DIET: New Developments and Recent Results

A. Amar[1], R. Bolze[1], A. Bouteiller[1], A. Chis[1],
Y. Caniou[1], E. Caron[1], P.K. Chouhan[1], G.L. Mahec[2],
H. Dail[1], B. Depardon[1], F. Desprez[1], J. S. Gay[1], A. Su[1]

LIP Laboratory (UMR CNRS, ENS Lyon, INRIA, UCBL 5668) / GRAAL Project

LPC / PCSV (CNRS / IN2P3 UBP Clermont-Ferrand)

29 August 2006
CoreGrid Workshop

Outline

1. Introduction
2. Distributed Interactive Engineering Toolbox
3. DIET Applications
4. Discussion

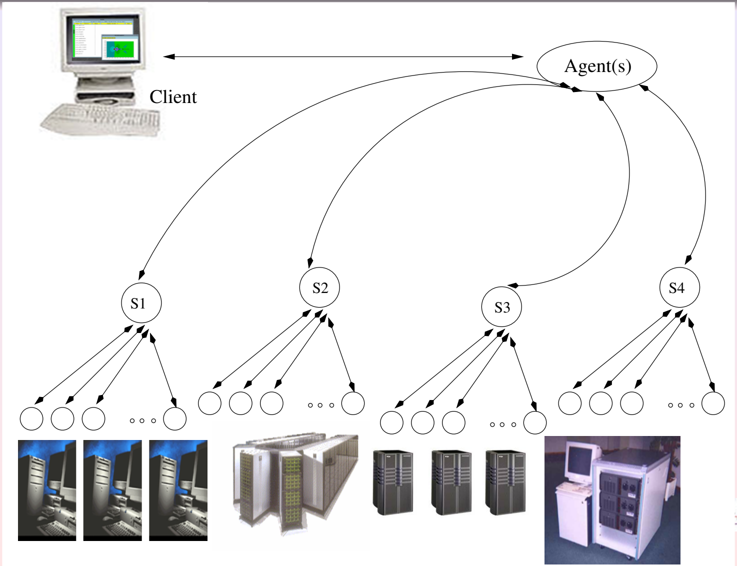
Outline

1. Introduction
2. Distributed Interactive Engineering Toolbox
3. DIET Applications
4. Discussion

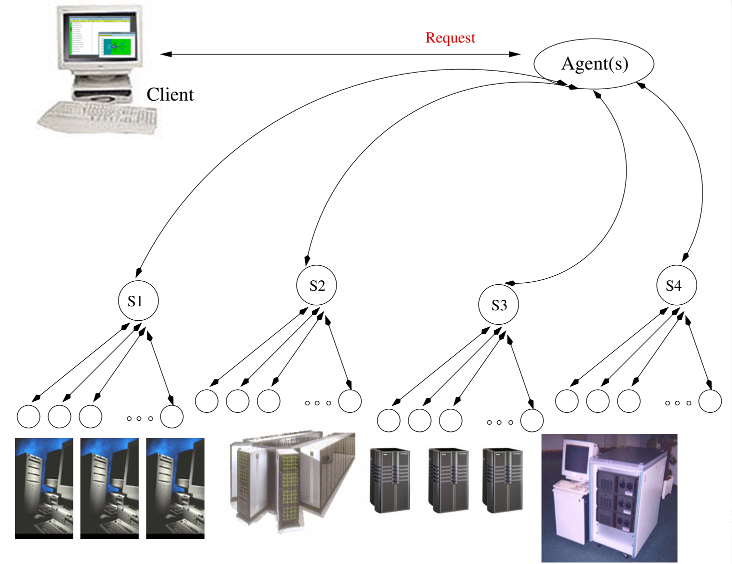
Why we need tools like DIET?

- To access large computational power and/or storage capacity through the Internet.
- Large scale computing over heterogeneous platforms was coined by Ian Foster in the mid-90's - Grid.
- Implementation of RPC programming model over the Grid - GridRPC.
- Existing implementations: NetSolve, Ninf, DIET, XtremWeb, OmniRPC, · · ·

