Deadline-Driven Auctions for NPC Host Allocation in P2P MMOGs

MMVE'09

Lu Fan, Phil Trinder, and Hamish Taylor
Heriot-Watt University, Edinburgh, UK

Overview

- Introduction
- Related Work
- DDA Design
- Design Analysis
- DDA Aspects
- Implementation
- Evaluation
- Discussion

Introduction

- In a MMOG large numbers of users interact via avatars with
  - each other
  - NPCs: AI-controlled virtual actors who drive storylines or combat player characters

Introduction

- Client/Server (C/S) Architectures
  - Dominant architecture for conventional MMOGs
  - Exhibit cost & reliability drawbacks as they scale up…
  - Hence, research interest in P2P MMOGs
  - Entail a set of design issues…
- Among these issues
  - NPC hosting is a key challenge
  - Previously, NPCs were hosted by game servers
  - Now need to be hosted by peers

Introduction

- Deadline-Driven Auctions (**DDA**) for **NPC** hosting in P2P **MMOGs**
  - A distributed task mapping infrastructure
  - Highly heterogeneous environment
    - Peers
    - Communication latency
    - Task size/deadlines

How have NPCs been hosted previously?

- Region-based approaches
  - Partition a game world into multiple regions
  - Select a super-peer in each region
  - The super-peer hosts all the NPCs in its region
  - E.g. ***P2P Support '04***, ***Zoned Federation '04***
- Virtual-distance-based approaches
  - Attempt to distribute NPCs to more game participants
  - Allocate NPCs according to locality in game world
  - A NPC is hosted by the closest player
  - E.g. ***AtoZ '04***, ***Colyseus '06***, ***Voronoi '08***

Pros & Cons

- **Region-based**

☺ Easy to implement

☺ Easy to secure

☹ Lack of load-balancing

☹ Super-peer selection issue

☹ Super-peer dependability issue

☹ No guarantee of communication latency

Pros & Cons

- **Region-based**

☺ Easy to implement

☺ Easy to secure

☹ Lack of load-balancing

☹ Super-peer selection issue

☹ Super-peer dependability issue

☹ No guarantee of communication latency

- **Virtual-Distance-based**

☺ Better resource utilisation

☺ Minimises communication latency

☹ Not optimal for 1:N interactions

☹ NPC host switching delays

☹ Higher computation overhead

☹ Hard to prevent/detect cheating

DDA Design

General Design Aims

- ***Self-Organisating:*** infrastructure is assembled & managed automatically
- ***Real-time Resource Allocation:*** large number of tasks within deadlines
- ***QoS for 1:N Interactions:*** minimise latency between NPC hosts & players
- ***Cooperative Economic Model:*** provide an incentive mechanism & shares tasks fairly

DDA Specific Design Objectives

- Specific Design Objectives:
  - ***Efficient Resource Utilisation***

    Allocate NPC hosts according to resource availability
  - ***Game Interactivity***

    Reduce communication latency for 1:N interactions
  - ***Efficiency***

    Keep up with the fast pace of a MMOG
  - ***Dependability***

    Recover from a range of exceptions
  - ***Viability***

    Persuade application participants to contribute resources

DDA Design

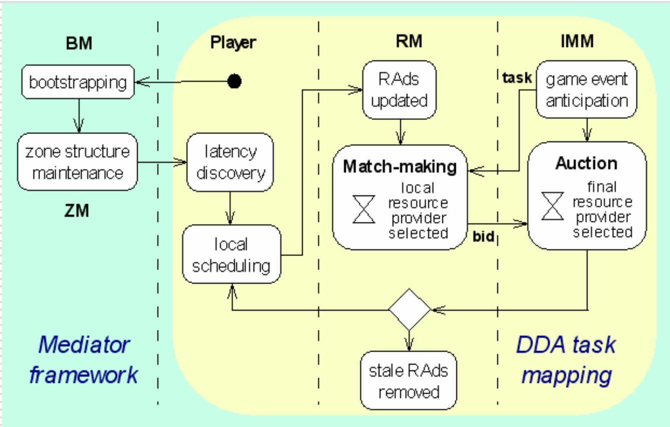
- System Model
  - Abstraction of a NPC tasks
    - **indivisible** – hosted by a single peer
    - **computational** – consumes processing power
    - **interactive** – communicates with other peers
    - **real-time** – must start working before deadline
  - Task mapping in a network with heterogeneous peers, tasks & communication links
  - System components
    - **Work Source** – that generates NPC tasks
    - **Resource Providers** – the peers
    - **Matchmakers** – a super-peer infrastructure

