DAPSYS Conference
Innsbruck, 21-23 September 2006

The wandering token
Congestion Avoidance of a Shared Resource

Augusto Ciuffoletti
INFN-CNAF - Bologna
CoreGRID Institute of Grid Information,
Resource and Workflow Monitoring Services

Problem and solution

- Problem: design a **scalable** (thousands of agents), resource sharing solution
- Requirements:
  - access granted regularly, not strict real time;
  - overload gradually degrades service, with limited impact on the resource;
  - algorithm runs on clients; resources and infrastructure are legacy

A case study: Video on Demand

- A provider offers a VoD service to a number of susbscribers (news on a 150*120 screen)
- Video is trasferred using a 200 Mbps infrastructure
- Each transmission uses a 650 Kbps bandwidth
- Each subscriber downloads distinct chunks of video
- The server uploads chunks of video on demand
- The infrastructure allows short bursts of transfer rate up to two times the agreed 200Mbps

Other applications

- Intrusive network monitoring applications
  - only one test per run to avoid interference and overload
- Expensive queries to a centralized database
  - only one query at a time allowed

The token wandering idea

- A unique token circulates in the system, enabling members to access the resource only when holding the token.
- Token circulation does not follow a predetermined path: next hop destination is selected randomly among all members.
- According to many theoretical results, the interarrival time is quite regular, although randomly variable.
- Like any token circulation algorithm, it suffers from token loss and duplication.

Token (re)generation rule

- The agent sets up a timer: when such timer expires, a new token is generated, bearing a unique identifier.
- The timer value has the following properties:
  - it is randomized, to avoid synchronization effects;
  - minimum value considers application requirements, and is well below the required access rate;
  - expected value is such that the **overall** distribution of timeouts on the time line has an average interarrival time below the required access rate;
- The token regeneration rule can **introduce** spurious tokens (with different id).

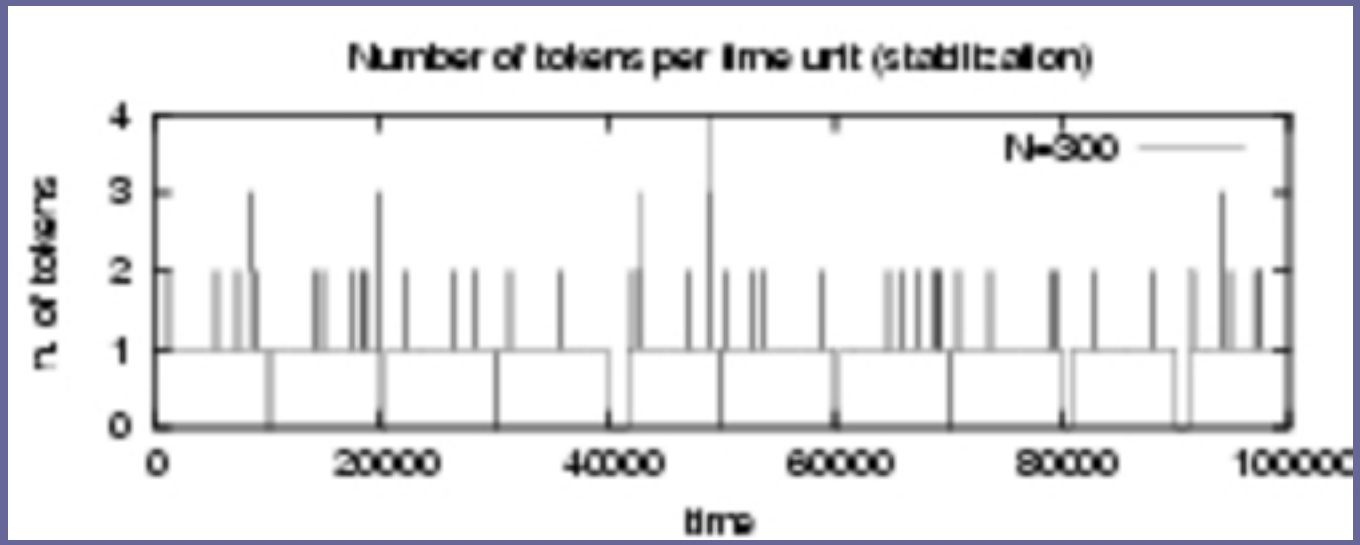
Token removal rule

- This rule aims at removing tokens unnecessarily generated by the token regeneration rule
- Duplicate tokens are prevented by the 3-way token passing protocol
- The token removal rule is applied whenever a token is received.
- Such token is **silently dropped** if:
  - the same token was already received and
  - from that time another token was received with a timestamp lower than that of the resident token
- Removal occurs with a latency, and during that time the token can be effectively lost.