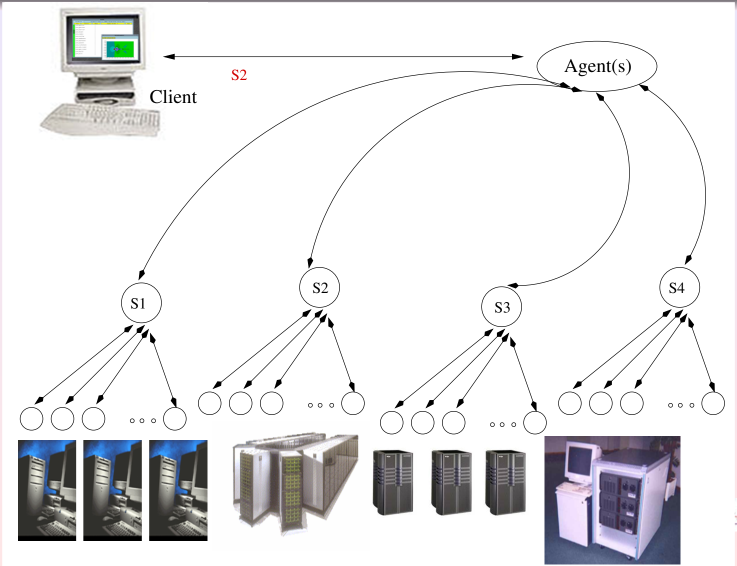
GridRPC Paradigm



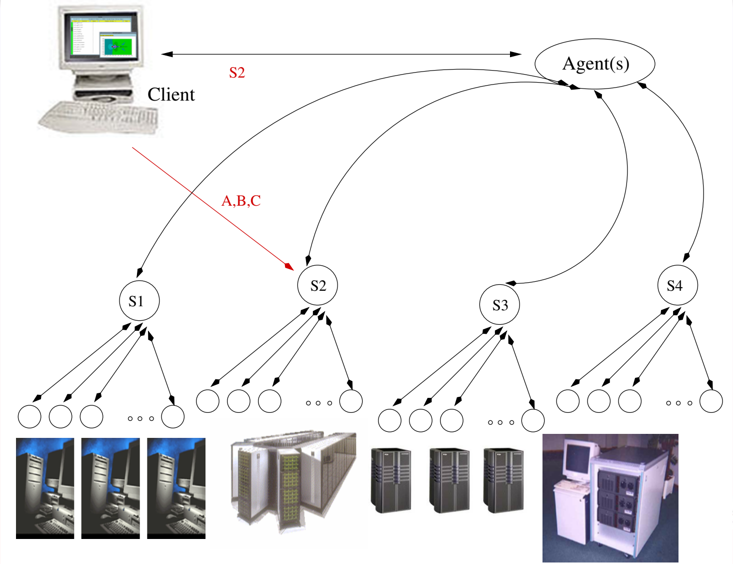
GridRPC Paradigm



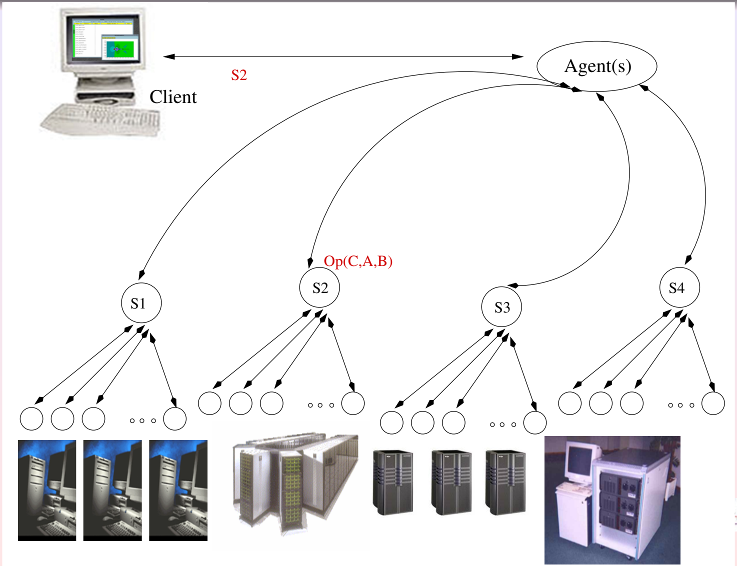
GridRPC Paradigm



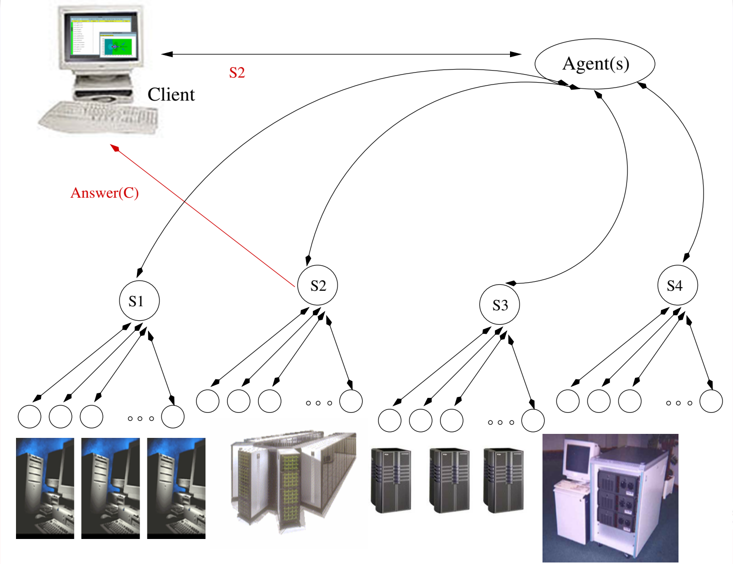
GridRPC Paradigm



GridRPC Paradigm



GridRPC Paradigm

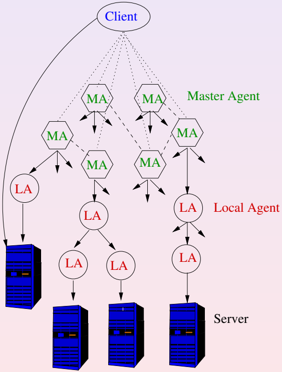


Why DIET?

Hierarchical architecture for an improved scalability.

Why DIET?

Hierarchical architecture for an improved scalability.