DDA Design

- Self-Organising Super-Peer Infrastructure
  - Provided by ***Mediator***
    "Mediator: A Design Framework for P2P MMOGs", NetGames '07.

• Boot Mediator – bootstrapping
• Zone Mediator – zone maintenance

• Player – **Resource Providers**
• IM Mediator – **Source of NPC tasks**
• Resource Mediator – **Matchmakers**



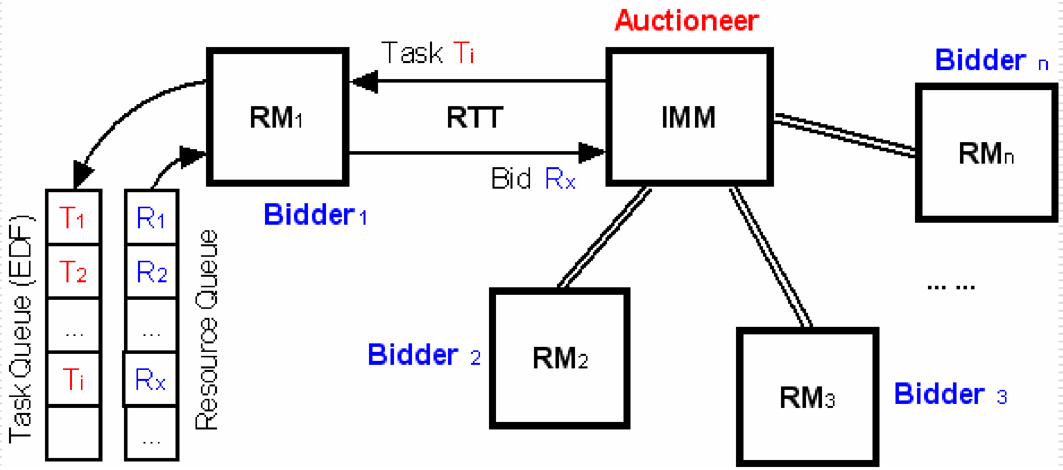
- **Task Mapping** – to allocate a NPC to a peer with
  - adequate computing resources
  - low comm. latency to peers 'near' the NPC

DDA Design

- Local Scheduling
  - Game participants' own scheduling activities
  - Includes:
    - Disclose resource availability to an RM
    - Discover communication latency to other zone members
    - Volunteer for super-peer backup at the ZM
  - Driven by accounting mechanisms
    - Reward according to contributions
    - Charge according to playing time
  - Reinforced by reputation mechanisms
    - Discourage false resource offers
    - Punish anti-social behaviours

DDA Design

- Zone-Level Scheduling
  - Cooperation between an IMM & multiple RMs



IMM:

- Distributes NPC task requests to RMs
- Maintains auction for each task

RM:

- Buffers the NPC task requests
- Matches tasks to resource offers
- Sends bid to IMM with good resource

Design Analysis

- Meeting the deadline for each task
  - Every match-making task T_i must be completed before its deadline D_i :

$$\forall_{i \in 1 \ldots n} \bullet C(T_i) < D_i \qquad (1)$$

  - Completion time $C(T_i)$ comprises:
    - Time for processing the previous *i-1* tasks
    - Time for RMs' match-making
    - Time for the communication among IMM & RMs
  - Deadline D_i is determined by:
    - Spawning interval – for periodically spawned NPCs
    - A specific time – if the NPC is triggered by a game event

