

HIVEK: the HIVE Project

Distributed VR Runtime Kernel

Chris Greenhalgh

University of Nottingham
(School of Computer Science and
Information Technology)

Contents

- Features and Characteristics
- Agents and Environments
- Environment Services
- Environment Realisation
- Locales
- Other Features
- Current Use and Limitations

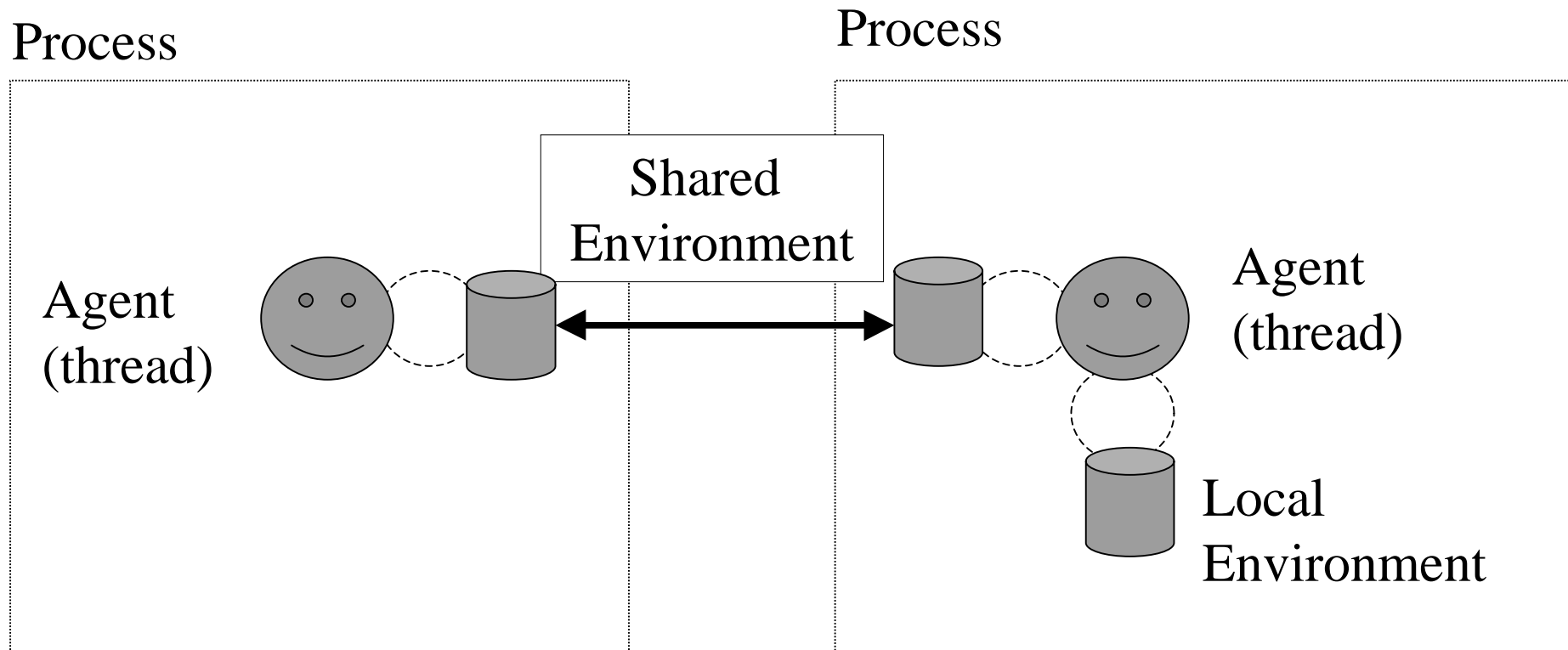
Features and Characteristics

- As a Library plus example applications.
 - Enhancement of applications on-going.
- Currently runs on SGs (NT soon), using elements of CRGSHARE (like MASSIVE-2).
- Provides CVE shared data service, implementing consistency ideas from Reading University.
- Provides virtual world structuring through Locales (originally in MERL Spline).

Agents and Environments

- Agent = process/thread (active).
- Environment = shared data partition (passive).
 - Agents create, share, and maintain Environments (providing CPU time).
 - An agent can join many environments; an environments can be joined by many agents.
 - Each environment is fully replicated.

Agents and Environments



Environment Services

- An environment contains a hierarchy of data items (i.e. a scene graph, plus annotations).
- An agent can add, update, or delete an item.
- Only an item's owner can update or delete it.
- Ownership is transferable (but can be fixed).
- Local operations are performed immediately.
- Simple clock synchronisation is used between replicas of an environment (e.g. for trajectories).