Outline

1 Introduction

2 Distributed Interactive Engineering Toolbox
- DIET Scheduling
- DIET Deployment
- Fault Tolerance in DIET
- DIET Visualization

3 DIET Applications

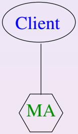
4 Discussion

DIET Components

- Client : An application that uses DIET to solve problems
- Master Agent (MA) :
  - Receives requests from clients
  - Collects computational abilities from servers and selects
  - Returns the reference of the chosen server to the client
- Local Agents (LA) :
  - Act as transmitter
  - Share the workload of scheduling
- Servers (SeD) : Perform actual computation for client

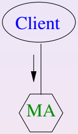
DIET Components

- Client : An application that uses DIET to solve problems
- Master Agent (MA) :
  - Receives requests from clients
  - Collects computational abilities from servers and selects
  - Returns the reference of the chosen server to the client
- Local Agents (LA) :
  - Act as transmitter
  - Share the workload of scheduling
- Servers (SeD) : Perform actual computation for client



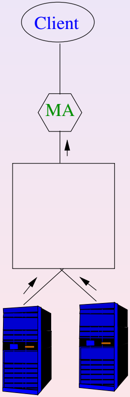
DIET Components

- Client : An application that uses DIET to solve problems
- Master Agent (MA) :
  - Receives requests from clients
  - Collects computational abilities from servers and selects
  - Returns the reference of the chosen server to the client
- Local Agents (LA) :
  - Act as transmitter
  - Share the workload of scheduling
- Servers (SeD) : Perform actual computation for client



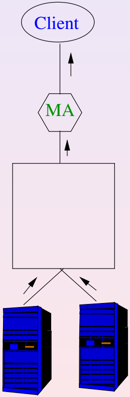
DIET Components

- Client : An application that uses DIET to solve problems
- Master Agent (MA) :
  - Receives requests from clients
  - Collects computational abilities from servers and selects
  - Returns the reference of the chosen server to the client
- Local Agents (LA) :
  - Act as transmitter
  - Share the workload of scheduling
- Servers (SeD) : Perform actual computation for client



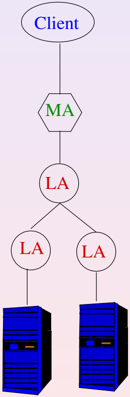
DIET Components

- Client : An application that uses DIET to solve problems
- Master Agent (MA) :
  - Receives requests from clients
  - Collects computational abilities from servers and selects
  - Returns the reference of the chosen server to the client
- Local Agents (LA) :
  - Act as transmitter
  - Share the workload of scheduling
- Servers (SeD) : Perform actual computation for client



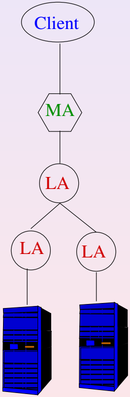
DIET Components

- Client : An application that uses DIET to solve problems
- Master Agent (MA) :
  - Receives requests from clients
  - Collects computational abilities from servers and selects
  - Returns the reference of the chosen server to the client
- Local Agents (LA) :
  - Act as transmitter
  - Share the workload of scheduling
- Servers (SeD) : Perform actual computation for client



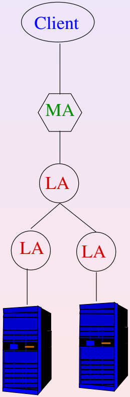
DIET Components

- Client : An application that uses DIET to solve problems
- Master Agent (MA) :
  - Receives requests from clients
  - Collects computational abilities from servers and selects
  - Returns the reference of the chosen server to the client
- Local Agents (LA) :
  - Act as transmitter
  - Share the workload of scheduling
- Servers (SeD) : Perform actual computation for client



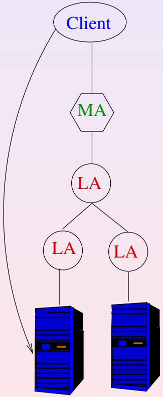
DIET Components

- Client : An application that uses DIET to solve problems
- Master Agent (MA) :
  - Receives requests from clients
  - Collects computational abilities from servers and selects
  - Returns the reference of the the chosen server to the client
- Local Agents (LA) :
  - Act as transmitter
  - Share the workload of scheduling
- Servers (SeD) : Perform actual computation for client



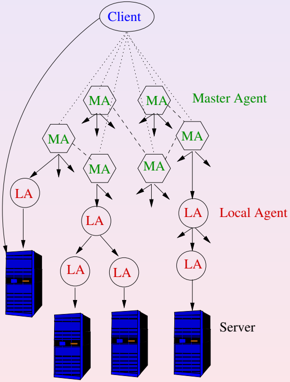
DIET Components

- Client : An application that uses DIET to solve problems
- Master Agent (MA) :
  - Receives requests from clients
  - Collects computational abilities from servers and selects
  - Returns the reference of the chosen server to the client
- Local Agents (LA) :
  - Act as transmitter
  - Share the workload of scheduling
- Servers (SeD) : Perform actual computation for client