Design Analysis

- Variables involved in our analysis:

Variable	Meaning	Nominal Value
P	zone population	
R	NPC : PC ratio	5 : 1
TTL	NPC life time expectation	300 (second)
r	event triggering rate	1/60 (per second)
Int	respawning interval	
RTT	round trip time	0.5 (second)
l	RM resource queue length	50 (RAds)
t	matchmaking time	1 (ms per RAd)

- To meet NPCs' deadlines, should satisfy this equation

$$RTT + \left(\frac{P * R}{TTL} + r * P\right) * Int * l * t < Int \qquad (4)$$

Design Analysis

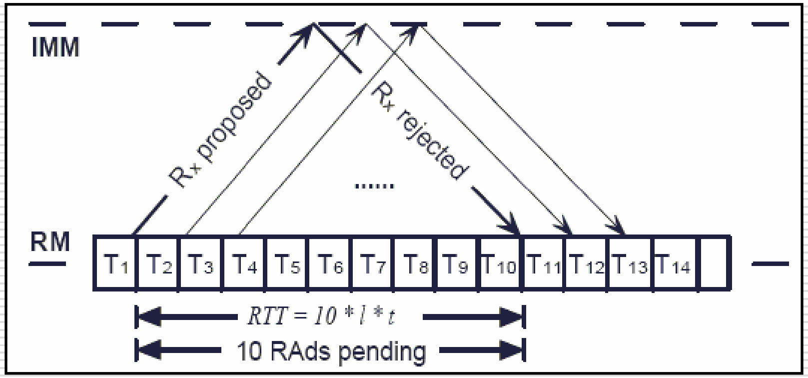
- Result & Inferences:

$$(600 - P) * Int > 300 \qquad (5)$$

  - According to the assumptions, the design can support up to 600 players in each game zone.
  - For a P2P MMOG whose maximal zone population is 500 players, the minimal spawning interval is 3 seconds.
  - As a peer may obtain a NPC task in every 10 spawning intervals, the credits awarded should exceed corresponding playing costs.

DDA Aspects 1: Reducing Resource Tie-up

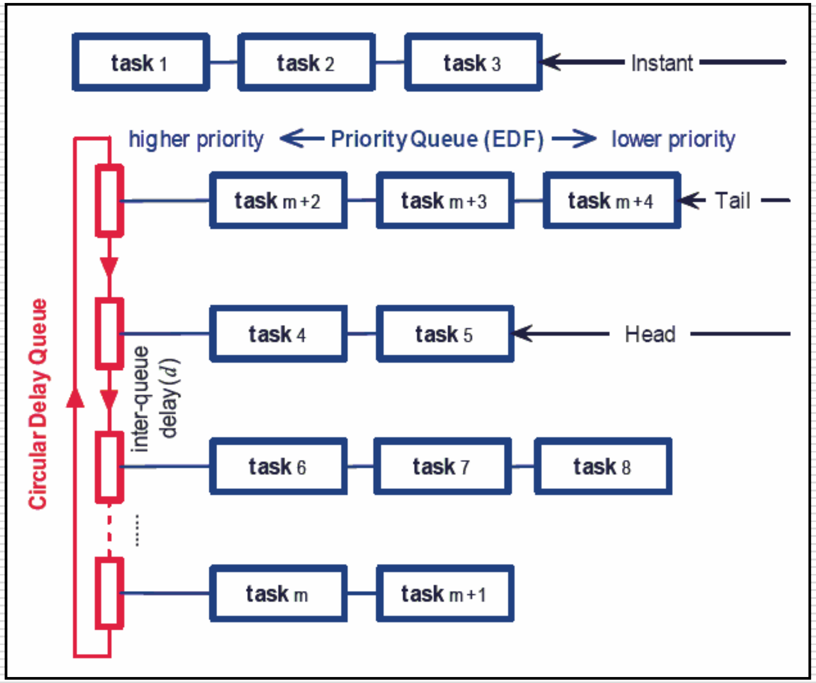
- The Resource Tie-up problem
  - RMs’ match-making time is short compared with communication time
  - As a result, many rejected resource bids are tied up



  - RMs should slow down match-making process, but still guarantee that all tasks meet their deadlines.

