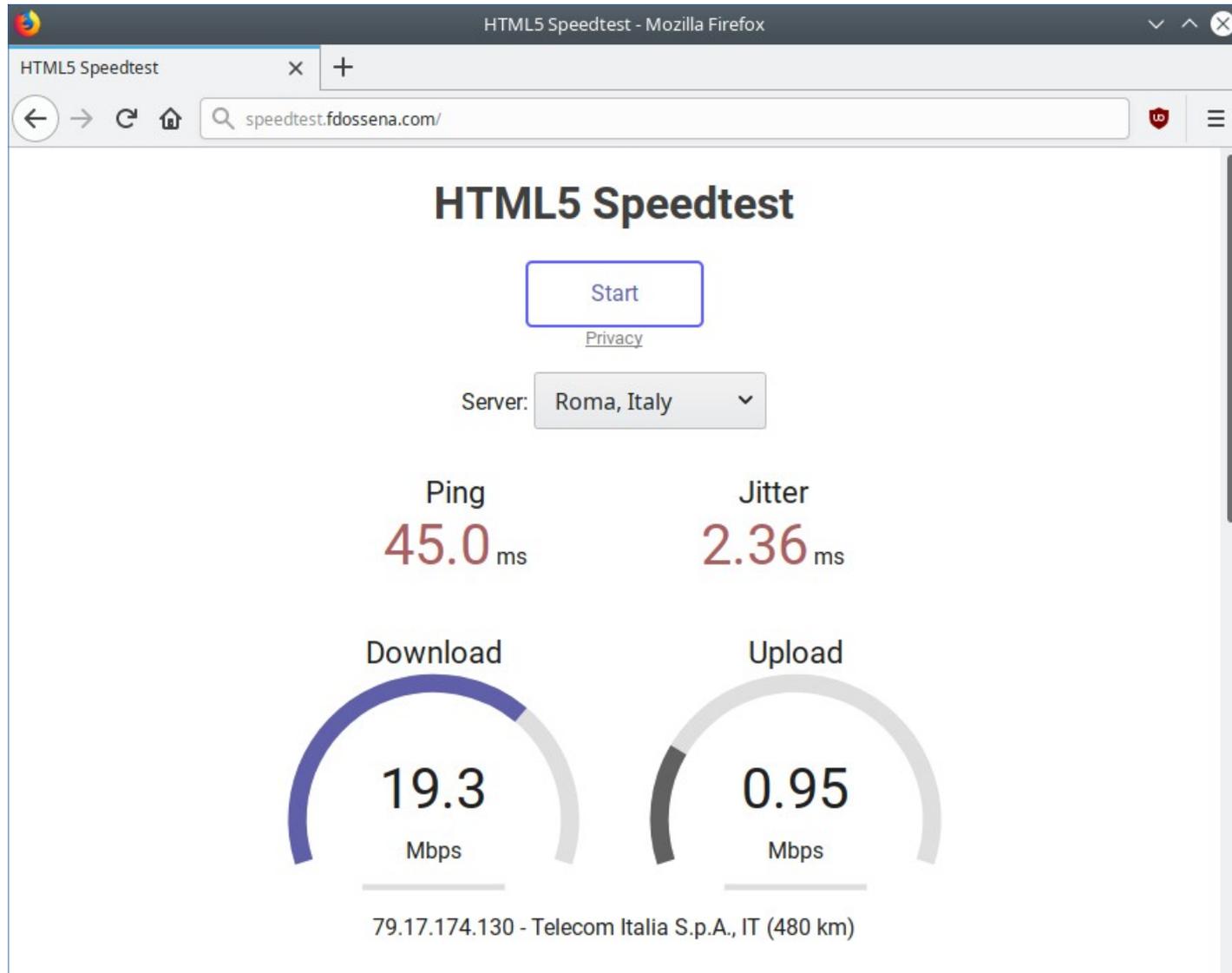
# Test di ping e jitter

- Il ping è misurato osservando il tempo richiesto per trasferire un file vuoto non cachabile su una connessione HTTP persistente, ~1 RTT
- Utilizza le Performance API per migliore precisione (se disponibile)
- Non è un ping ICMP!
- Il jitter è la varianza tra ping consecutivi
- Risultati sono leggermente manipolati per rimuovere gli outliers
- Ping e jitter visibili in tempo reale durante il test

# IP, ISP e distanza dal server

- L'indirizzo IP, il nome dell'ISP, e la distanza dal server possono essere mostrati durante il test
- Il nome dell'ISP e la posizione approssimativa sono ottenuti utilizzando le API pubbliche di [ipinfo.io](https://ipinfo.io), i quali hanno fornito accesso illimitato al servizio per gli utenti di questo progetto
- La distanza approssimativa è calcolata dalla distanza in linea d'aria tra le posizioni approssimative di client e server
- Distanza misurata in km o miglia, con precisione di 15km
- Il server deve potersi connettere a [ipinfo.io](https://ipinfo.io), ovviamente

# Screenshot: test nel browser



# Telemetria

- I risultati dei test possono essere salvati sul server. Opzionalmente, si possono anche salvare il log e i test abortiti o falliti
- 3 backends di database sono supportati con minima configurazione:
  - MySQL (o MariaDB)
  - PostgreSQL
  - SQLite
- Viene fornita una pagina protetta da password dove visualizzare e cercare i risultati dei test
- Opzionalmente è possibile censurare gli indirizzi IP nella telemetria per migliore privacy

