

Simulazione Parallela e Distribuita

Approccio ottimistico



Gabriele D'Angelo

gda@cs.unibo.it

<http://www.cs.unibo.it/~gdangelo>

**Dipartimento di Scienze dell'Informazione
Università degli Studi di Bologna**

Sommario

- Il problema della sincronizzazione: approccio ottimistico
- L'algoritmo di Time Warp (by Jefferson)
 - Local Control Mechanism
 - *Rollback*
 - *Gli anti-messaggi*
 - *Cancellazione di eventi*
 - Global Control Mechanism
 - *Il problema della Fossil-Collection*
 - *Global Virtual Time*
- Confronto di prestazioni: ottimistico o pessimistico?
- Conclusioni

Il problema della sincronizzazione

Vincolo di causalità locale: gli eventi in ogni LP devono essere processati in ordine di time stamp

Nota: se ogni LP rispetta il vincolo, la simulazione parallela avrà gli stessi risultati della corrispondente esecuzione sequenziale

Algoritmi di sincronizzazione

- Approccio conservativo: *evitare la violazione del vincolo*
 - Null messages (Chandy/Misra/Bryant)
 - Reazione in caso di Deadlock
- Approccio ottimistico: *permettere la violazione del vincolo, rilevare l'avvenuta violazione, recovery per mezzo di rollback (ritorno allo stato "safe" precedente)*
 - Time Warp (Jefferson)

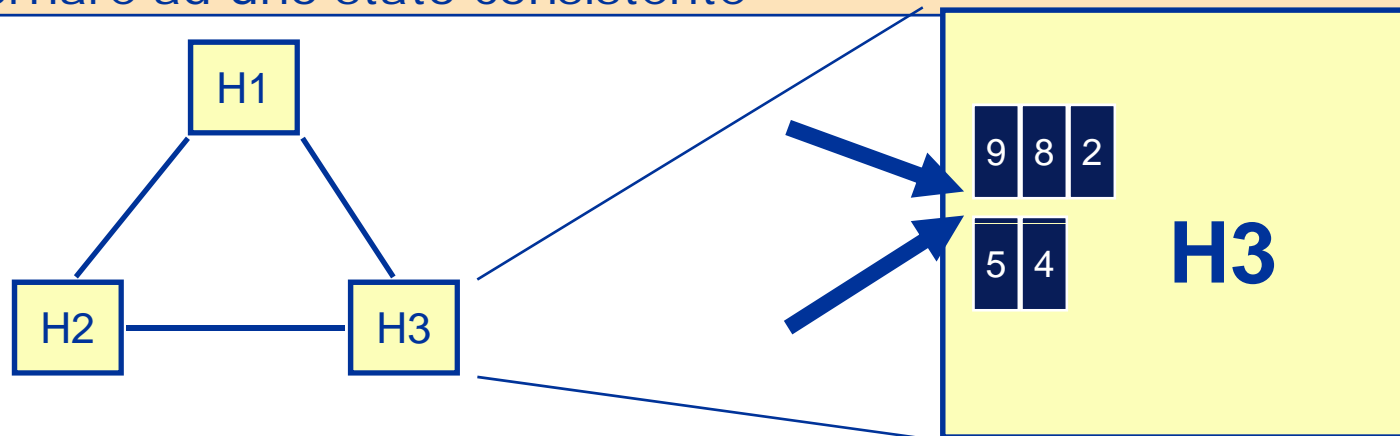
Time Warp (Jefferson)

Assunzioni:

- i logical process (LPs) si scambiano eventi time stamped (messaggi)
- la topologia della rete così come gli LP possono essere **dinamici**
- i messaggi spediti su un link non devono necessariamente essere in ordine di time stamp
- la rete è **affidabile** ma non necessariamente deve preservare l'ordine