Environment Data Types

- Entity = 3D transformation.
- Geometry = 3D geometry (filename at present).
- Attribute = text name/value pair.
- Trajectory = time-varying update to Entity.
- Boundary = connection between Locales.
- Link = reference to another sub-tree.
- UpdateRequest = request to change locked item.
- Behaviour = code - not yet defined.

Environment Consistency

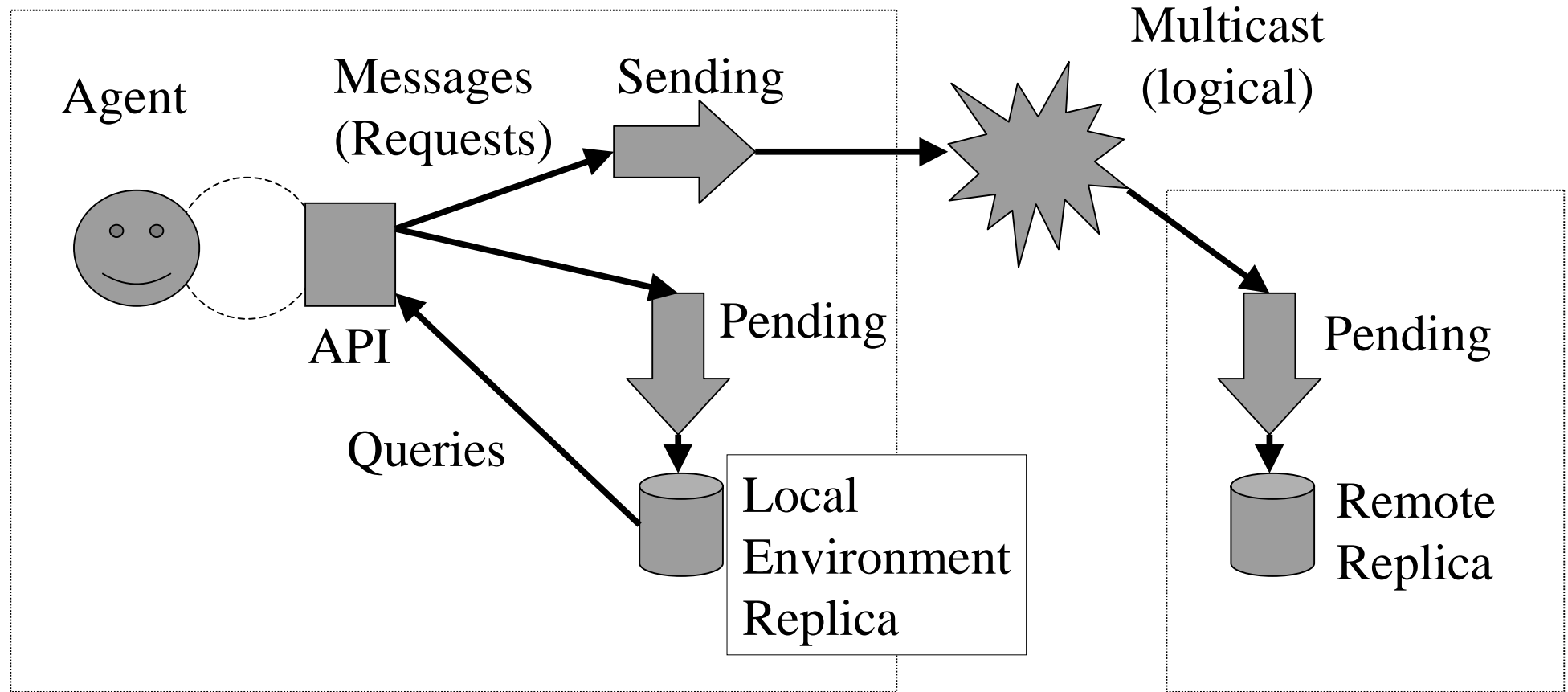
- Operations on a single item are totally ordered (lock transfer includes sequencer transfer).
- Additional causal (ordering) constraints between operations can be specified explicitly.
- Operations can have a minimum delivery time (allows advance transmission of operations).
- Updates can be reliable or unreliable (selected per update).

Environment Realisation

- Using environment API creates messages representing requests (add, update, delete).
- Messages are copied to environment's local "sending" and "pending" queues.
- Agent periodically flushes sending and pending queues.
- Pending queue enforces causal constraints and minimum delivery time before updating the local copy of the environment.