# Screenshot: risultati

HTML5 Speedtest - Stats

Logout

Search test results

Test ID  Find Show last 100 tests

<b>Test ID</b>	15ibgms (deobfuscated: 203007)
<b>Date and time</b>	2019-09-21 16:58:08
<b>IP and ISP Info</b>	79.21.58.65 { "processedString": "79.21.58.65 - Telecom Italia S.p.A., IT (480 km)", "rawIspInfo": { "ip": "79.21.58.65", "hostname": "host65-58-dynamic.21-79-r.retail.telecomitalia.it", "city": "Bergamo", "region": "Lombardy", "country": "IT", "loc": "45.6960,9.6672", "org": "AS3269 Telecom Italia S.p.A.", "postal": "24100", "timezone": "Europe\\Rome" } }
<b>User agent and locale</b>	Speedtest-Android/1.1 (SDK 28; lineage_oneplus3; Android 9) en-US
<b>Download speed</b>	17.93
<b>Upload speed</b>	0.93
<b>Ping</b>	40.18
<b>Jitter</b>	4.22
<b>Log</b>	1569077808486 Test started 1569077809005 GetIP: 79.21.58.65 - Telecom Italia S.p.A., IT (480 km) (took 519ms) 1569077810379 Ping: 40.178962 4.218345647681536 (took 1372ms) 1569077823073 Download: 17.925984376634702 (took 11694ms) 1569077841385 Upload: 0.925224704105451 (took 17311ms)
<b>Extra info</b>	{ "server": "Roma, Italy" }

# Condivisione dei risultati

- Genera un link ad un'immagine che può essere condiviso, inserito nelle firme dei forum, ecc.
- Nessuna informazione sensibile è esposta da questa funzionalità
- Opzionalmente è possibile offuscare gli ID dei test per nascondere gli ID interni al database
- Richiede FreeType2

# Screenshot: results sharing

## Share results

Test ID: 154gt10

<http://speedtest.fdossena.com/results/?id=154gt10>

**Ping**

45.0 ms

**Jitter**

2.36 ms

**Download**

19.3

Mbps

**Upload**

0.95

Mbps

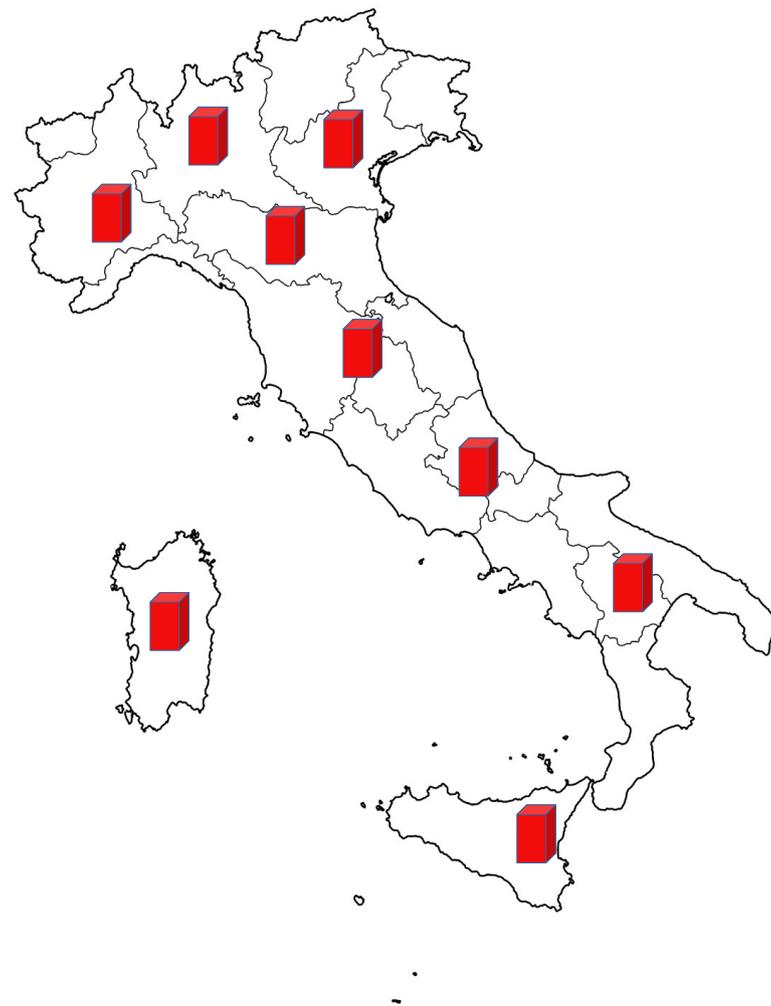
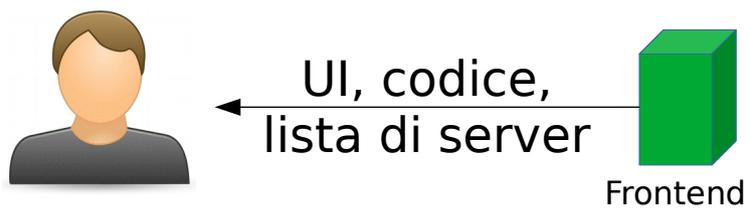
Telecom Italia S.p.A., IT