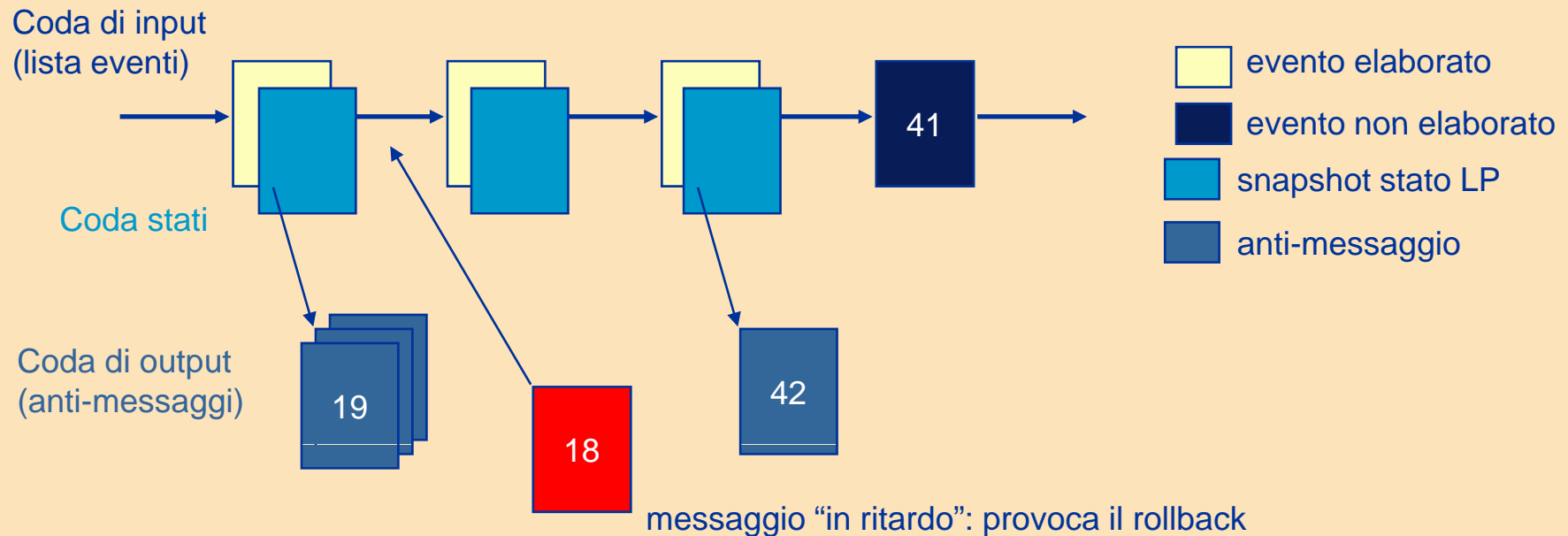
Idea:

- processare gli eventi senza preoccuparsi di cosa avverrà in seguito
- qualora avvenga una violazione di causalità, attuare un rollback per tornare ad uno stato consistente



Processa tutti gli eventi al momento disponibili (2, 4, 5, 8, 9)

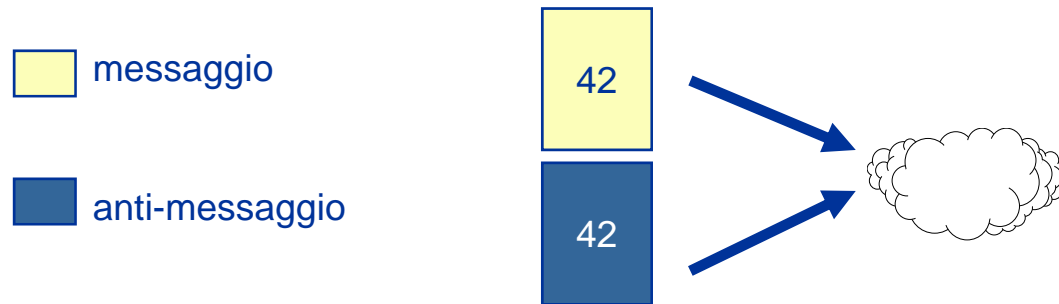
Ogni LP processa gli eventi in ordine di time stamp, avendo cura di (1) NON cancellare gli eventi già processati, (2) attuare un meccanismo di rollback



Implementazione del rollback:

