Integration of P2P and Clouds to Support Massively Multiuser Virtual Environments

MMVE/Netgames – Taipei 17 Nov 2010

contrail

Laura Ricci
Massimo Coppola
Emanuele Carlini

XtreemOS
Enabling Linux for the Grid

Contrail is co-funded by the EC 7th Framework Programme

Outline

- Initial design of an MMVE architecture
  - Exploiting and adapting existing techniques from different fields
  - Focus on passive objects
- Motivations
- Architecture issues
  - Passive object management
    - Latency-aware mapping
  - Interest Management
  - Consistency Model
- Conclusions

Motivations

- Client/Server: classical commercial solutions
  - Expensive: size, scalability, operating costs
  - Total platform control: trust and reliability
- P2P architectures
  - P2P scales cost-free, unreliable
- Clouds sit somewhere in the middle
  - Dynamic and reliable, but not free

- Our aim: merging together these resources

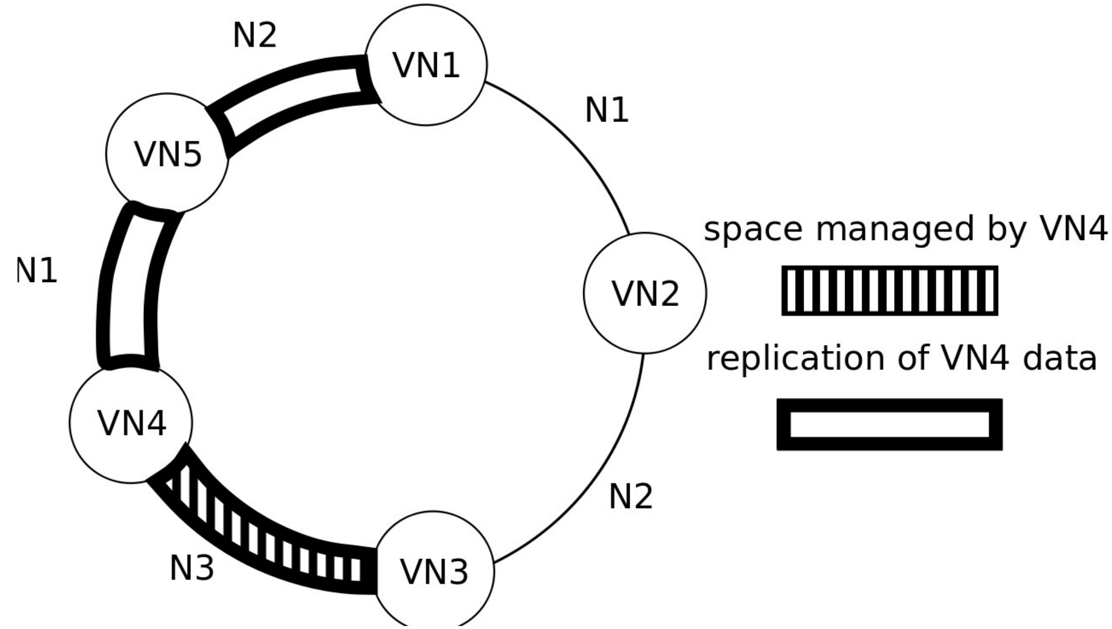
Aimed Vision

- We target next generation clouds
  - In the future the problem will be the bandwidth: easy to add cluster, costly to add bandwidth
- Our approach is promising but is still to be validated
  - Easy to integrate different kind of computational resources (even potentially non trusted)
  - Built-in mechanisms to provide failure tolerance
  - Latency-aware task assignment
  - Backup for sensible information
  - Optimistic consistency model

Passive Objects Management

Object Management

- Distributed Hash Tables
  - Consistent hashing
- Virtual nodes
  - [Stoica et al. 2001, 2004]
- Basic properties
  - Replication
  - Load balancing
  - Mapped on real nodes