HTML5 Speedtest

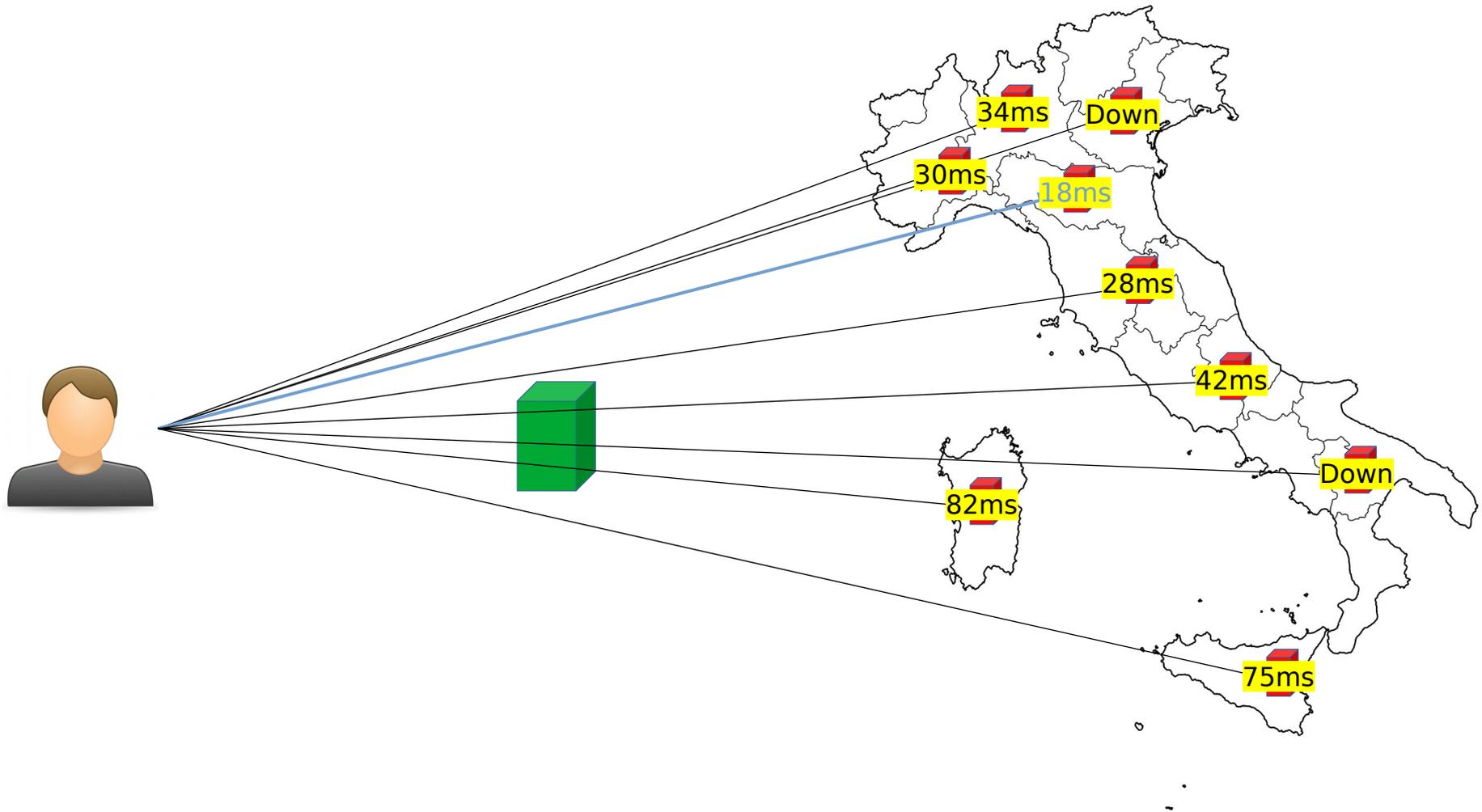
# Punti di test multipli

- Molto utile per gli ISP che hanno server su tutto il paese
- Lo speedtest sceglie automaticamente da una lista il server che ha il ping migliore sotto una certa soglia. I server offline vengono ignorati
- Selezione automatica o manuale