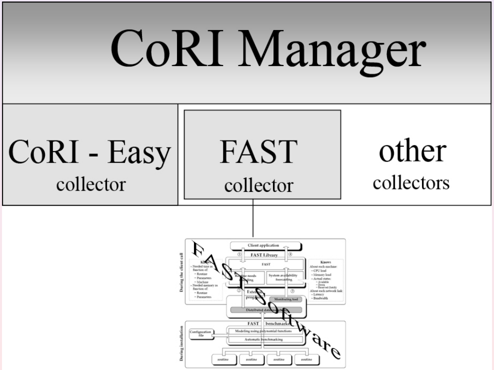
DIET Components

- Client : An application that uses DIET to solve problems
- Master Agent (MA) :
  - Receives requests from clients
  - Collects computational abilities from servers and selects
  - Returns the reference of the chosen server to the client
- Local Agents (LA) :
  - Act as transmitter
  - Share the workload of scheduling
- Servers (SeD) : Perform actual computation for client



Collector of Resource Information (CoRI)

CoRI-Easy: provides basic measurements of the environment
CoRI Manager: manages the use of different collectors



Plug-in schedulers

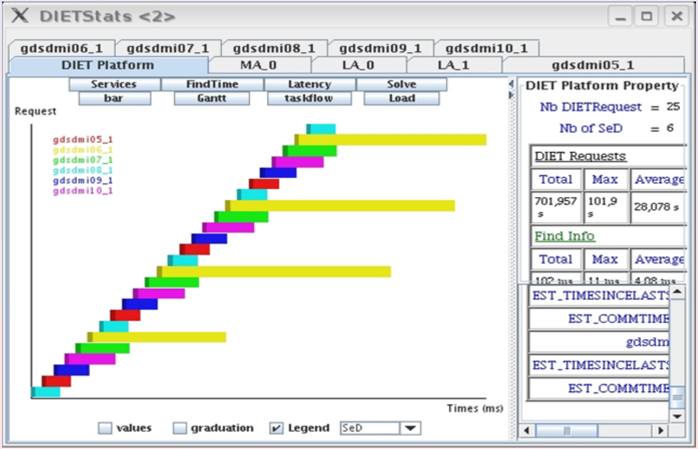
- Applications vary in terms of performance demands
- Performance Estimation Vector (PEV) is used for scheduling
- SeDs sends PEV to MA as a response to a request

`Information tag starts with EST_`	`Explanation`
TCOMP	the predicted time to solve a problem
LOADAVG	CPU load average
FREEMEM	amount of free memory (Mb)
NBCPU	number of available processors
⋮	⋮

Verification of plug-in schedulers

- Six heterogeneous servers
- Sequential and independent requests
- Inter-arrival time for the request is 1 minute
- Compare two scheduler
  - Round Robin scheduler
  - CPU scheduler

Round Robin Scheduler



DIET Platform Property

Nb DIETRequest = 25

Nb of SeD = 6

DIET Requests		
Total	Max	Average
701,957 s	101,9 s	28,078 s

Find Info		
Total	Max	Average
102 ms	11 ms	4,08 ms

EST_TIMESINCELASTS

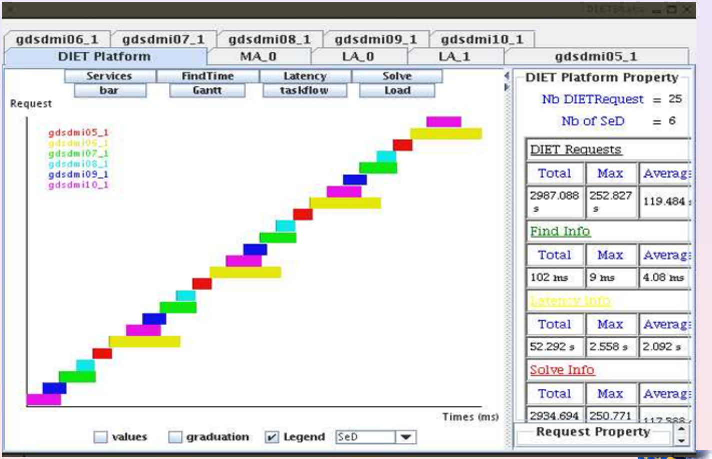
EST_COMMTIME

gdsdm

EST_TIMESINCELASTS

EST_COMMTIME

CPU Scheduler



DIET Platform Property

Nb DIETRequest = 25

Nb of SeD = 6

DIET Requests

Total	Max	Averag
2987.088 s	252.827 s	119.484 s

Find Info

Total	Max	Averag
102 ms	9 ms	4.08 ms

Latency Info

Total	Max	Averag
52.292 s	2.558 s	2.092 s

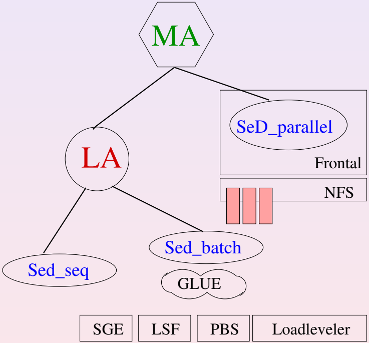
Solve Info

Total	Max	Averag
2934.694	250.771	117.388

Request Property

Batch Scheduler Management

- System dependent
  - NFS: copy the code?
  - MPI: LAM, MPICH?
- Batch system dependent
  - No homogeneity
  - Scheduler behaviour
  - Information about the internal scheduling process
- Monitoring and performance prediction

