

# SOMMARIO

- Introduzione a cpprestsdk
- Programmazione asincrona
- Modello dei task di pplx
- Esercizi con i task
- Funzionalità di cpprestsdk
- Esempi con cpprestsdk
- Domande

# YURI VALENTINI

- SW Windows e Linux
- FW schede embedded
- Linguaggi: C/C++, C#, Python
- Ambiti:
  - Videoconferenza e VOIP
  - Automazione industriale
- [yuroller@gmail.com](mailto:yuroller@gmail.com)
- <https://github.com/yuroller>

# CPPRESTSDK

*The C++ REST SDK is a Microsoft project for cloud-based client-server communication in native code using a modern asynchronous C++ API design. This project aims to help C++ developers connect to and interact with services.*

# SERVIZI WEB / API CLOUD

- Google Maps
- Facebook Graph API
- Twitter
- Dropbox
- Wordpress
- ...

# SERVIZI WEB / API CLOUD

- Google Maps
- Facebook Graph API
- Twitter
- Dropbox
- Wordpress
- ...

Raramente libreria o sdk per c++ 😞

# PROTOCOLLI

- http/https
- websocket
- json
- oauth/oauth2

# PROTOCOLLI

- http/https
- websocket
- json
- oauth/oauth2

Ci aiuta cpprestsdk 😁

# INSTALLAZIONE

```
> git clone https://github.com/Microsoft/vcpkg.git  
> cd vcpkg  
> bootstrap-vcpkg.bat  
(admin)> vcpkg integrate install  
> vcpkg install cpprestsdk:x64-windows
```



# CARATTERISTICHE

- asincrona
- multiplatforma:
  - Windows desktop/UWP
  - Linux / Android
  - Mac OS X / iOS
  - Legacy (XP, phone8/8.1)
- sufficientemente completa
- abbastanza di alto livello

# "COLAZIONE" ASINCRONA

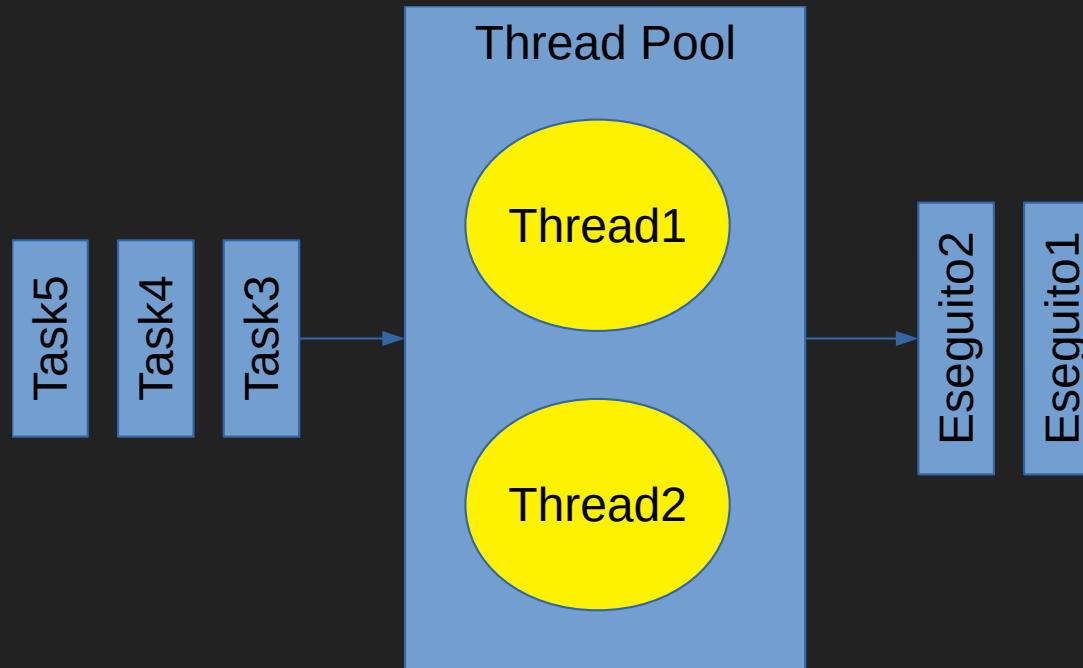
- Metto una tazza di acqua nel microonde
- Intanto apparecchio il tavolo
- Il microonde suona: metto la bustina di tè
- Intanto preparo fette biscottate e marmellata
- Il tè è pronto
- Faccio colazione

Op. C.Carruth@cppcon2014	ns	ms
Ciclo su processore 3GHz	1	
Mutex lock/unlock	25	
Tx 1KB su rete 1Gbps	10'000	0.01
Lettura 4KB da disco SSD	150'000	0.15
Lettura 1MB da disco SSD	1'000'000	1
Seek HD	10'000'000	10
Lettura 1MB seq da HD	20'000'000	20
Round-trip CA-Olanda	150'000'000	150ms