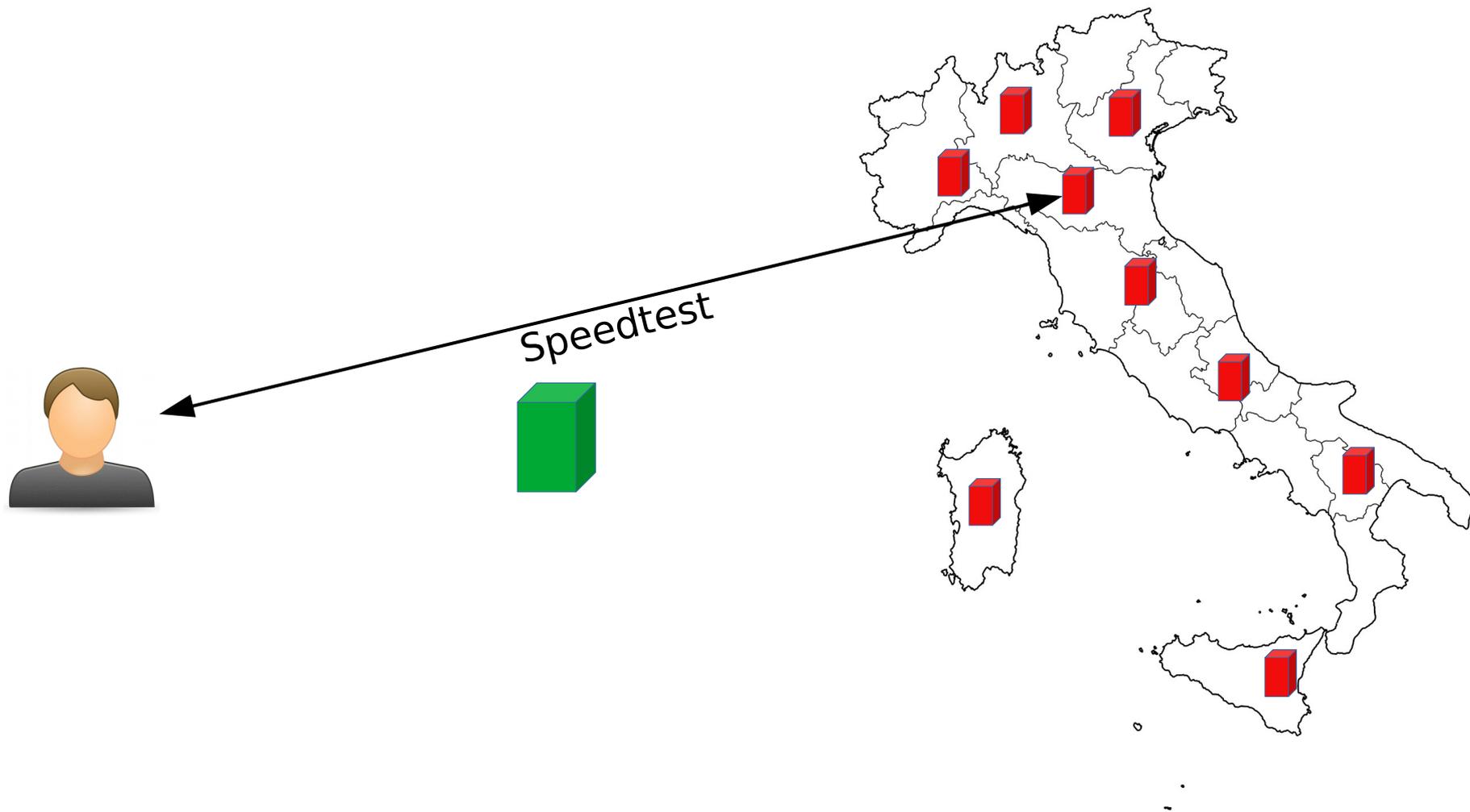
# Punti di test multipli



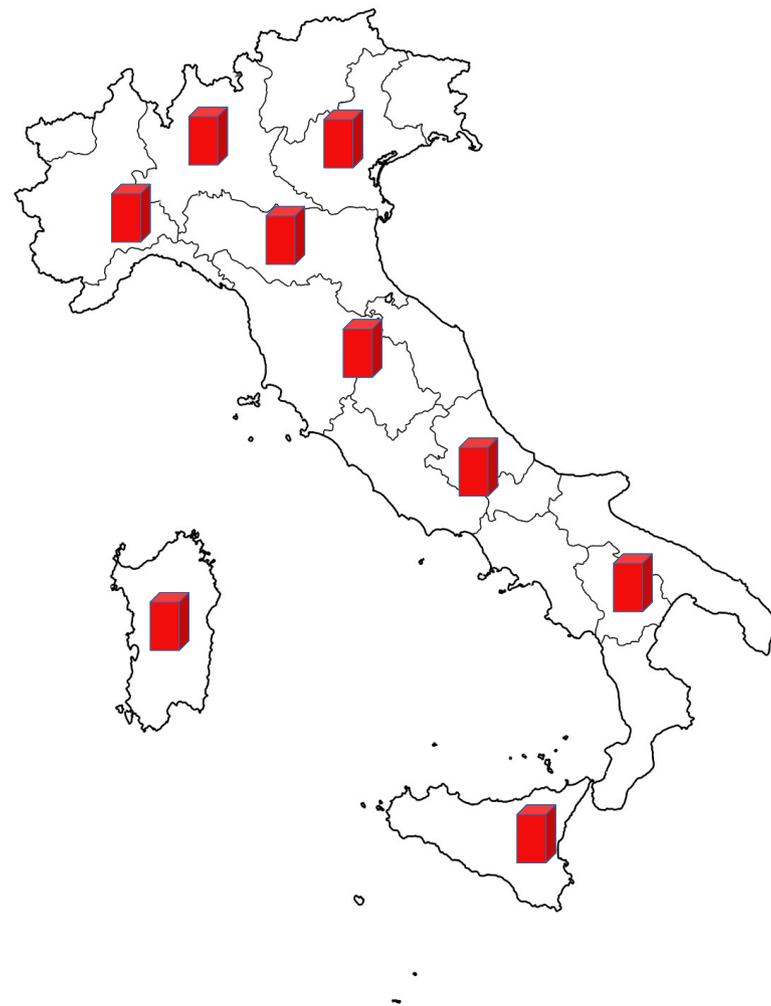
# Punti di test multipli



# Punti di test multipli



# Punti di test multipli

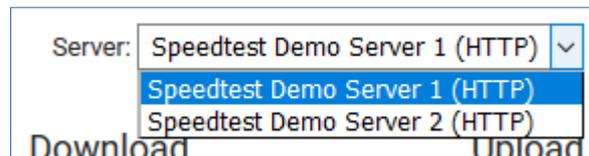