DDA Aspects 1: Reducing Resource Tie-up

- Solution: RMs buffer NPC tasks using a Multilevel Feedback Delay Queue (MFDQ)
  - A MFDQ comprises multiple run queues
  - Each run queue is Earliest Deadline First (EDF)
  - Run queues are organised in a cycle: ***Instant*** → ***Head*** → ***…*** → ***Tail***
  - A run queue is executed when its delay has expired



DDA Aspects 2: Flexible Match-Making Policies

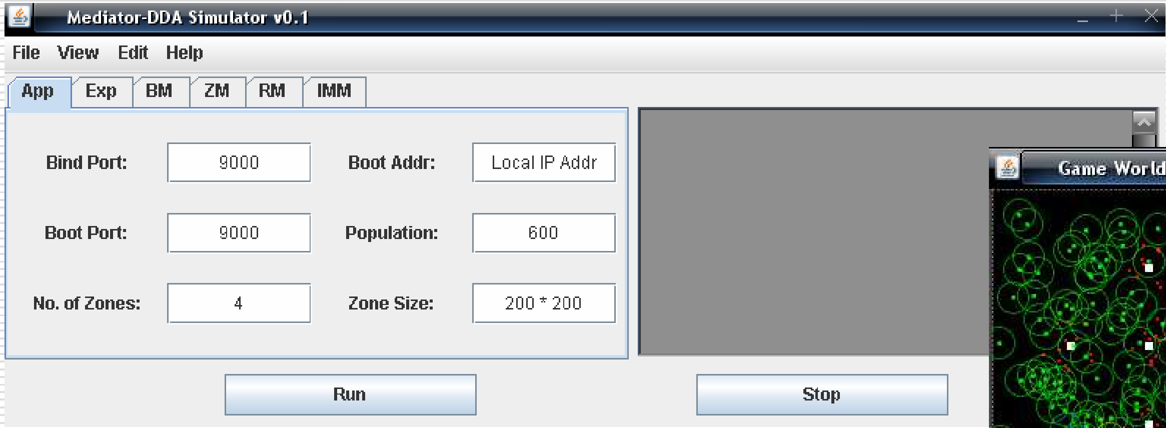
- A friendly incentive policy
  - By default RMs select resource providers offering best game interactivity.
  - Hence less competitive peers are starved of credits
  - A friendly policy relaxes selection criteria & favours poor peers with acceptable interactivity.
- Other possible policies
  - RM may favour a trustworthy peer according to its reputation.
  - RM may favour a dependable peer according to its history.
  - RM may favour a senior peer for its loyalty to the game.
  - …