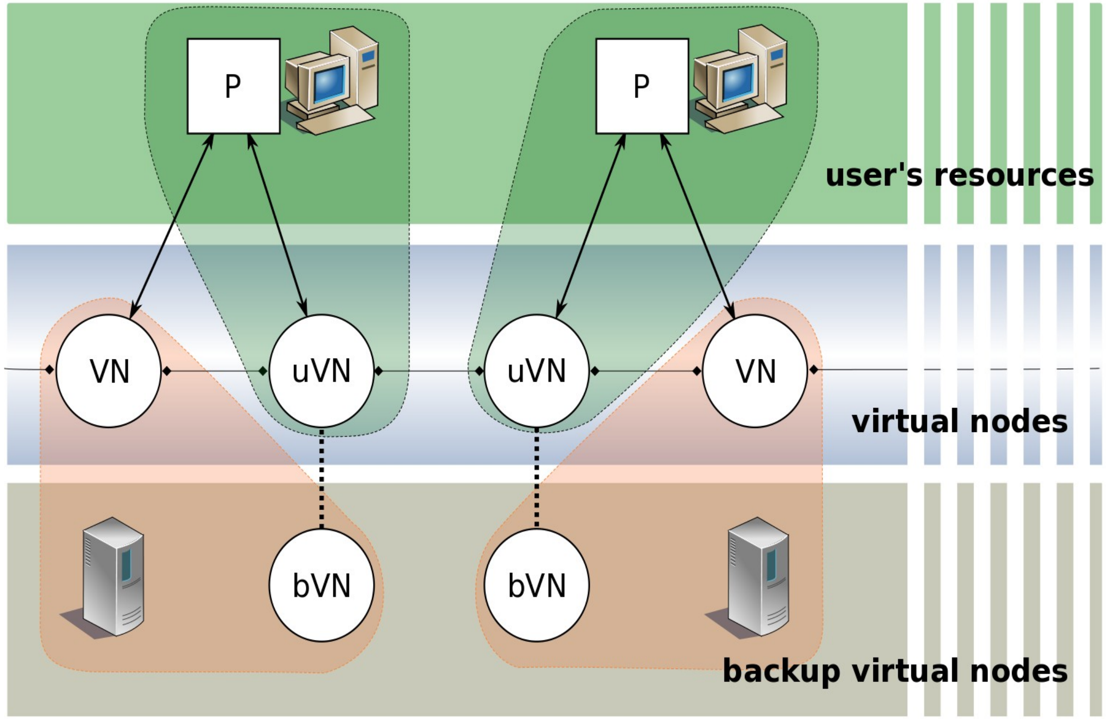


Virtual Nodes for Passive Objects

- Passive objects are spread on DHT
- A VN contains the state of objects
  - Acts as the server
  - Security checks – cheating mitigation
- Maintains DHT infrastructure
  - Manages connections – routing tables
  - Replicas management
- VN as workload unit in the DHT
  - Cost of each VN is measurable in terms of
    - Required bandwidth
    - Computational power

Backup Virtual Nodes

- Distribution of VNs among heterogenous resources
  - Free resources are unreliable
  - bVN is spawned whenever a VN is assigned to an unreliable node



Backup Virtual Nodes

- BackupVN manages untrusted/unreliable resources
  - Assigned to a reliable node
  - Acts as secondary server
  - Does cheating mitigation
- The “unreliable” VN becomes a uVN – limited functionalities
  - Maintains the topology connections
  - Manages the status of the objects
  - Periodically sends updates to bVN

Locality-aware mapping of Virtual Nodes

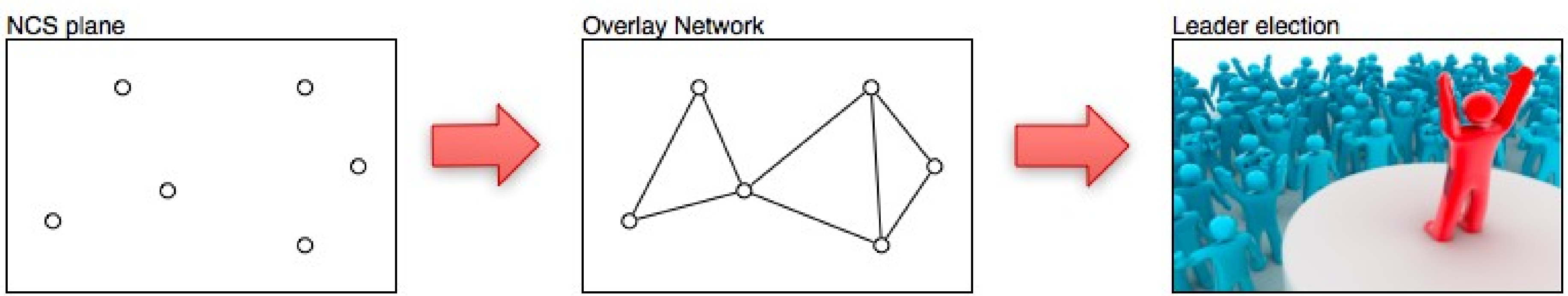
Virtual Nodes Reallocation

- VNs are distributed among different resources
  - Reallocation
- Load balancing
  - Relief heavy loaded machines
  - Turn-off poorly used machines
- Node failure