Environment Realisation

Local Process



Environment Issues

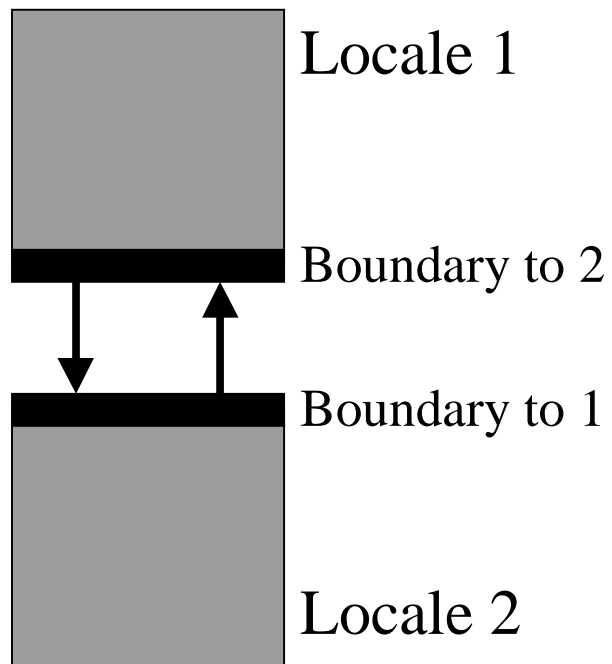
- Explicit sending and pending queues allow insertion of additional message handling, e.g.:
 - alternative ordering constraints.
 - elimination of “old” messages.
 - summarisation or compression of message sequences.
- Can expose information for flow control.
- Synchronised with, and observed by, agent.

Locales

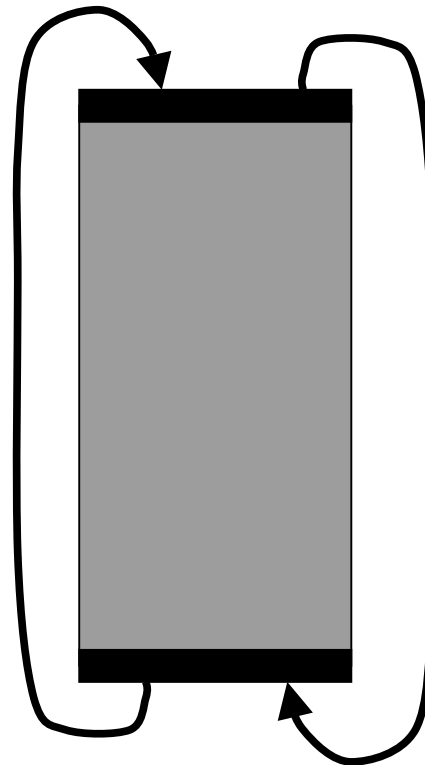
- A Locales is a region of virtual space, which defines its own coordinate system.
- Each environment (currently) forms a Locale.
- Worlds are constructed by linking together Locales using boundaries.
- Supports scalability through world partitioning.
- Supports heterogeneity through flexible replication policies.

Locales examples

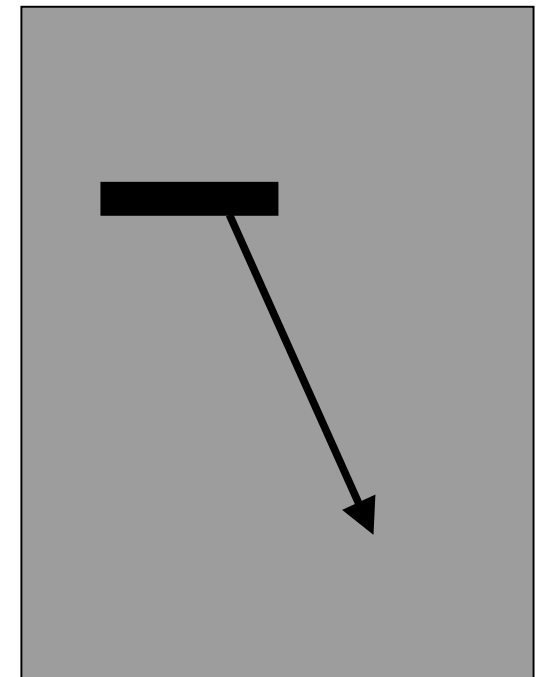
Simple adjoining
“rooms”



Infinite corridor



Teleport



Other Features

- Has a standard Trader/name service to map:
 - text name to environment id.
 - environment id to controlling agent (IP address).
- API supports “waiting for ownership” mode:
 - accumulates messages in separate queue pending ownership transfer.
- API supports “controlled” locking and updates:
 - transforms update messages into UpdateRequests within the environment: owner can see and respond.

Current Use and Limitations

- HIVE project: ownership prediction, event prediction, and advance communication.
- Jim/EPSRC: support for persistence in CVEs, year 1 prototype and experiment.
- Mina: scalable audio, year 1 prototype.
- Limitations:
 - Currently TCP communication only (add m'cast).
 - Primitive user client.
 - Application programming model undeveloped.