Simulation setup

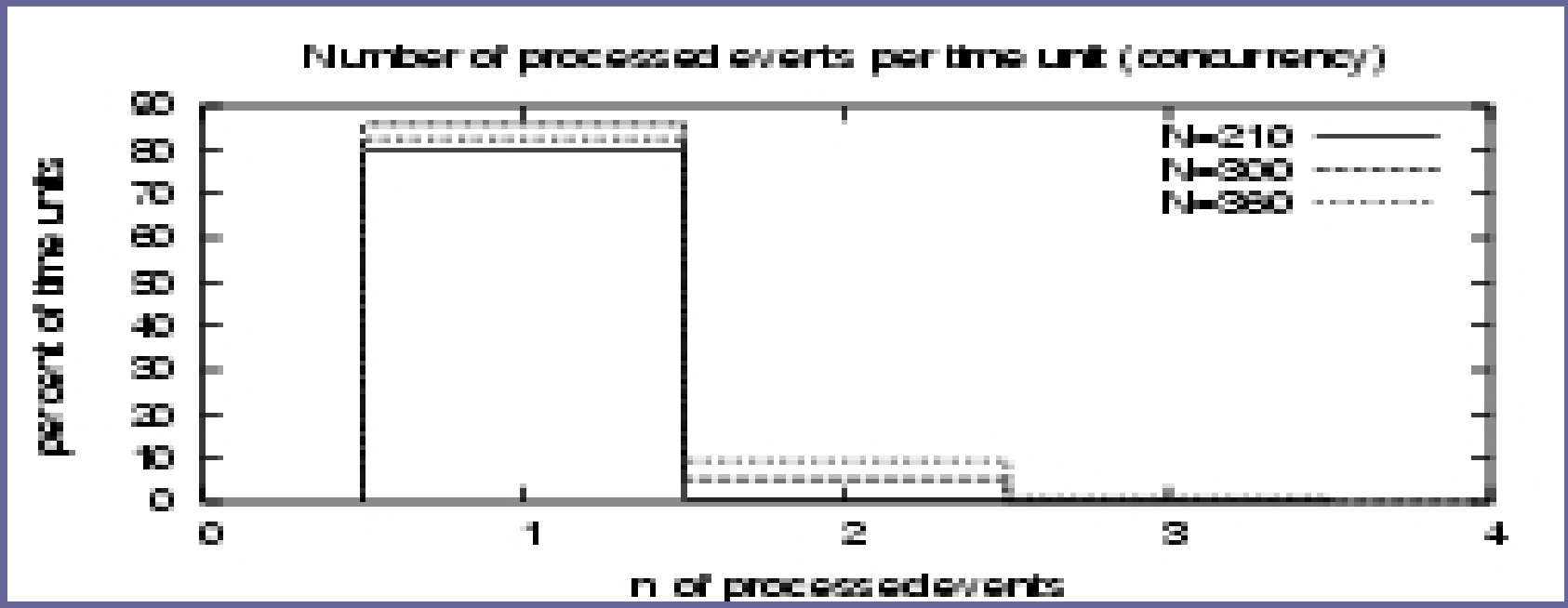
- The simulator is an ad-hoc finite state machine that simulates a variable number of agents:
  - synchronous token passing operations, and
  - asynchronous alarms.
- The case study (650 Kbps VoD on a 200Mbps backbone) accommodates up to 300 subscribers.
- The access rate should be on the average one every 1200 seconds.
- Each subscriber downloads a chunk of 100 Mbytes during a slot of time of 4 seconds