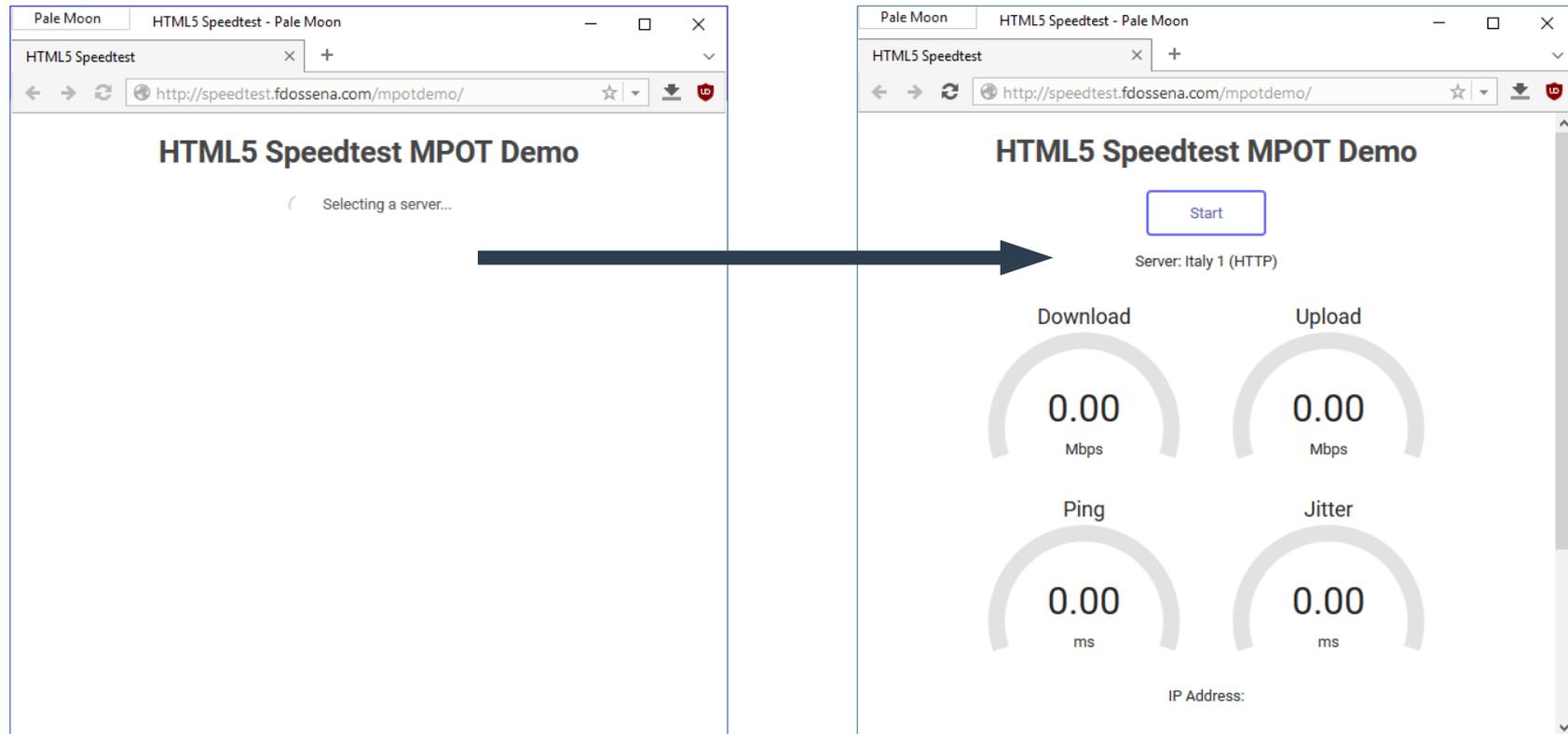


# Punti di test multipli

## Attenzione:

- Al server di frontend non serve molta banda, ma ai backend si
- Involve header CORS!