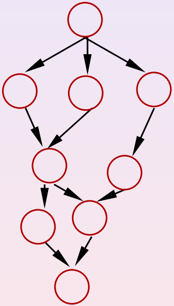


Parallel and batch submissions

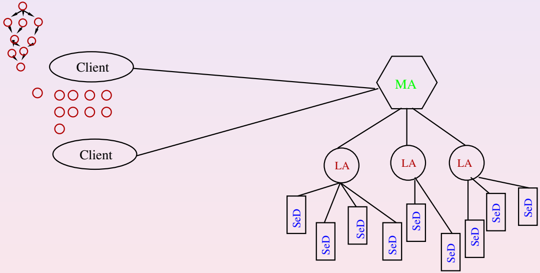
- Performance prediction: SIMBATCH
  - Batch, cluster, parallel tasks simulator plugin for SimGrid
  - Goal: test new grid schedulers more realistically and performance prediction
  - Implement FIFO, Conservative BackFilling, $\cdots$
- More problems
  - Asynchronous, long term production jobs
  - Performance prediction
    - How to decide number of processors for application?
    - If reservation available, how to compute deadline?
  - Co-scheduling?
  - Data and job migration?

Workflow Management

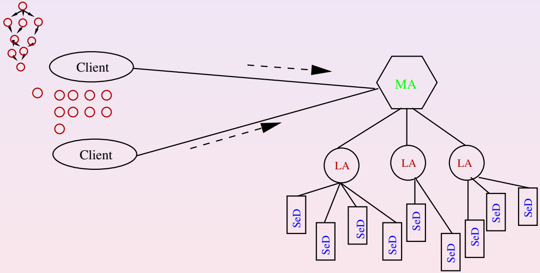
- Workflow is represented as DAG
- Use different heuristic methods to solve scheduling problems
- Extensibility to address multi-workflows submission and large grid platform
- Manage heterogeneity and variability of environment



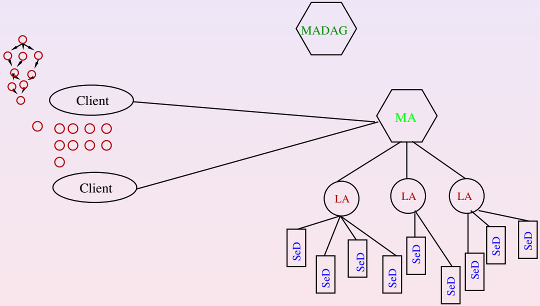
DIET workflow engine



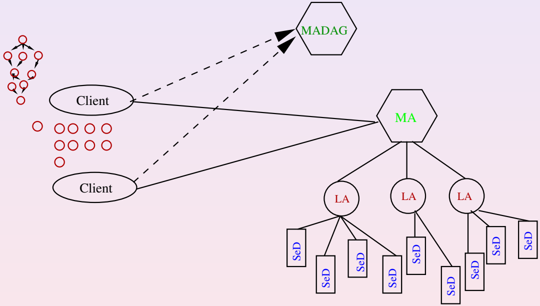
DIET workflow engine



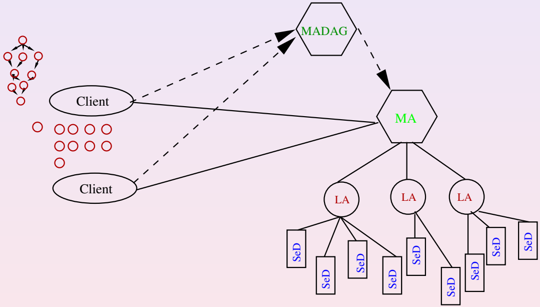
DIET workflow engine



DIET workflow engine



DIET workflow engine



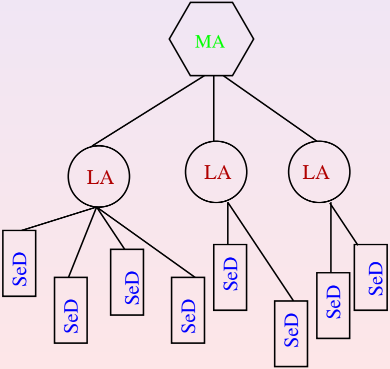
DIET Deployment

Mapping of DIET's components on available resources.

DIET Deployment

Mapping of DIET's components on available resources.

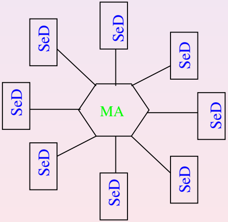
- Homogeneous resources



DIET Deployment

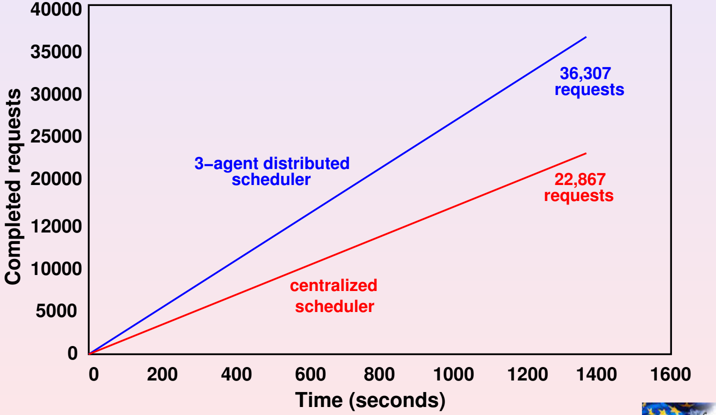
Mapping of DIET's components on available resources.