- Interactivity
  - Latency

Latency-aware VN Mapping

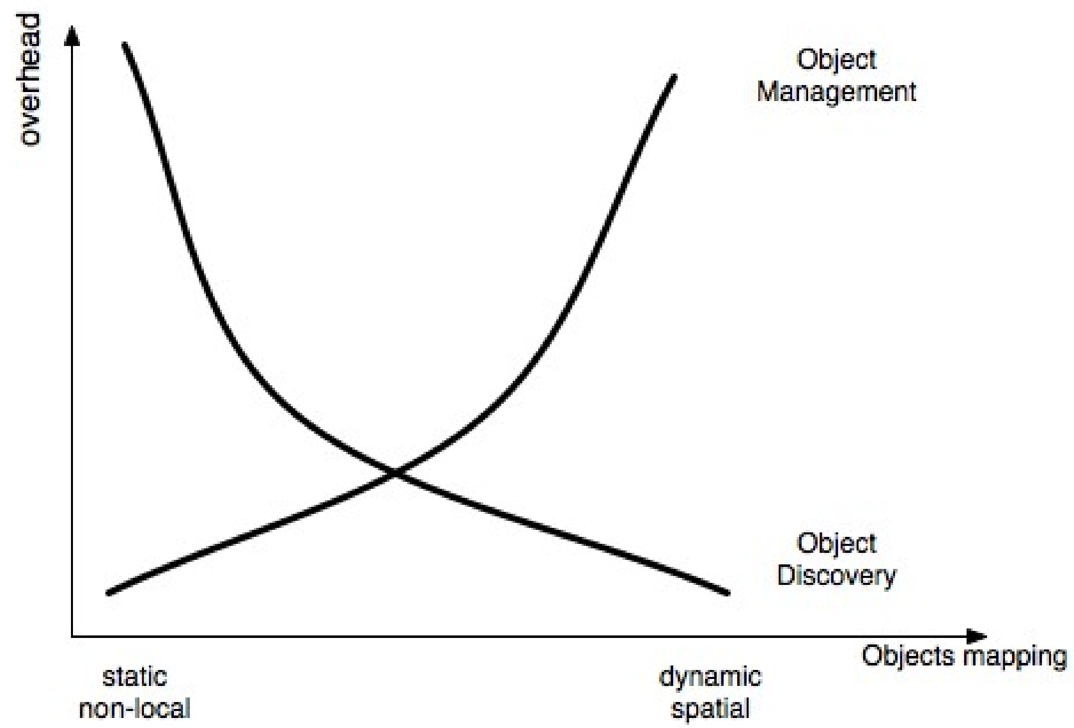
- Aim: find the “closest” node for the set of peer connected to a VN
  - NCS coordinates to map the distances between peers and servers
  - Overlay Network
    - Delauney
  - Leader Election to find the best server for a set of peers