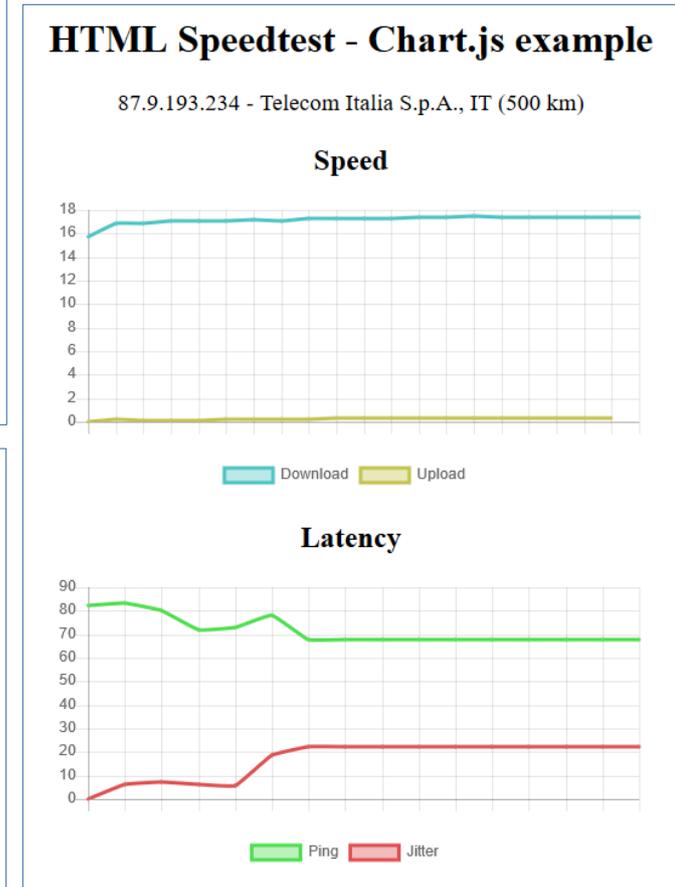
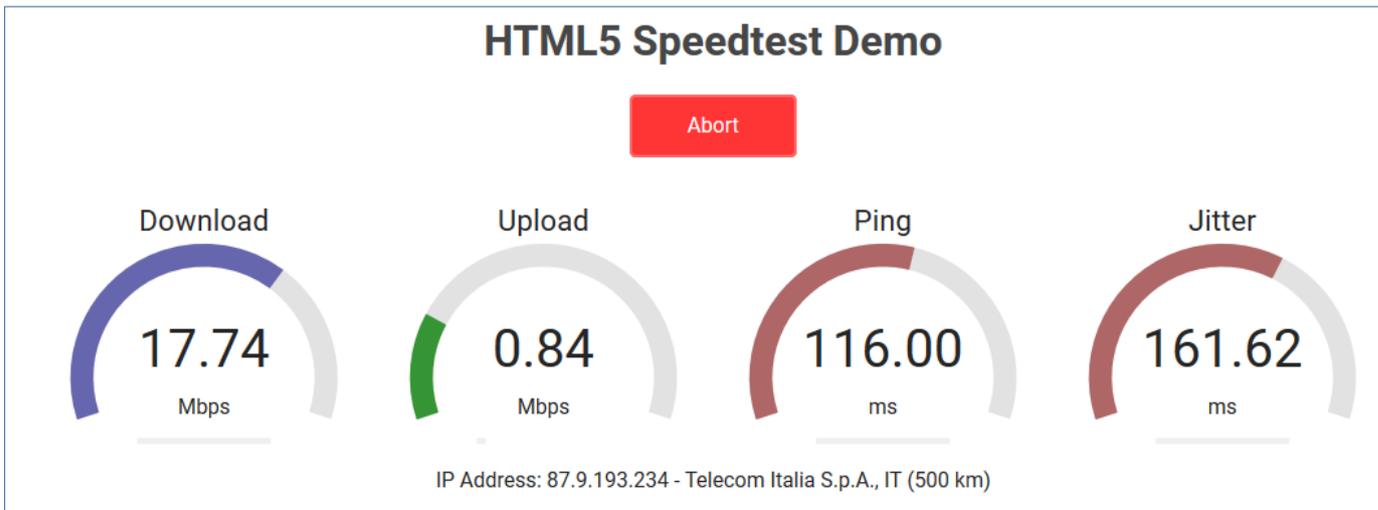
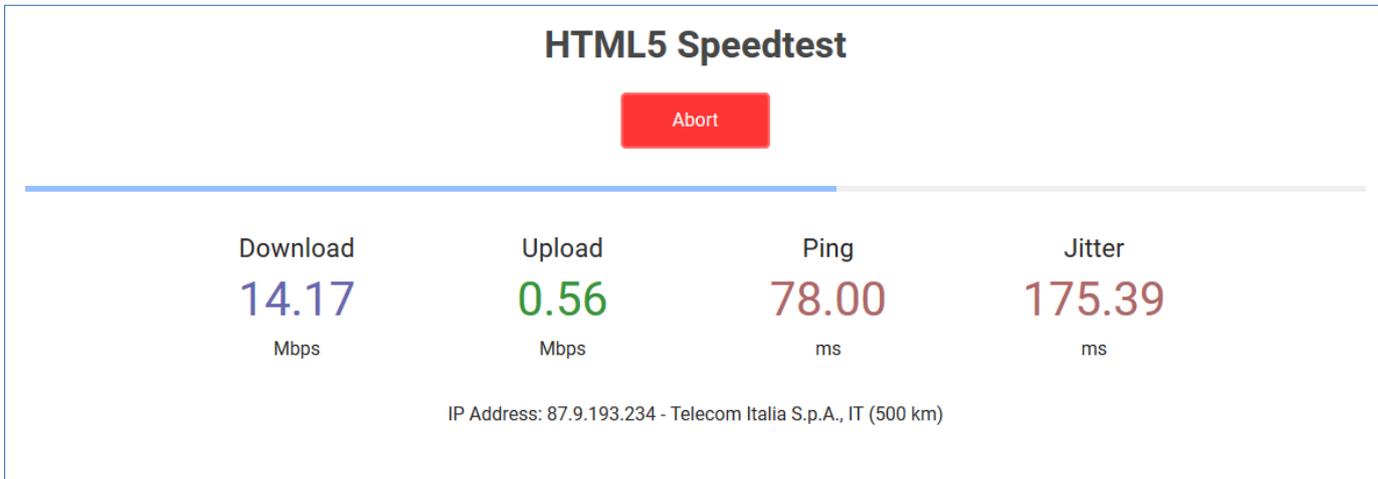
# Screenshot: selezione del server



# Personalizzazione

- Utilizzare lo Speedtest in una pagina web è molto semplice: viene fornito un file .js da importare, che fornisce una classe Speedtest e dei callback chiamati durante il test da usare per aggiornare la UI
- Molte impostazioni possono essere modificate, per adattare il test a scenari specifici, come ad esempio il lag di una connessione satellitare
- Sono inclusi diversi esempi, che mostrano tutte le funzionalità e come creare diversi tipi di UI
- Una documentazione estensiva spiega tutto da una semplice installazione fino a telemetria e frontend personalizzate