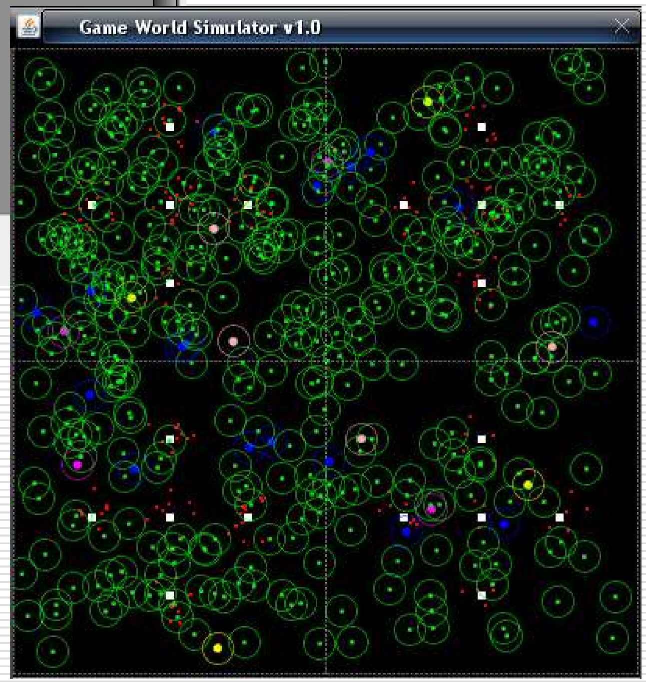
DDA Implementation

- DDA Prototype
  - Self-organised super-peer infrastructure
    - Uses Java-based FreePastry 2.1
    - Super-peer dependability enhanced by ***MAMBO*** "Membership-Aware Multicast with Bushiness Optimisation", DEBS '08.
  - Match-making mechanism
    - Uses ClassAds 2.2 from Condor
    - Local scheduling simulated by virtual resource managers
- Test-bed Application
  - Uses the Direct discrete event simulator
  - Network topology model created by GT-ITM
  - 2D game world & random way point algorithm

DDA Implementation

❖ Screen Shot



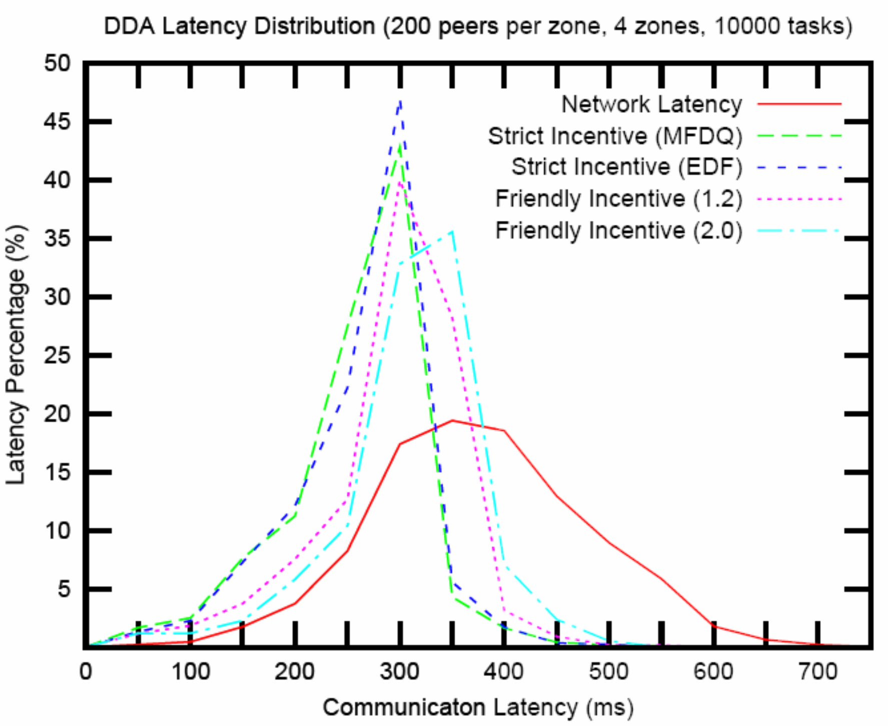


- 2.4GHz Intel Core2 Duo CPU + 2GB RAM
- Game sessions with 4 game zones, initially 200 players per zone, 800 players in total
- Communication latency & credit distribution measured using 10,000 NPC tasks

Evaluation

- Experimental Results
  - Real-time Resource Allocation

- All tasks' deadlines are met.
- 90% latencies below mean network latency using EDFQ
- 95% latencies below mean network latency using MFDQ
- More friendly incentive policies have more latencies greater than the mean network latency