Interest Management

Interest Management

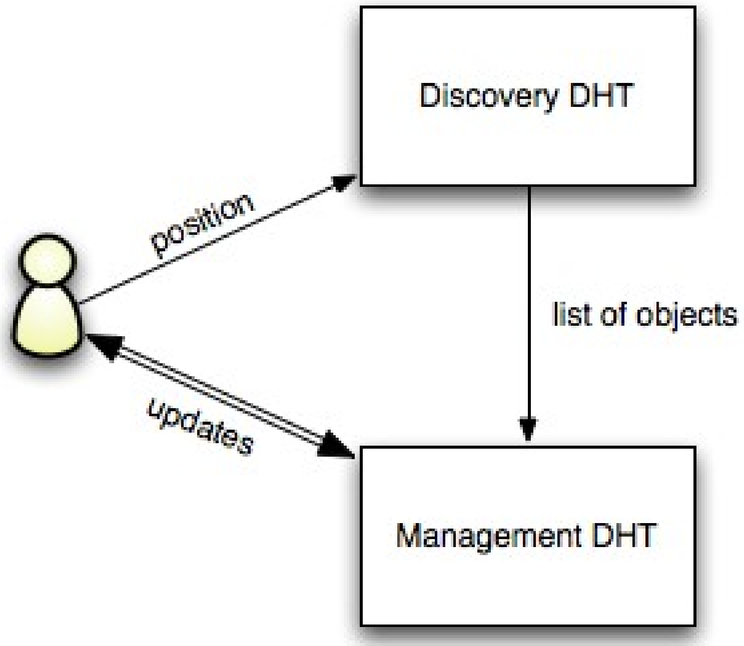
- Two phases
  - Objects discovery
  - Objects management
- Two different *specialized* structures for the discovery and management.
- [Abdallah et al. 2008]
  - We use two DHTs