- Homogeneous resources



DIET Deployment

Mapping of DIET's components on available resources.



DIET Deployment

Mapping of DIET's components on available resources.

- Homogeneous resources

Complete d-ary Spanning tree

E.Caron, P.K.Chouhan,H.Dail and F.Vivien. Automatic Middleware Deployment Planning for Clusters. RR-2000-50 LIP,October 2005.

DIET Deployment

Mapping of DIET's components on available resources.

- Heterogeneous resources

DIET Deployment

Mapping of DIET's components on available resources.

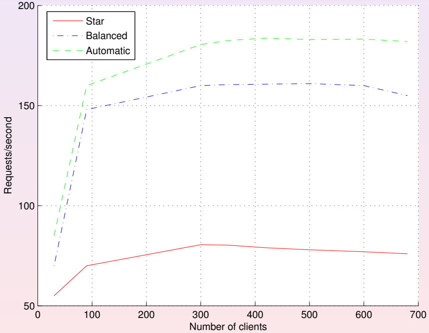
- Heterogeneous resources

Find the best broadcast tree on a general graph, which is known to be NP-complete.

DIET Deployment

Mapping of DIET's components on available resources.

- Heterogeneous resources



Thesis Automatic Deployment for Application Service Provider Environments by P. K. Chouhan

GoDIET: DIET deployment tool

- XML file as Input
- Generate configuration files
- Launches services (name service, logging services)
- Launches DIET elements
- Destroy deployed platform
- Remote cleanup of launched processes

GoDIET: DIET deployment tool

- XML file as Input
- Generate configuration files
- Launches services (name service, logging services)
- Launches DIET elements
- Destroy deployed platform
- Remote cleanup of launched processes

GoDIET: DIET deployment tool

- XML file as Input
- Generate configuration files
- Launches services (name service, logging services)
- Launches DIET elements
- Destroy deployed platform
- Remote cleanup of launched processes

GoDIET: DIET deployment tool

- XML file as Input
- Generate configuration files
- Launches services (name service, logging services)
- Launches DIET elements
- Destroy deployed platform
- Remote cleanup of launched processes

GoDIET: DIET deployment tool

- XML file as Input
- Generate configuration files
- Launches services (name service, logging services)
- Launches DIET elements
- Destroy deployed platform
- Remote cleanup of launched processes

GoDIET: DIET deployment tool

- XML file as Input
- Generate configuration files
- Launches services (name service, logging services)
- Launches DIET elements
- Destroy deployed platform
- Remote cleanup of launched processes

GoDIET: DIET deployment tool

- XML file as Input
- Generate configuration files
- Launches services (name service, logging services)
- Launches DIET elements
- Destroy deployed platform
- Remote cleanup of launched processes

E.Caron, P.K.Chouhan and H.Dail GoDIET: A Deployment Tool for Distributed Middleware on Grid'5000. In IEEE, editor, EXPGRID workshop, in conjunction with HPDC-15. pages 1-8.

Failure Detection in DIET

Failure detection

- Detection time: time between failure and definitive suspicion
- Accuracy: probability of the observer to be true at random time

Failure Detection in DIET

Failure detection

- Detection time: time between failure and definitive suspicion
- Accuracy: probability of the observer to be true at random time

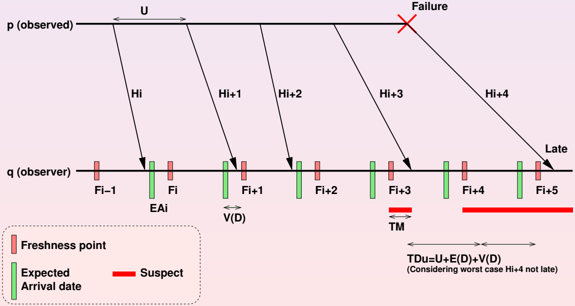
Chandra, Toueg and Aguilera Failure Detector

Failure Detection in DIET

Failure detection

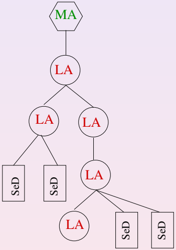
- Detection time: time between failure and definitive suspicion
- Accuracy: probability of the observer to be true at random time

Chandra, Toueg and Aguilera Failure Detector



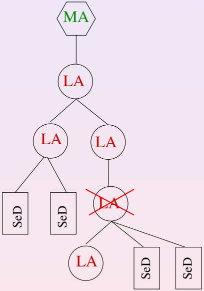
Architecture Recovery

- Problem: Keep architecture connected
- Solution: Keep ancestors list
- tolerates up to f-1 simultaneous failures



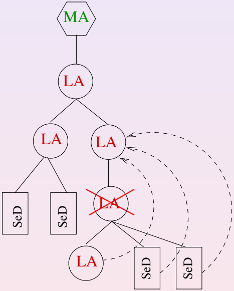
Architecture Recovery

- Problem: Keep architecture connected
- Solution: Keep ancestors list
- tolerates up to f-1 simultaneous failures