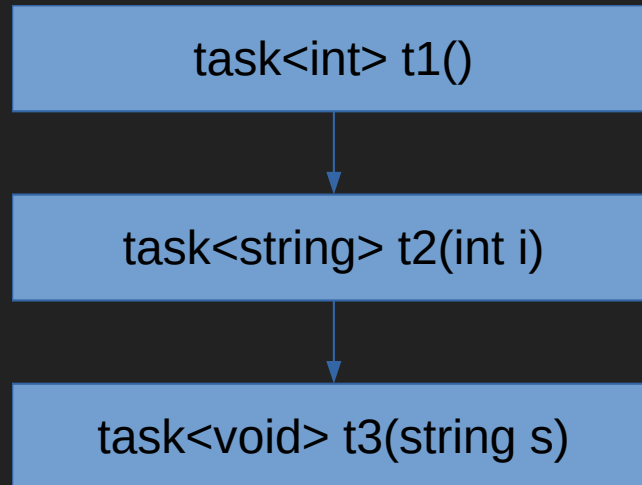
# PPLX::TASK

- `std::async` + `std::future` + continuation
- TPL di C#
- scheduler specifici:
  - Windows ThreadPool
  - UWP WorkItemHandler
  - Mac OsX GrandCentralDispatch
  - o generico `boost::asio`

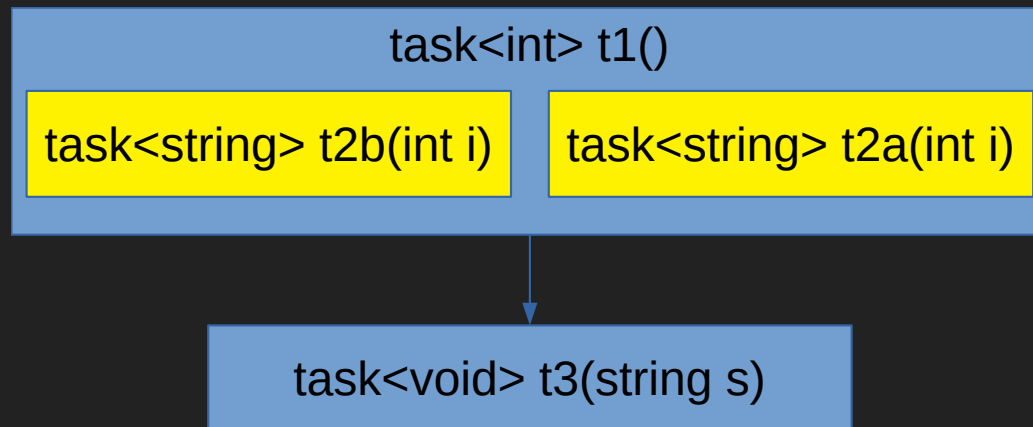
# ESECUZIONE TASK



# TASK SEQUENZIALI



# TASK INNESTATI



# ESEMPIO PPLX::TASK

```
pplx::task<int> risposta  
    = CalcolaRispostaAllaVitaUniversoETuttoQuanto();  
  
risposta.then([](int valore) {  
    std::cout << "La risposta è " << valore << '\n';  
});
```



# GESTIONE ERRORI

```
pplx::task<int> risposta
    = CalcolaRispostaAllaVitaUniversoETuttoQuanto();

risposta.then([](pplx::task<int> antecedent) {
    try {
        int valore = antecedent.get();
        std::cout << "La risposta è " << valore << '\n';
    }
    catch (...) {
        // ...
    }
}));
```

# GESTIONE ERRORI

```
pplx::task<int> risposta
    = CalcolaRispostaAllaVitaUniversoETuttoQuanto();

risposta.then([](pplx::task<int> antecedent) {
    try {
        int valore = antecedent.get();
        std::cout << "La risposta è " << valore << '\n';
    }
    catch (...) {
        // ...
    }
}));
```

Giochiamo con i task 

# TASK MONADIC CONTAINER

- impacchetta una computazione con side-effects
- il risultato viene spacchettato quando serve
- posso applicare una serie di trasformazioni con side-effect a un valore
- le signature dei task ci aiutano nella composizione

```
pplx::task<T> pplx::task_from_result<T>(T t); // Return  
pplx::task<U> t<T>.then(pplx::task<U>(*func)(T)); // Bind
```

# COMPONENTI DI CPPRESTSDK

- json
- stream asincroni
- uri
- http client
- http listener (no UWP, WinPhone8.1, XP)
- websocket client
- oauth/oauth2 client

# HTTP CLIENT

- Implementazioni:
  - WinHTTP (windows desktop)
  - IXMLHttpRequest2(UWP, WinPhone 8.1)
  - boost::asio + openssl
- Verifica certificati ssl con il sistema operativo
- HTTP 1.1
- Keepalive with chunked transfer encoding

# WEBSOCKET

- Implementazioni:
  - [websocketpp](#) + boost::asio + openssl
  - MessageWebSocket (UWP)

**DEMO TIME**

# DEMO TIME

http\_client, streams, websocket



# DEMO TIME

http\_client, streams, websocket

http\_listener

**DOMANDE?**

# DOMANDE?

Grazie 😎