

ESERCITAZIONE 9



STEFANO LENTINI

MARCO TURZI

ANDREA ZANNI

LEONE RUGGIERO

DAVIDE PERINTI

INTRODUZIONE

- ▶ APPLICAZIONE CLIENT SERVER CON USO DI RPC.

- ▶ REALIZZAZIONE DI DUE METODI REMOTI:

*int * esprimi_voto_1_svc(Input *input, struct svc_req *rqstp)*

*Giudici * classifica_giudici_1_svc(void *in, struct svc_req *rqstp)*

STRUTTURE DATI

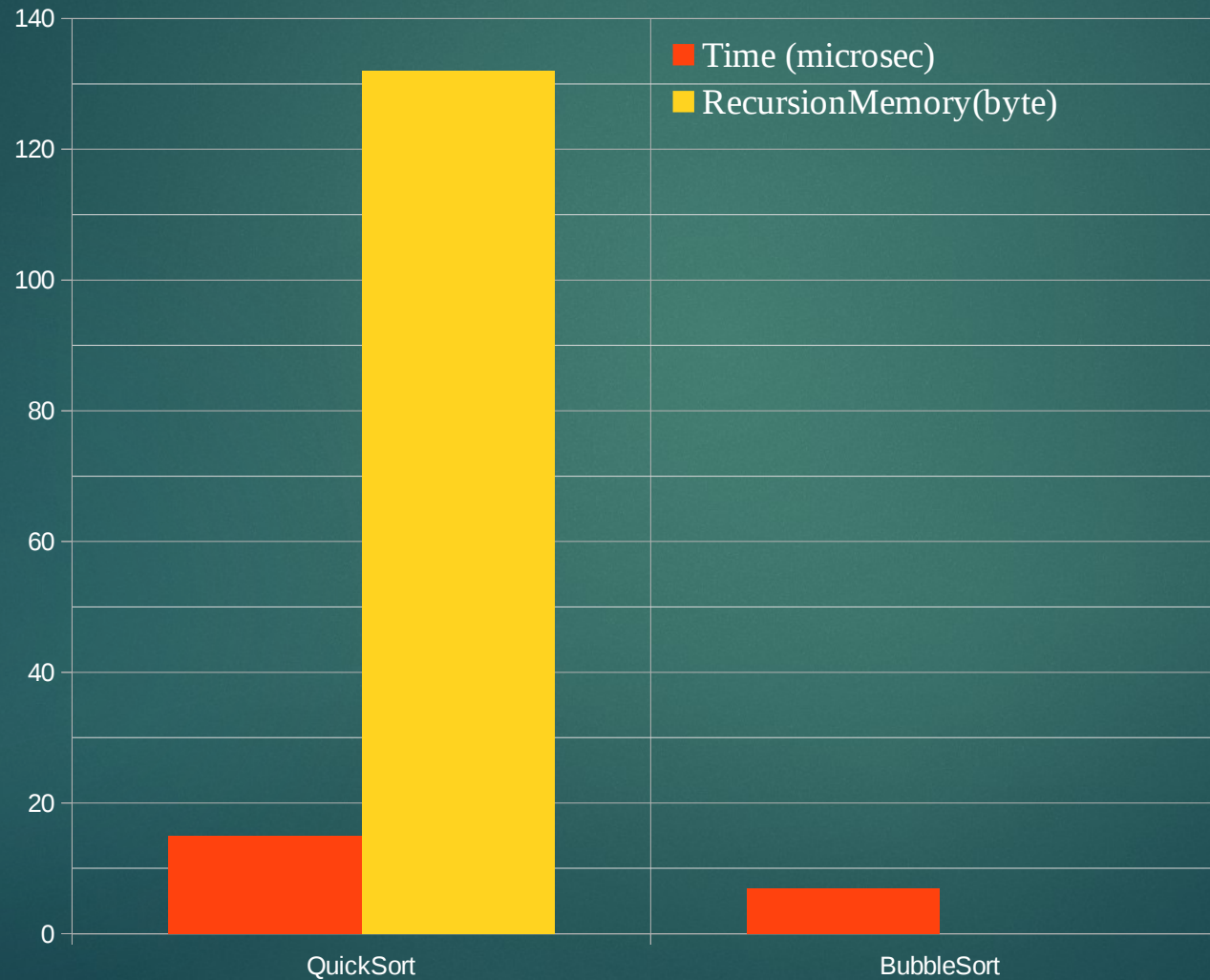
```
struct Candidato{  
    string nome <MAXNAME>;  
    string giudice <MAXNAME>;  
    char categoria;  
    string nomefile <MAXFILENAME>;  
    char fase;  
    int voto;  
};
```

```
struct Giudice{  
    string nome <MAXNAME>;  
    int punteggio;  
};
```