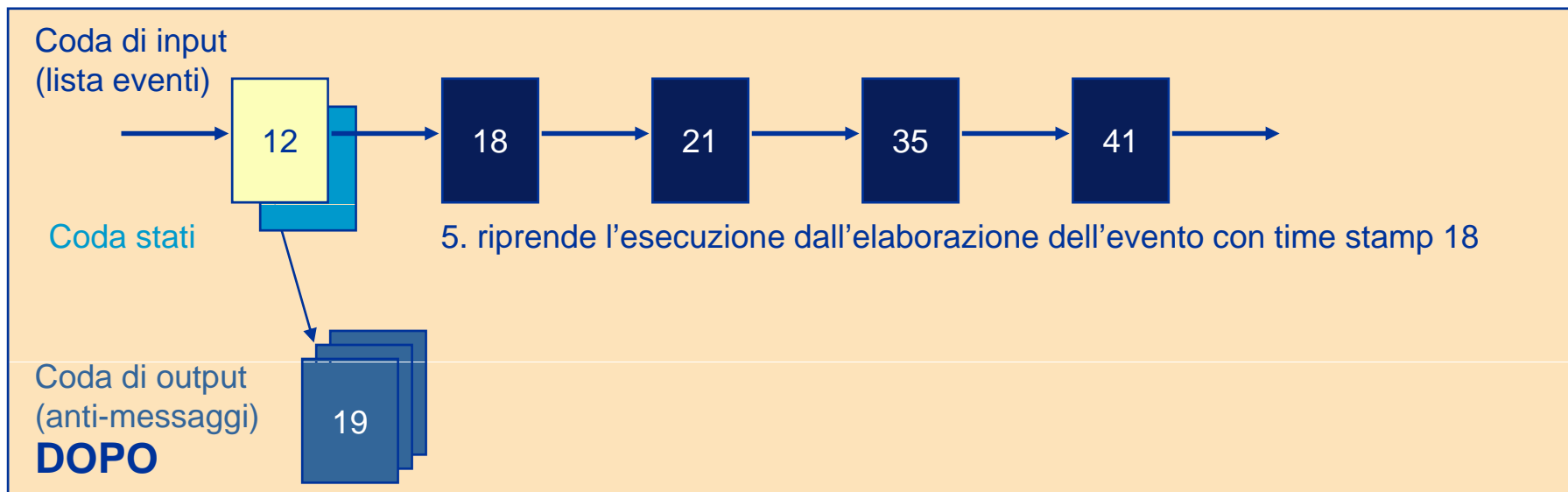
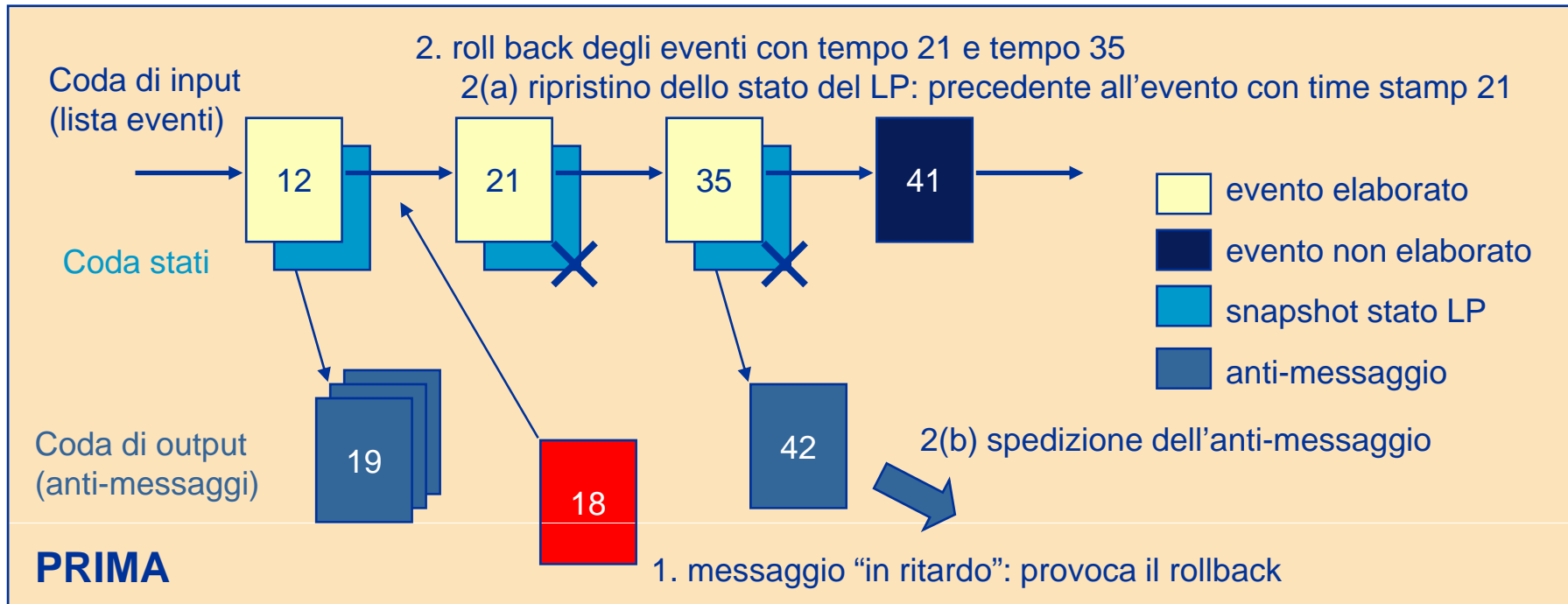
- l'arrivo di un messaggio con "tempo passato" causa rollback nel LP
- il rollback di un evento computato consiste in
 - ripristinare le variabili di stato interessate dall'evento;
Salvataggio di "stati incrementali", gestione dei checkpoint
 - considerare i messaggi già spediti
Anti-messaggi e "annichilimento" dei messaggi