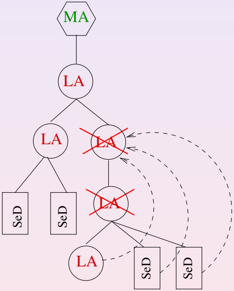
Architecture Recovery

- Problem: Keep architecture connected
- Solution: Keep ancestors list
- tolerates up to f-1 simultaneous failures



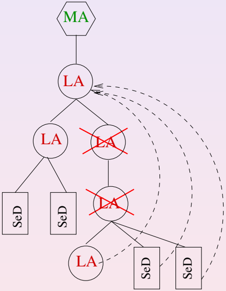
Architecture Recovery

- Problem: Keep architecture connected
- Solution: Keep ancestors list
- tolerates up to f-1 simultaneous failures



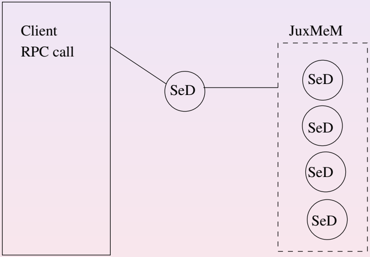
Architecture Recovery

- Problem: Keep architecture connected
- Solution: Keep ancestors list
- tolerates up to f-1 simultaneous failures



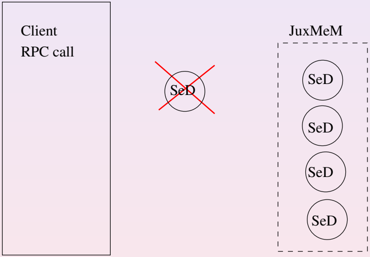
Application Recovery

- Replication and checkpointing
- Replication divides the available computing by f
- Checkpointing takes periodic snapshot of process state, saving it to another place
- JuxMem manages data persistence across failures by replicating it on nodes



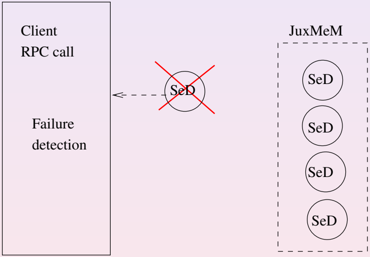
Application Recovery

- Replication and checkpointing
- Replication divides the available computing by f
- Checkpointing takes periodic snapshot of process state, saving it to another place
- JuxMem manages data persistence across failures by replicating it on nodes



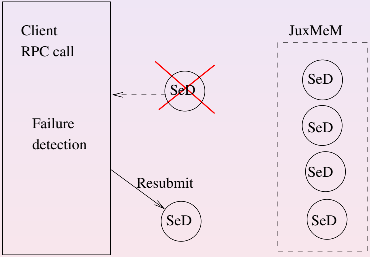
Application Recovery

- Replication and checkpointing
- Replication divides the available computing by f
- Checkpointing takes periodic snapshot of process state, saving it to another place
- JuxMem manages data persistence across failures by replicating it on nodes



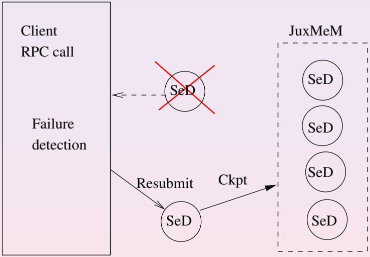
Application Recovery

- Replication and checkpointing
- Replication divides the available computing by f
- Checkpointing takes periodic snapshot of process state, saving it to another place
- JuxMem manages data persistence across failures by replicating it on nodes



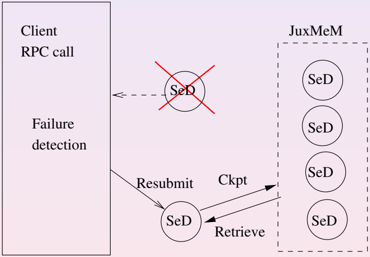
Application Recovery

- Replication and checkpointing
- Replication divides the available computing by f
- Checkpointing takes periodic snapshot of process state, saving it to another place
- JuxMem manages data persistence across failures by replicating it on nodes



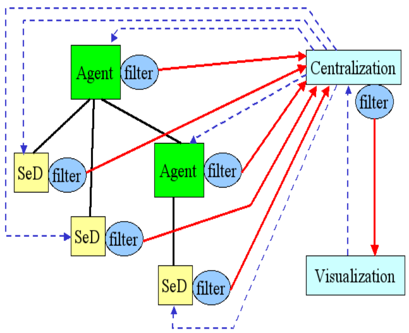
Application Recovery

- Replication and checkpointing
- Replication divides the available computing by f
- Checkpointing takes periodic snapshot of process state, saving it to another place
- JuxMem manages data persistence across failures by replicating it on nodes