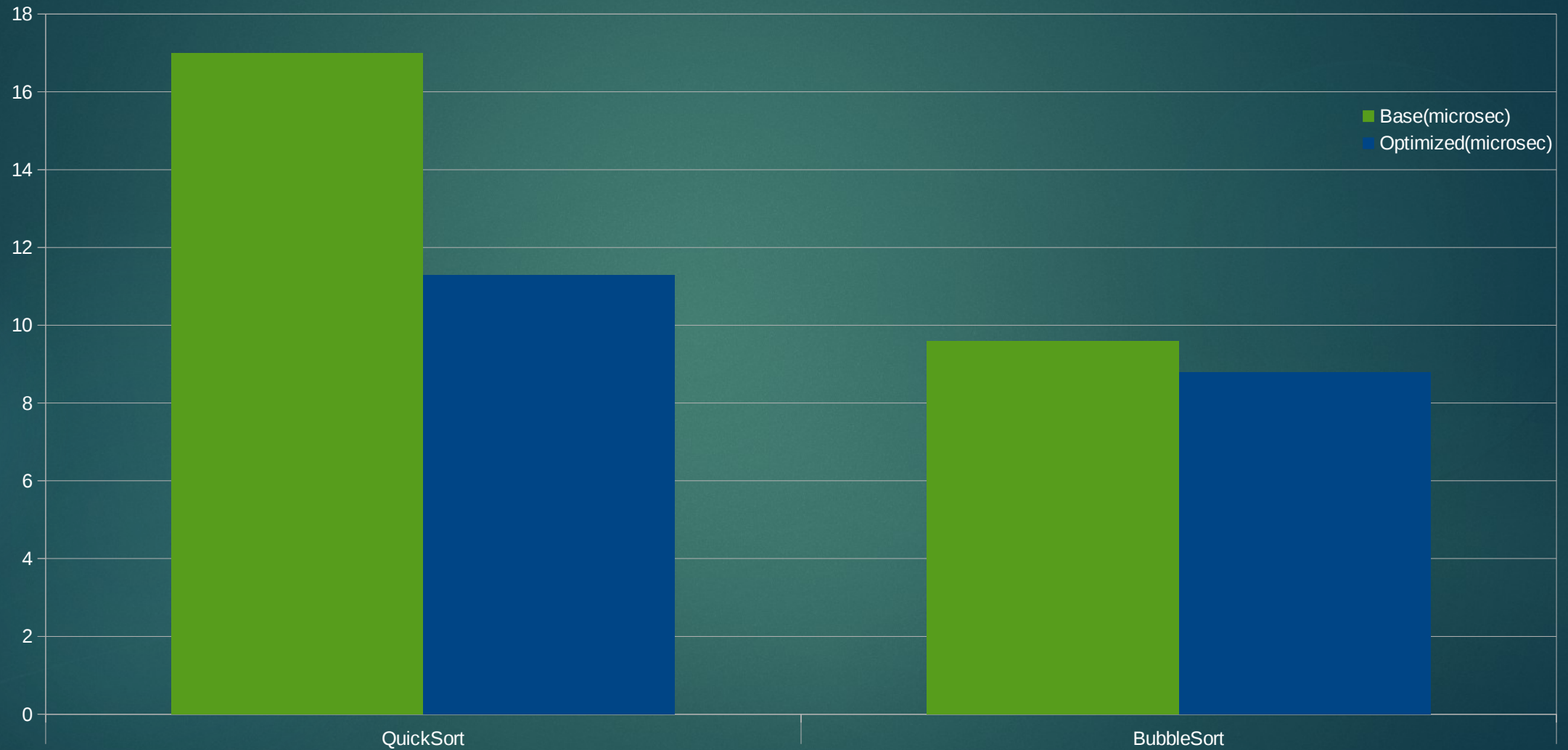

SCELTE SULLA CONNESSIONE

- ▶ ABBIAMO DECISO DI UTILIZZARE UNA CONNESSIONE DI TIPO UDP DATO CHE LE STRUTTURE UTILIZZATE NON HANNO DIMENSIONI MOLTO GRANDI E POSSONO ESSERE TRANQUILLAMENTE INVIATE CON UN SINGOLO DATAGRAM (maxDim=64Kbyte).

ALGORITMO DI ORDINAMENTO: PERFORMANCES



INTRODUZIONE della VARIABILE FLAG (GiudiciOrdinati)



TEST: DETTAGLI di IMPLEMENTAZIONE

```
for(cnt=0; cnt<numTest; cnt++)  
    {  
        int a=(int)(rand()%5);  
        ...  
    }
```

```
switch(a)  
{  
    case 1 :  
    {  
        start=clock();  
        classifica_giudici_1(in, cl);  
        stop=clock();  
        printf("CLASSIFICA: elapsed time:  
%0.4f\n", (stop-start)*1000/CLOCKS_PER_SEC);  
    }  
    default :  
    {  
        sprintf(input->nome, "%d",rand()%5);  
        input->operazione='+';  
        esprimi_voto_1(input, cl);  
    }  
}
```

CONCLUSIONE

- ▶ Nel modellare coppie C/S di questo tipo, dove il focus è il mantenimento di strutture dati sul server soggette a richieste da parte del cliente, non si devono sottovalutare le importanti implicazioni che la natura di tali strutture e tali richieste comportano.
- ▶ Queste considerazioni si traducono non solo in scelte riguardanti il tipo di connessione (in questo caso per esempio udp rispetto a tcp), ma anche in diverse scelte riguardo agli algoritmi impiegati e all'ottimizzazione.