Anti-Messaggi



- Il loro scopo è quello di **cancellare messaggi**
- Ogni messaggio spedito da un LP ha un “pontenziale” anti messaggio
- L’ anti-messaggio ha contenuto identico al rispettivo messaggio, ad eccezione di un bit di segno (quantomeno dal punto di vista logico)
- Quando un anti-messaggio e il rispettivo messaggio si incontrano nella stessa coda avviene un **annichilimento di entrambi**
- Per “rimediare gli effetti” dei messaggi precedentemente spediti, l’LP spedisce i corrispondenti anti-messaggi
- La spedizione di un messaggio comporta anche una copia del messaggio con segno negativo, mantenuta nella coda di output del LP (coda di anti-messaggi che potrebbero servire durante un eventuale rollback)

Ricezione di un messaggio in ritardo



Ricezione di un Anti Messaggio

Caso I: l'evento corrispondente non è ancora stato elaborato

- annichilimento della coppia messaggio/anti-messaggio

Caso II: il messaggio corrispondente è già stato elaborato

- roll back al tempo precedente all'elaborazione del messaggio
- annichilimento della coppia messaggio/anti-messaggio



sono possibili rollback "a cascata": esecuzione ricorsiva dell'algoritmo

Caso III: il messaggio corrispondente non è ancora stato ricevuto

- l'anti-messaggio viene mantenuto in coda
- l'annichilimento avverrà in futuro quando si sarà formata la coppia

Sommario

- Il problema della sincronizzazione: approccio ottimistico
- L'algoritmo di Time Warp (by Jefferson)
 - Local Control Mechanism
 - *Rollback*
 - *Gli anti-messaggi*
 - *Cancellazione di eventi*
 - Global Control Mechanism
 - *Il problema della Fossil-Collection*
 - *Global Virtual Time*
- Confronto di prestazioni: ottimistico o pessimistico?
- Conclusioni

Global Virtual Time (GVT)

Abbiamo la necessità di un algoritmo che:

- liberi la memoria occupata (salvataggi di stato ed eventi)
- gestisca le operazioni “irrevocabili” (ad esempio I/O)