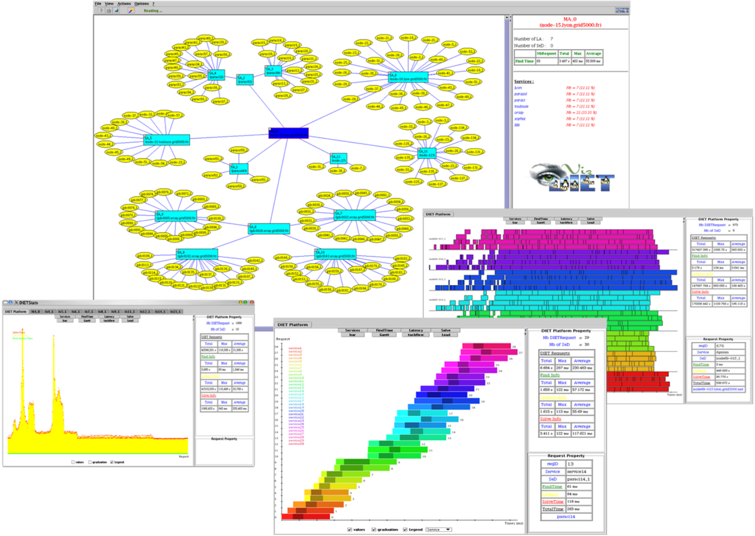
VizDIET: DIET Visualization Tool

- Current view of the DIET platform
- Communication between agents
- State of SeDs
- Scalability
- Available services
- Data persistency
- Name information
- CPU, memory and network load



R.Bolze, E.Caron, F.Desprez, G.Hoesch, and C.Pontvieux. A Monitoring and Visualization Tool and Its Applications. Computational Science and Its Applications-ICCSA 2006, volume 3984 of LNCS, pages 202-213.

Screen-shot of VizDIET



MA_0
(node-15.lyon.grid5000.fr)

Number of LA : 7
Number of SeD : 0

	NbRequest	Total	Max	Average
Find Time	63	3.497 s	452 ms	55.508 ms

Services :

lyon	Nb = 7 (11.11 %)
parasol	Nb = 7 (11.11 %)
paraci	Nb = 7 (11.11 %)
toulouse	Nb = 7 (11.11 %)
orsay	Nb = 21 (33.33 %)
sophia	Nb = 7 (11.11 %)
lille	Nb = 7 (11.11 %)

DIET Platform Property
Nb DIETRequest = 1999
Nb of SeD = 10

DIET Requests		
Total	Max	Average
42500,301 s	116,035 s	21,305 s
Find Info		
Total	Max	Average
3,695 s	80 ms	1,848 ms
Latency Info		
Total	Max	Average
41510,203 s	115,406 s	20,769 s
Solve Info		
Total	Max	Average
1066,420 s	843 ms	533,403 ms

Request Property

DIET Platform Property
Nb DIETRequest = 29
Nb of SeD = 30

DIET Requests		
Total	Max	Average
6.684 s	287 ms	230.483 ms
Find Info		
Total	Max	Average
1.658 s	122 ms	57.172 ms
Latency Info		
Total	Max	Average
1.615 s	113 ms	55.69 ms
Solve Info		
Total	Max	Average
3.411 s	122 ms	117.621 ms

Request Property

reqID	13
Service	service14
SeD	paraci14_1
FindTime	61 ms
Latency	84 ms
SolveTime	118 ms
TotalTime	263 ms
paraci14	

DIET Platform Property
Nb DIETRequest = 679
Nb of SeD = 8

DIET Requests		
Total	Max	Average
317407.385 s	1585.75 s	363.582 s
Find Info		
Total	Max	Average
3.179 s	134 ms	3.641 ms
Latency Info		
Total	Max	Average
147067.764 s	863.055 s	168.463 s
Solve Info		
Total	Max	Average
170336.442 s	1103.756 s	195.116 s

Request Property

reqID	676
Service	dgemm
SeD	node69-015_1
FindTime	3 ms
Latency	448.093 s
SolveTime	90.776 s
TotalTime	538.872 s
node69-015.lyon.grid5000.net	

Outline

1. Introduction
2. Distributed Interactive Engineering Toolbox
3. DIET Applications
4. Discussion

Applications

- BLAST application using DIET
    - Basic Local Alignment Search Tool.
    - Compare biological sequences such as nucleotides sequences
    - N-sequences versus one database
    - Multi-request files are partitioned intro several smaller requests
- DIET to analyse Cosmological Results
    - Study large scale structure and galaxy formation
    - RAMSES collects raw data
    - Galices analysis the data

Outline

1. Introduction
2. Distributed Interactive Engineering Toolbox
3. DIET Applications
4. Discussion
   - Conclusion
   - Future Work

Conclusion

- Scalable, open-source, and multi-application platform
- Concentration on several issues
  - performance evaluation (CoRI, FAST)
  - scheduling (plugin schedulers, workflow management)
  - deployment (planning and tool GoDIET)
  - data management and replication (JuxMem)
  - fault toleration (architecture and application)
  - monitoring (LogService and VizDIET)
- Large scale validation on the Grid'5000 platform

http://graal.ens-lyon.fr/DIET http://www.grid5000.org

Future work

- Need of implementation and validation of algorithms from the scheduling literature in real-life middleware infrastructures?
- Still room for fundamental research on algorithms
- Some problems waiting for solution
  - Finding accurate models
  - Large scale validation of algorithms (simulators, real grids?)
- Take a look at other applications
- . . .