# Screenshot: UI personalizzate



# Docker

- Una versione Docker dello Speedtest è disponibile sul Docker Hub
- Fornisce un semplice container con una installazione di base con un singolo punto di test
- Presto saranno fornite versioni per punti di test multipli (backend e frontend), facilmente configurabili

# Altri backend

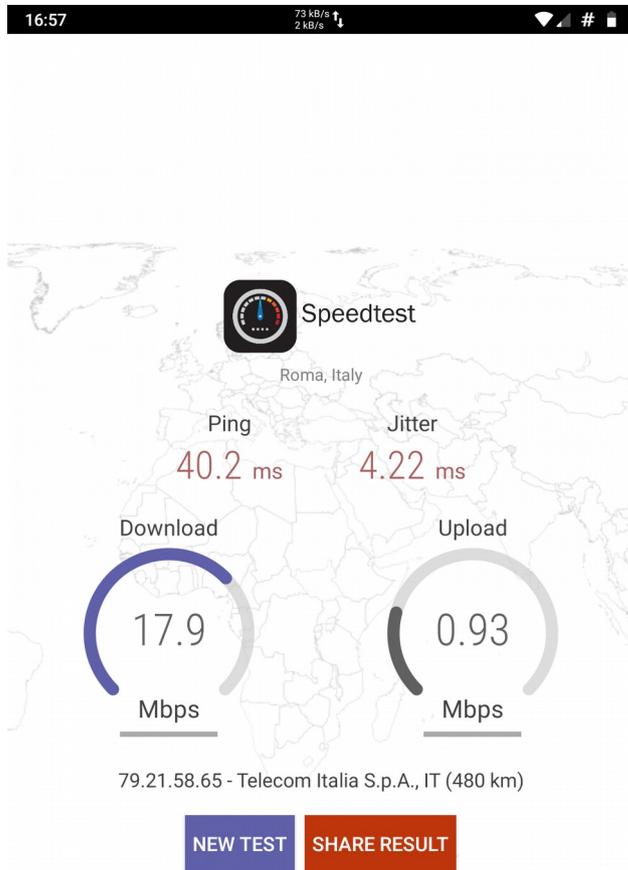
- Node.js
  - Ufficiale, mantenuto da un contributor
  - Non ancora feature complete, ma in sviluppo
- Solo file server
  - Un grosso file di dati non comprimibili può essere usato per eseguire il test di download
  - Un file vuoto può essere usato per i test di upload, ping e jitter
  - Nessun'altra funzionalità può essere usata
- ASP.net (non ufficiale)
- JSP (non ufficiale)
- Go (non ufficiale)
- Probabilmente altri

# Client Android



- Template per creare applicazioni Android che utilizzano i propri server di LibreSpeed
- Android 4.0.3+, non dipende da servizi Google
- Personalizzabile, brandizzabile
- Migliori prestazioni su mobile rispetto alla versione web

# Client Android



- Tutte le funzionalità della versione web sono implementate
- Supporto a punti di test multipli
- Interfaccia e impostazioni del test personalizzabili per adattarlo a diversi scenari
- Creare UI personalizzate più complesso rispetto alla versione web, ma comunque possibile e consentito

# Client Android: personalizzazione

- Branding (loghi, testi, colori, ...)
- Elenco di punti di test
- Impostazioni dello speedtest, anche a basso livello (socket)
- Impostazioni della telemetria
- Privacy policy

# Stato del progetto

- 3.5 anni, piuttosto maturo, pochissimi bug report
- 2.7k stelle su GitHub, di gran lunga il più popolare
- Oltre 700 fork, alcuni dei quali attivi
- 21 contributor, 2 sviluppatori attivi al momento
- Media di 50 git clone/giorno e 600 visitatori
- Il server di demo serve ~1000 test/giorno
- In uso da molti ISP nel mondo, alcuni famosi

# Limitazioni attuali

- Non è accurato come iperf, non è pensato per rimpiazzarlo
  - Nel browser, la velocità massima è limitata dalla velocità della CPU (quanto velocemente riesce a processare gli eventi XHR)
  - Sul server, la velocità massima è limitata dalla velocità con cui PHP riesce a generare dati spazzatura
- Nessun team di supporto

# Licensing

- Utilizzare LibreSpeed non costa niente (ma apprezziamo contributi in codice e donazioni)
- Licenza GNU LGPLv3
  - Libero di utilizzare, studiare, modificare, ridistribuire
  - Utilizzabile come libreria anche in software proprietario, ma modifiche al codice di LibreSpeed devono essere rese pubbliche sotto la stessa licenza (eccetto modifiche alla sola configurazione)

# Il futuro

## In sviluppo:

- Client nativo per iOS
  - Simile al client Android, si vuole fornire un template che gli amministratori di server LibreSpeed possono configurare e distribuire
- Server Node.js feature complete
- Nuovo container docker

## Possibili contributi:

- Client nativi per altre piattaforme
  - Command line, desktop, ...
- Ulteriori server backend in giro per il mondo per la demo

# Collaborazione con GARR



- GARR ha scelto di utilizzare LibreSpeed sia in versione web che su Android, con le proprie personalizzazioni
- Un sistema basato su Ansible mantiene automaticamente aggiornati i server e la versione Android con l'ultimo master
- Client iOS attualmente in sviluppo
- GARR ha gentilmente offerto l'utilizzo dei punti di test di Bologna, Roma e Bari per la demo di LibreSpeed, con rete multi-gigabit
- <https://speedtest.garr.it>

