



Eratostene distribuito

SCD

Anno accademico 2014/15
Sistemi Concorrenti e Distribuiti

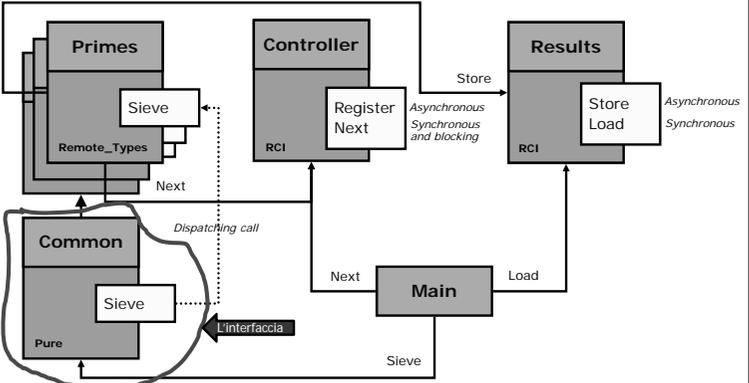
Tullio Vardanega, tullio.vardanega@math.unipd.it

Laurea Magistrale in Informatica, Università di Padova 1/11



Sistemi distribuiti: il modello di distribuzione di Ada

Architettura del sistema



Laurea Magistrale in Informatica, Università di Padova 3/11



Sistemi distribuiti: il modello di distribuzione di Ada

Algoritmo distribuito - 1

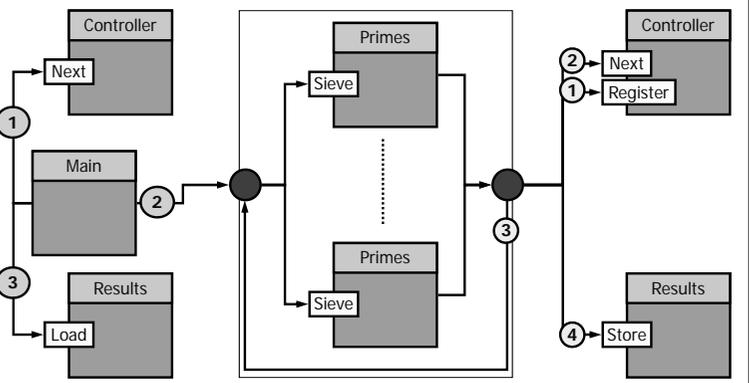
- ❑ Vogliamo distribuire l'equivalente logico delle unità Sieve del Crivello di Eratostene in versione centralizzata
- ❑ Invece di una discesa ricorsiva dinamica di partecipanti (uno per ogni primo) ne fissiamo staticamente un insieme che l'algoritmo visita circolarmente
 - Usiamo un *overlay network* in forma di anello
- ❑ Ogni unità mantiene localmente un sottoinsieme disgiunto dei numeri primi individuati
 - Ogni nuovo primo si deposita sull'unità che può dichiararlo come tale
 - Per farlo serve interrogare tutti i partecipanti che fanno da crivello
- ❑ L'unità Main (equivalente a Odd) attiva la ricerca in modo asincrono ma poi attende il risultato sincronamente

Laurea Magistrale in Informatica, Università di Padova 2/11



Sistemi distribuiti: il modello di distribuzione di Ada

Algoritmo distribuito - 2



Laurea Magistrale in Informatica, Università di Padova 4/11

Sistemi distribuiti: il modello di distribuzione di Ada

Algoritmo distribuito – 3

```

for Number in Search_Range loop
  Sieve(N, Number); -- asynchronous
  Load(Divider, Where); -- blocking
  if Divider = Number then
    -- Number is a prime
  else
    -- Node "Where" has the divisor
  end if;
end loop;
    
```

Laurea Magistrale in Informatica, Università di Padova 5/11

Sistemi distribuiti: il modello di distribuzione di Ada

Algoritmo distribuito – 5

- ❑ Ogni unità Primes mantiene localmente un sottoinsieme disgiunto di numeri primi
 - Ciascun primo si deposita su unità distinte visitate circolarmente
 - I sottoinsiemi locali non sono consecutivi
- ❑ Per decidere se un numero è primo può essere quindi necessario interrogare le unità Primes più volte
 - Di conseguenza il metodo Sieve di ogni unità Primes è potenzialmente ricorsivo in modo indiretto
 - Poiché Sieve lavora su valori statici il suo algoritmo deve essere reso o dimostrato robusto rispetto alla ricorsione
 - La chiave è l' incremento del valore sentinella Current prima della possibile chiamata ricorsiva
 - Quale l'effetto se l'incremento fosse ritardato?

Laurea Magistrale in Informatica, Università di Padova 7/11

Sistemi distribuiti: il modello di distribuzione di Ada

Algoritmo distribuito – 4

```

if Current <= Tail then -- sentinel check
  if (Number mod Prime(Current)) = 0 then
    -- we have a divisor so we record it
    Store(Prime(Current), Self);
  else -- we have no divisor so we ask next
    Current := Current + 1;
    Sieve(N, Number);
  end if;
else -- we have a new prime Who stores it?
  Tail := Tail + 1;
  Prime(Tail) := Number;
  Store(Prime(Tail), Self);
end if;
    
```

Laurea Magistrale in Informatica, Università di Padova 8/11

Sistemi distribuiti: il modello di distribuzione di Ada

Distribuzione – 1

Laurea Magistrale in Informatica, Università di Padova 8/11

Sistemi distribuiti: il modello di distribuzione in Ada