Assumption: moving objects

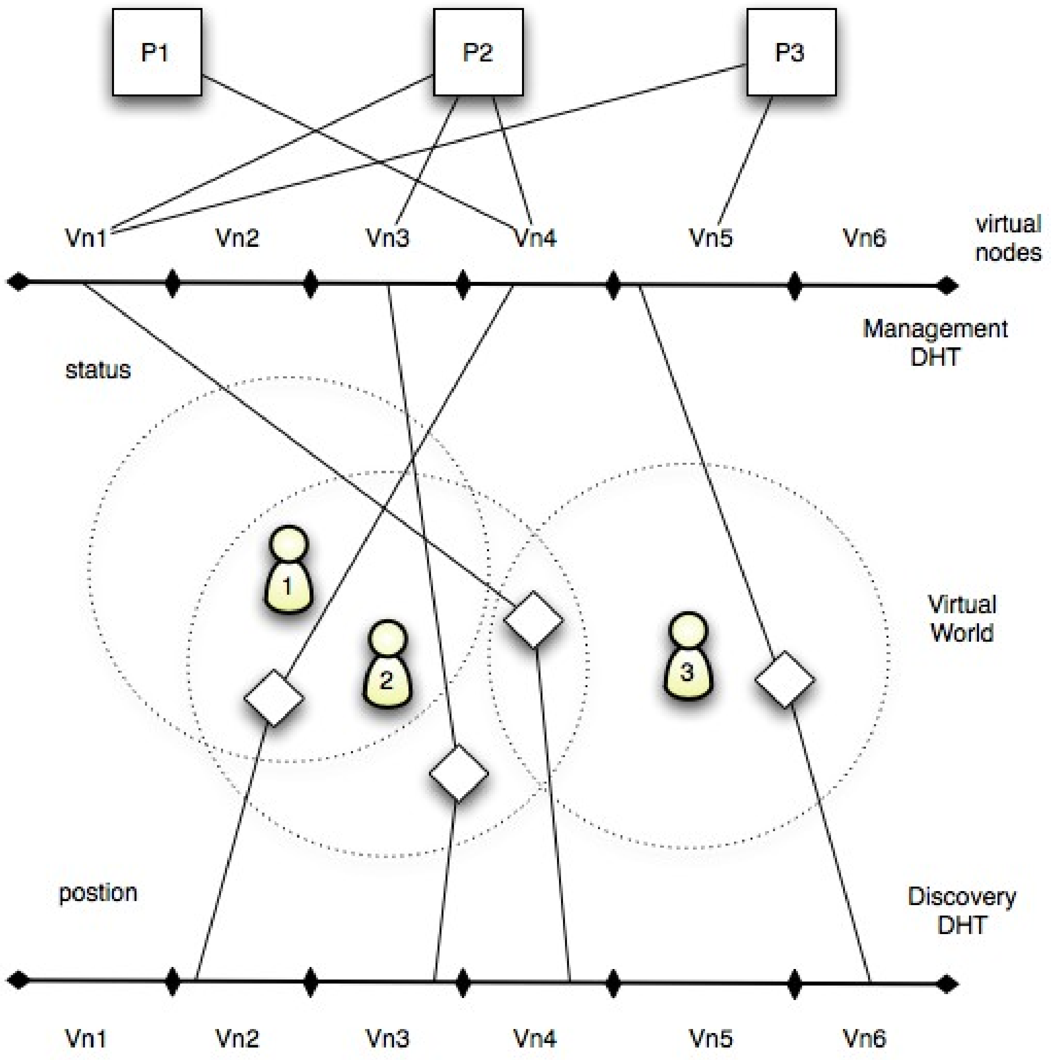
Interest Management with two DHTs

- IM goes through two DHTs
  - Discovery: spatial mapping
  - Management: static mapping
- Main issue: latency
  - Prefetch – not THE solution
- Locality preserving hash
  - Range queries
- One hop DHTs
- Queries in parallel



Interest Management with two DHTs

- Random mapping in management DHT
- Spatial mapping in discovery DHT

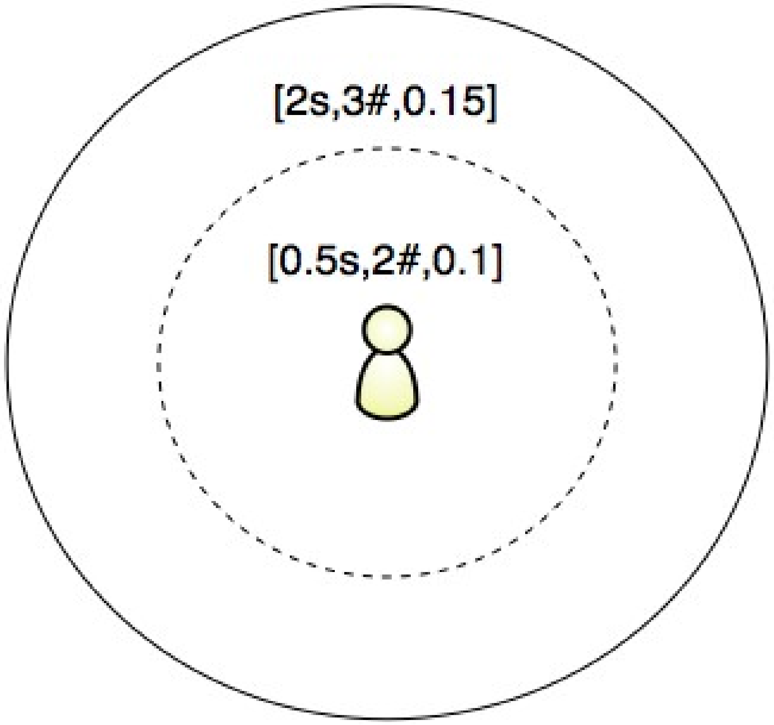


Consistency Model

Consistency Model

- Aim: tuning the bandwidth requirements when managing replicas
- VFC: optimistic consistency model developed for mobile games
  - [Santos et al. 2007]
- 3D vector: 3 consistency view
  - Time – delay
  - Sequence – number of operation
  - Value – magnitude of change

- Extendible & Adaptable
  - Adding/removing views according to the context
- General enough to be used:
  - Among VNs
  - Between bVN and uVN
  - Between server and client



Conclusions

- The presented ideas can result in
  - an MMVE environment
    - Exploit different kind of HW resources
    - Focused on bandwidth saving
- Our current work includes various tools and results
  - From other research groups in various fields
    - VNs, VFC, etc...
  - That we are still studying, at different stages of investigation
    - bVN, locality-aware server mapping

Conclusions

- From our group (already studied and implemented)
  - One-Hop DHTs for request with temporal locality
  - Bandwidth tuning when disseminating data in DHTs
  - Locality hash functions

E. Carlini, M. Coppola, D. Laforenza and L. Ricci, Reducing Traffic in DHT-based Discovery Protocols for Dynamic Resources, CoreGrid ERCIM Working Group Workshop on Grids, P2P and Service Computing at EuroPAR , 2009

E. Carlini, M. Coppola, L. Ricci, "Modelling PUSH-PULL Data Dissemination over Distributed Hash Tables" (under submission)

Thank you!
Questions?

Contrail: Open Computing Infrastructures for Elastic Services is funded by the EU under the Seventh Framework Programme Project No. 257438, 36 months span starting October 2010

The XtreemOS project (June 2006-September 2010) was funded by the European Commission under contract IST-FP6-033576

Extra Slides