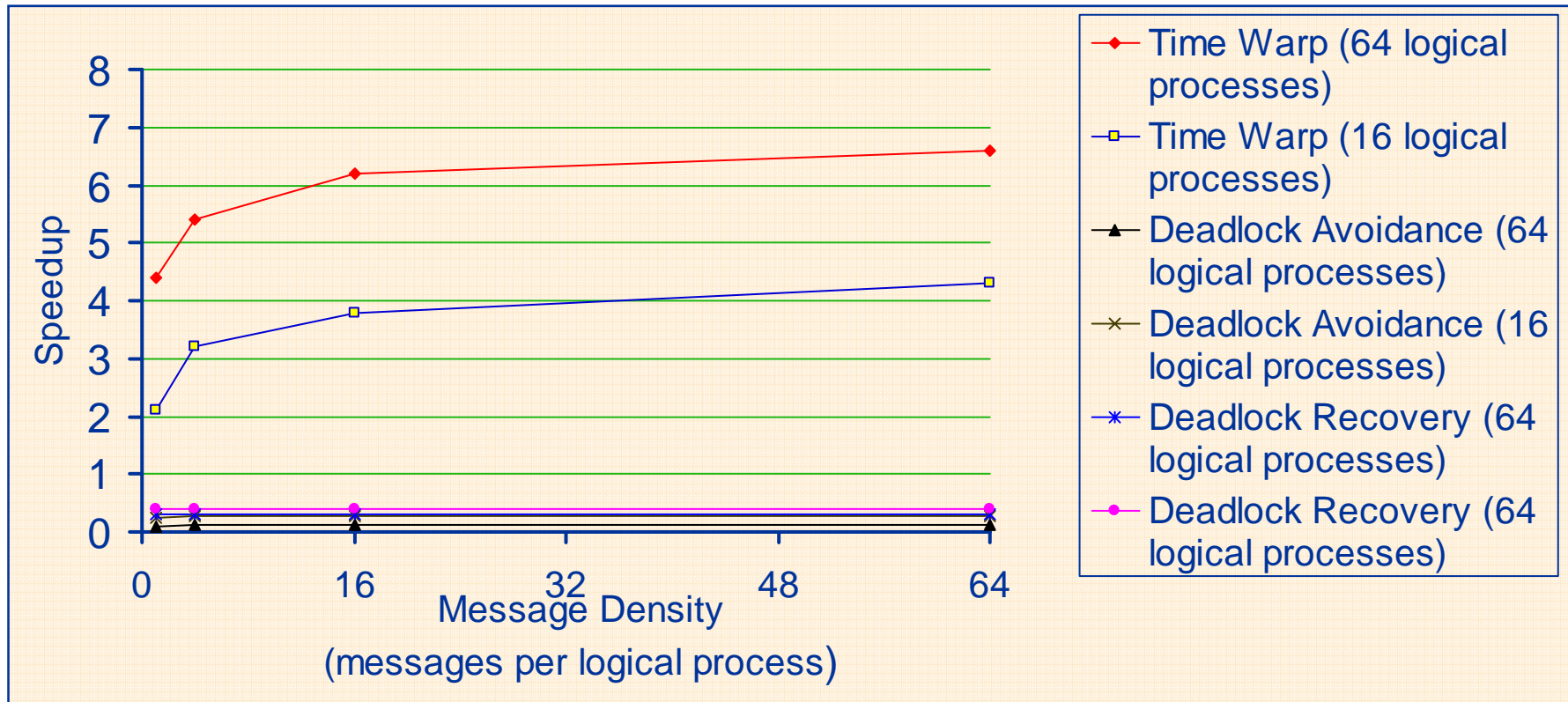
Soluzione: è necessario determinare un lower bound al time stamp dei rollback che possono avvenire in futuro

Global Virtual Time (GVT): il time stamp minimo di ogni messaggio o anti-messaggio, parzialmente o totalmente processato, all'interno del sistema in un dato momento. Il GVT fornisce un lower bound al time stamp dei rollback futuri

- lo spazio di memoria per eventi e salvataggi di stato, con tempo precedente al GVT, può essere deallocato (nel caso dello stato deve comunque essere presente almeno un salvataggio)
- Le operazioni di I/O con time stamp minore del GVT possono essere eseguite

Il GVT ci fornisce uno stato del sistema dal quale si può solo evolvere

Confronto prestazionale



La simulazione mostra un caso particolarmente favorevole all'approccio ottimistico, la disparità rispetto al caso conservativo è molto ampia.

Le prestazioni del confronto dipendono fortemente dal tipo di modello

Simulazione Parallela e Distribuita

Approccio ottimistico



Gabriele D'Angelo

gda@cs.unibo.it

<http://www.cs.unibo.it/~gdangelo>

Dipartimento di Scienze dell'Informazione
Università degli Studi di Bologna