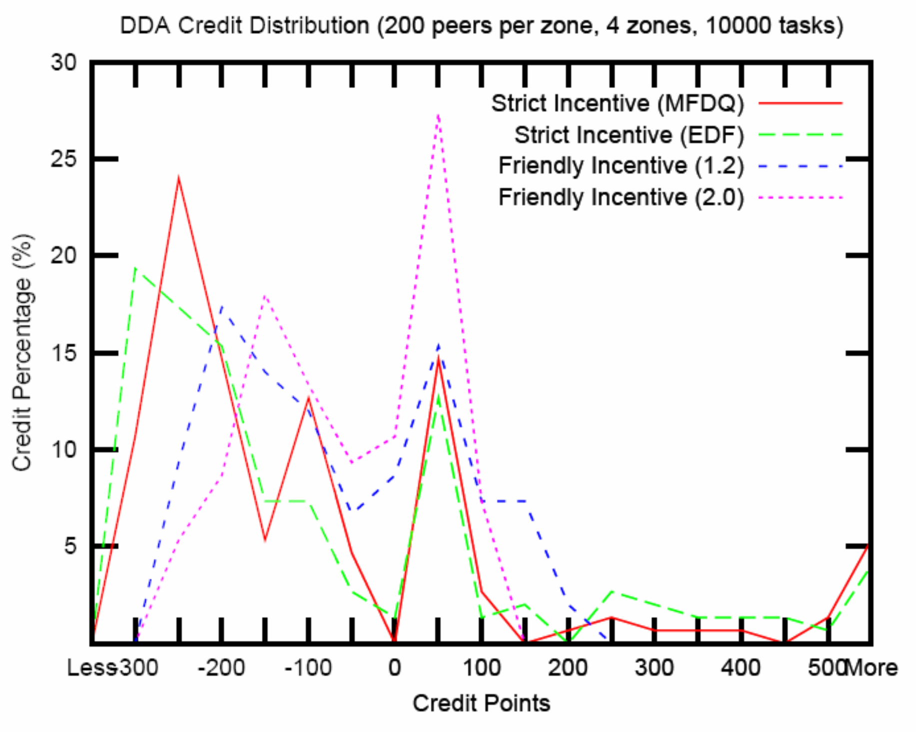
Evaluation

- Cooperative Economic Model

➢ Economic gap is significant using strict incentive policy (5% peers are richer than 500, while 20% peers are poorer than -300)

➢ This gap is narrowed using friendly incentive policies.

➢ Low resource peers are always in debt



Discussion

- DDA's Strengths
  - Task mapping infrastructure in a highly heterogeneous environment
    - Assembled & Managed automatically
    - Super-peers are fault-tolerant, e.g. sustain peer churn of 5% per hour
    - Meets deadlines for large numbers of tasks
    - Provides DCRC-like incentive mechanism & shares tasks fairly
  - Compared to virtual-distance-based approaches
    - Reduces communication latency for 1:N interactions
    - Allocates NPC hosts according to actual resource availability
    - Applicable to any P2P application requiring real-time computation & interactive tasks

Discussion

- DDA's Limitations
  - Opportunistic resource allocation
    - Only optimises game interactivity for players in the vicinity of an NPC's initial position, but
    - both NPCs and player avatars are mobile
    - New players may arrive
  - Not ideal for 1:1 interactions
    - NPCs are likely to be hosted by other peers, inducing communication latency & overhead
    - A supplement rather than a substitute for virtual-distance-based approaches

Discussion

- Conclusions
  - DDA is a novel NPC host allocation mechanism for P2P MMOGs, using heterogeneous task mapping
  - DDA design and analysis
  - DDA implementation & evaluation
  - DDA evaluation

- Future Work
  - To enhance NPC host dependability with a reputation system
  - To explore DDA's usage for other P2P applications, e.g. distributed video encoding

Thank you for your attention!

Q & A

Lu Fan lf16@hw.ac.uk

Phil Trinder p.w.trinder@hw.ac.uk

Hamish Taylor h.taylor@hw.ac.uk

School of Mathematical and Computer Sciences

Heriot-Watt University, Edinburgh, UK