<https://play.google.com/store/apps/details?id=it.garr.speedtest.android> (in test pubblico)

# Collaborazione con GARR



Home | GARR | Cookies

GARR speedtest

## GARR SPEEDTEST - VERIFICA CONNETTIVITÀ

Start  
Privacy

Server selezionato: Roma

Ping  
**14.8** ms

Jitter  
**0.47** ms

Download  
**405** Mbps

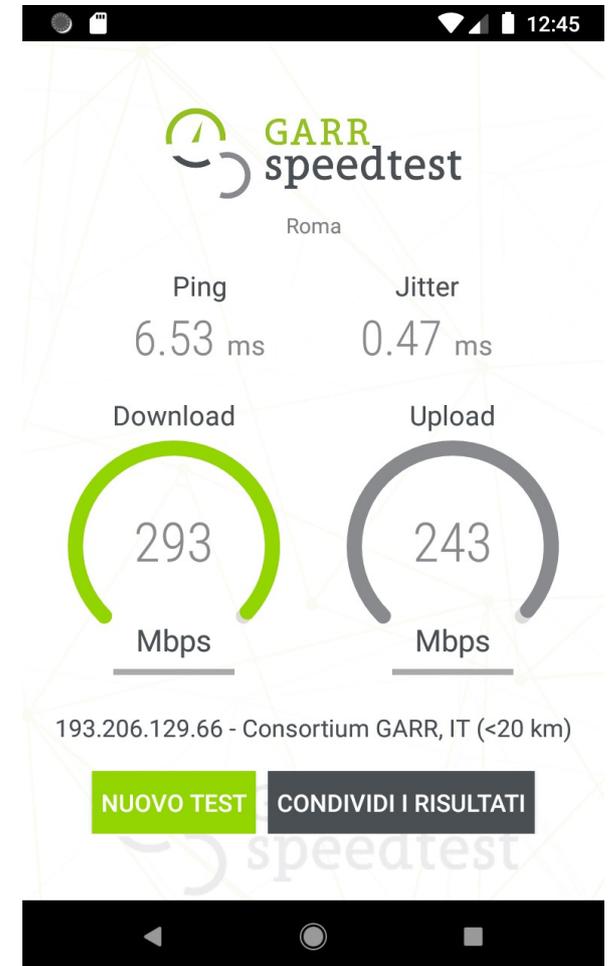
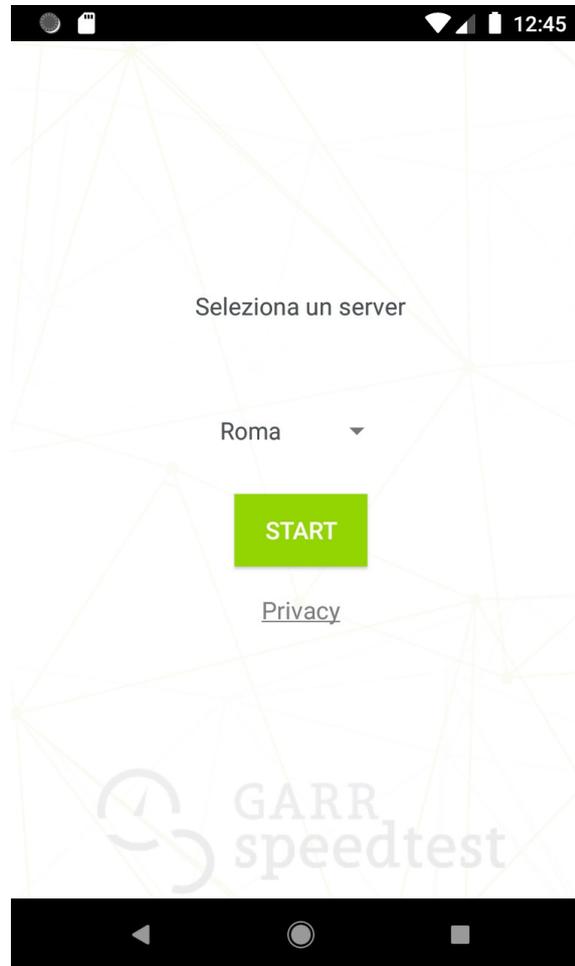
Upload  
**358** Mbps

Indirizzo IP: 193.206.129.66 - Consortium GARR, IT (<20 km)

Condividi i risultati  
Test ID: 1cwf3k1

<a href="https://speedtest.garr.it/results/?id=1cwf3k1">https://speedtest.garr.it/results/?id=1cwf3k1</a>			
Download	Upload	Ping	Jitter
404.83	358.31	14.82	0.47
Mbps	Mbps	ms	ms
Consortium GARR, IT		Server: Roma	GARR SpeedTest

# Collaborazione con GARR



# Collaborazione con GARR



A screenshot of a CI/CD pipeline interface. On the left is a sidebar for a project named "speedtest-android" with navigation options like Project, Repository, Issues, and CI/CD. The main area shows a terminal log for a "build:release" job, listing various Gradle tasks such as "generateReleaseResValues", "compileReleaseJavaWithJavac", and "packageRelease". The log concludes with "BUILD SUCCESSFUL in 38s" and "Job succeeded". On the right, a summary panel for the "build:release" job shows a duration of 1 minute 43 seconds, a timeout of 1h, and a runner of "gitlab-runner-1-docker (#43)". It also lists job artifacts and commit information.

Per ulteriori informazioni:  
<https://github.com/librespeed>

**Domande?**