

Managed by



A parallel and distributed abstract model (The MxN meta-model of PaCO++ and GridCCM)

Christian Perez Joint work with André Ribes & Thierry Priol

INRIA Rennes Christian.Perez@irisa.fr











- Context
- Expected behavior
- A Parallel Distributed (Abstract) Model
- Projecting the model
 - GridCCM
 - (Parallel) Fractal
- Conclusion





Application in hydrogeology: Saltwater intrusion

Excess pumpina

of fresh

water

Fresh

groundwater aquifer

lowered water tabl

Coupled physical models One model = one software Saltwater intrusion

- flow / transport
- **Reactive transport**
 - transport / chemistry

Hydrogrid project, supported by the French ACI-GRID



flow : velocity and pressure function of the density Density function of salt concentration Salt transport : by convection (velocity) and diffusion

Well

contaminated

with salt water

salt water

intrusion

original

water table

Salt Water Intrusion in Coastal Areas

orignal salt

water

interface

C. Ophardt c.1997

Salt

Water







European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GRID and Peer-to-Peer Technologies



Requested Features

Support of legacy codes – Multi-languages, multi-OS, multi-processors Support of parallel codes – MPI, PVM, OpenMP, ... Support for network transparency Support for deployment

- "Single-provider" application
- "Multi-provider" application

Software component!

European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GR D and Peer-to-Peer Technologies

Flow

ranspo





Distributed components: OMG CCM

A distributed component-oriented model

- An architecture for defining components and their interaction
- Interaction implemented through input/output interfaces
- Synchronous and asynchronous communication
- A runtime environment based on the notion of containers (lifecycle, security, transaction, persistent, events)
- Multi-languages/OS/ORBs/vendors, ...
- With a packaging/deployment model
 - A packaging technology for deploying binary multi-lingual executables
 - Could be deployed and run on several distributed nodes simultaneously



Component-based application

Courtesy of P. Merle, LIFL, France





Component and Parallelism

How to embed a parallel code into a component?







Parallel Component

What the application designer should see...



... and how it must be implemented !





Managed by



Behavior of a parallel distributed model











Distributed Component Model

// Caller Code o->factorize(m);

// Receiver Code void serv::factorize(const Matrix mat) { ...;}

Receiver

Distributed component (a process)

Caller





SPMD Parallel Component Model (1)







SPMD Parallel Component Model (2)







SPMD Parallel Component Model (3)









A Parallel Distributed (Abstract) Model











Why an abstract model?

Many distributed abstractions – Objet, composant, service, ...

Many middleware systems and languages – CORBA, JAVA RMI, .NET, WS, ...





Content of the abstract model

Four models

- Definition of a parallel distributed entity
- Connection and communication
- Distributed data management (in a communication)
- Exception management (in a communication)





An entity

Representation of a "program"

- Either an object, a component, a service, ...



An entity





A distributed entity

Offer a distributed reference

- May provide services
- May use services



A distributed entity





A Parallel Entity

- Is made of a collection of entity
- Share a state
- Can communicate internally



A parallel entity





A parallel distributed entity (1)

Straightforward merge

Parallel distributed entity

- Each element of the parallel entity is a distributed entity
- Offer several (distributed) references

Notion of abstract reference is missing

- Choice
- Added later

Learnt from the past

- Functional code
- Non-functional code







A parallel distributed entity (2)

Non functional part



Global state







Connections and communications : objectives

Parallel communications

- Retrieval of functional node references

Notion of « abstract » reference

 A parallel distributed entity should be manipulated as a distributed entity



European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GRID and Peer-to-Peer Technologies





Specialization of a parallel distributed entity

Adding connection managers

- In-Proxy : a unique connection point
- Out-Proxy : handle out connection



European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GRID and Peer-to-Peer Technologies

















Data redistribution

Three actors

- User
- Redistribution library
- Middleware system

Objective : define relationships







User – Redistribution library

- Configuration of the library with respect to data
 Depend on the data type and/or the library
- > Out of the scope of the parallel distributed model







User – Middleware System

- Data transfer
- Find the redistribution library/ies
- Implementation dependent
 - Out of the scope of this model







Middleware System – redistribution library

- Do the data redistribution within a communication
- Definition of a (abstract) interface
 - Model the step of communication
 - Both caller and receiver sides







Projection of the model

Objective : concrete programming model

- Object, component, service, peer, ...

Straightforward projection

- E.g. : a distributed entity == an object
- Follow the model

Already done:

- PaCO++ : Parallel CORBA Object Model
- GridCCM : Parallel CORBA Component Model

Other possible projections:

- Parallel Fractal, Parallel Web Services, Parallel JXTA, ...





Example of projection: GridCCM

Extends the CORBA Component Model

- No modification of CCM specification
- Portable
- Extension of CCM descriptor (XML)

Parallel component

- Parallelism is a non-functional property of a component
 - It is an implementation issue
- Collection of sequential component
 - SPMD execution model
- Support of distributed arguments
 - API for data redistribution
 - 1D bloc-cyclic, Identity, RedSym
 - API for communication scheduling w.r.t. network properties
- Support of parallel exceptions





RedSym: Parallel viz with VTK (EPSN/RedGrid Project)





GridCCM Component Example (External)



European Research Network on Foundations, Software Infrastructures and Applications for large scale













European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GRID and Peer-to-Peer Technologies





(Parallel) Fractal : using specialized controllers



European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GRID and Peer-to-Peer Technologies





Conclusion

Parallel distributed (abstract) model Notion of parallel distributed entity

- Foundation for building other models
 Dynamic and parallel connection
 Distributed data handling
- Unbounded number of redistribution libraries
 Not presented:
 - Communication Scheduling
 - Parallel Exception Handling

Status:

- PaCO++, available, http://www.irisa.fr/paris/Padico
- GridCCM in development
- Post-doc integrating it into Salome, CEA/EDF
- Should be "integrated" in ASSIST

Contribution to WP3

"NxM Parallel feature" for GCM