Simulation results: n. of tokens



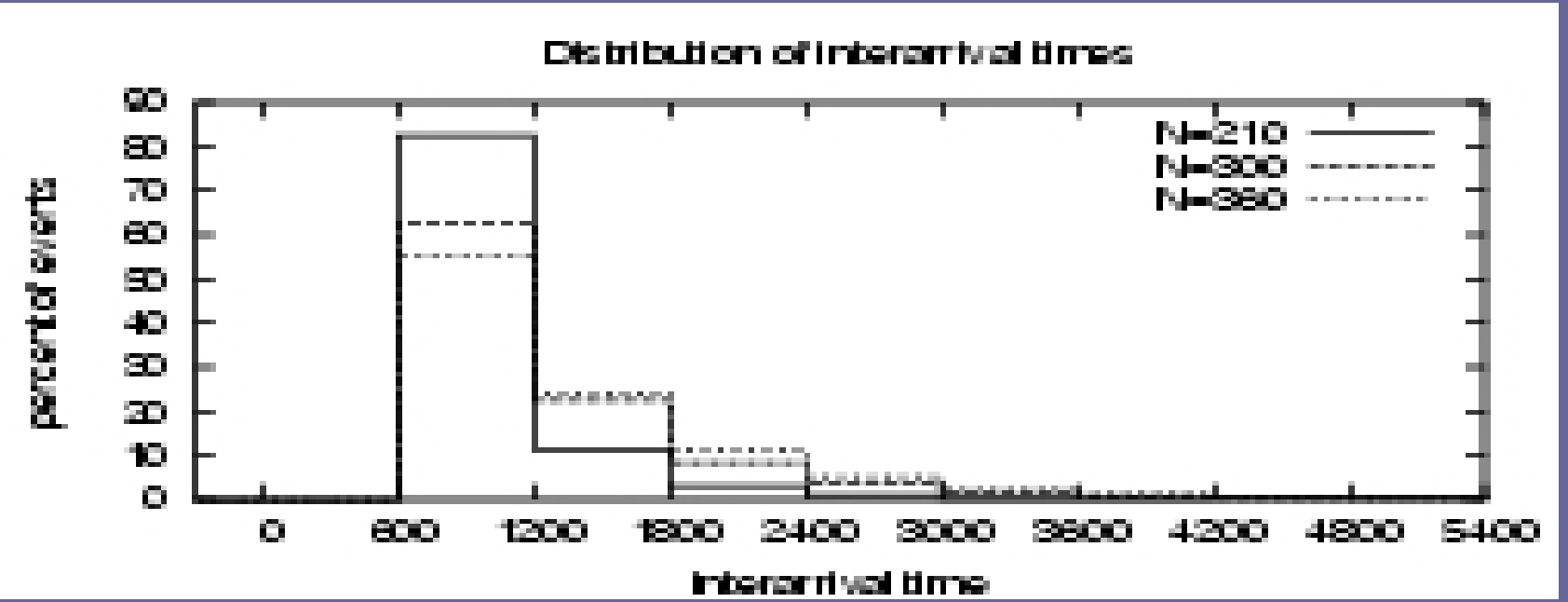
- **The requirement of a unique token is enforced**
- Token loss events (injected one every 10000 time units) are recovered: largest gap is 1700 time units.
- No transients after token loss recovery.
- Limited presence of spurious tokens

Simulation results: concurrency



- **Resource is protected from overload**
- Even under 20% overload:
  - 80% of time there is just one protected event running
  - during less than 10% of the time there are two concurrent downloads (expensive rate applied)
  - during 1% of the time there are three or more concurrent downloads (shaping may occur)

Simulation results: interarrival time



- **Observed performance depends on load**.
- Most events occur before 1200 time units, even on full or overload conditions.
- Longer tail in case of full and overload conditions (buffering needed)

Conclusions

- A **Proof of Concept**: a random walk approach can provide a robust, scalable and efficient solution to a distributed coordination problem.
- A solution to a class of problems: we provide a solution to the **Soft Mutual Exclusion** problem, based on a wandering token.
- A real world case study: the simulation in a **Video on Demand** environment demonstrates applicability.