Outline

• Context and goals
• Vigne overview
• Focus on 3 services
  - Node interface service
  - Communication service
  - Resource allocation service
• Conclusion and future works
Outline

• Context and goals
• Vigne overview
• Focus on 3 services
  - Node interface service
  - Communication service
  - Resource allocation service
• Conclusion and future works
Context: The facts

- Numerical simulations in various scientific fields
  - High Performance Computing
- Institutions and companies use clusters
- Heterogeneity of clusters
  - Hardware (CPU architecture, memory, network, ...)
  - Software (OS, cluster scheduler, libraries, ...)
- In some cases, not enough power with a cluster -> Grid Computing
Context: In a perfect world, we want ...

- Use a lot of resources
  - dynamic behavior
  - spread all around the world
- Use of all resources without taking care about resource location and access method
  --> Single System Image for user and applications point of view
Goals

😊 No security answers
😊 No firewall or proxy between resources
😊 Several thousand of heterogeneous clusters
😊 Transparency of use (SSI)
😊 Ease of use for administrators and users
😊 Support for dynamic behavior
  - Administration tasks
  - Multiple cluster failures
Outline

- Context and goals
- Vigne overview
- Focus on 3 services
  - Node interface service
  - Communication service
  - Resource allocation service
- Conclusion and future works
Vigne overview

- Project started by Louis Rilling (Memory sharing in large scale environment)
- Fully distributed set of operating system services for federated clusters
- Materialized by a daemon running on each node of the grid
- Running prototype
- Written in C (30,000 lines)
Outline

- Context and goals
- Vigne overview
- Focus on 3 services
  - Node interface service
  - Communication service
  - Resource allocation service
- Conclusion and future works
Node interface service: What is an node in Vigne?

- Single computer running Linux
  - Direct interface
- Cluster running a Single System Image (Kerrighed)
  - Interface with any node
- Cluster running batch scheduler (OpenPBS, LSF)
  - Interface with head-node
Node interface service:  
A common interface

- Set of generic functions
  - `os_interface_submit_job`
  - `os_interface_get_load`
  - `os_interface_kill_job`
  - `os_interface_get_state`
  - `os_interface_clean_job_files`
  - `os_interface_checkpoint_job`
  - `os_interface_restart_job`
  - `os_interface_book_resource`
  - ...
Outline

• Context and goals
• Vigne overview
• Focus on 3 services
  - Node interface service
  - Communication service
  - Resource allocation service
• Conclusion and future works
Scalable communication infrastructure

- Structured P2P overlay (Louis Rilling) [Pastry, Bamboo]
  - Location independent naming
  - Lookup in $\log(n)$ hops
- Unstructured P2P overlay [Scamp]
  - Attributes-based searches
- Good properties for large scale and dynamic behavior
  - Self-healing
  - Self-organizing
Scalable communication infrastructure

- 2 P2P overlays but a reduced overhead
  - Same naming scheme for both overlays
  - Same failures detector
  - Lightweight implementation of unstructured overlay (bootstrap, self-healing)
- Scalability experiments (500 to 2000 nodes)
Outline

• Context and goals
• Vigne overview
• Focus on 3 services
  - Node interface service
  - Communication service
  - Resource allocation service
• Conclusion and future works
Scalable resource allocation service

- To submit a job to the system:
  - describe job informations

  ```
  [Job desc]
  Exec=/home/ejeanvoi/app/app_bin
  In_File=/home/ejeanvoi/app/input
  Out_File=/home/ejeanvoi/app/output
  [Required]
  Architecture=X86_64
  OS=LINUX
  OS Version=2.6
  Cluster_scheduler=OpenPBS
  Cluster_scheduler_version=2.3
  Network=GIGABIT_ETHERNET
  Memory=2048
  Nodes=6
  [Opt]
  Memory=4096
  Nodes=12
  ```

- use Vigne client
  ```
  $> ./client -n $any_machine_running_Vigne -p $port -t 0 -f $query_file
  ```
Scalable resource allocation service

- Fully decentralized and fault tolerant service
- 2 features:
  - discovery: finding suitable resources according description
    - based on unstructured overlay
    - multi-attributes and range queries
    - several protocols: flooding, random walk, ...
  - choosing the most suitable resource according a policy (less loaded resource, proximity of a file server, ...)

Scalable resource allocation service

- Experiments on the Grid'5000 testbed
  - Nodes of 3 sites (Opteron at 2 GHz and 2.2 Ghz, Xeon at 2.4 Ghz)
  - Bag of tasks: iterative processes -> heavy load
  - Nodes dispersion

<table>
<thead>
<tr>
<th>Grid'5000 site</th>
<th>#CPU used</th>
<th>Execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bordeaux</td>
<td>88</td>
<td>1740 s</td>
</tr>
<tr>
<td>Orsay</td>
<td>416</td>
<td>1955 s</td>
</tr>
<tr>
<td>Rennes – Paraci</td>
<td>128</td>
<td>1953 s</td>
</tr>
<tr>
<td>Rennes – Parasol</td>
<td>112</td>
<td>2689 s</td>
</tr>
</tbody>
</table>

- Job have been submitted to Vigne every 20 s
- Average execution time (746 nodes) with Vigne resource allocator: 1918 s

average: 1972 s
Outline

- Context and goals
- Vigne overview
- Focus on 3 services
  - Node interface service
  - Communication service
  - Resource allocation service
- Conclusion and future works
Conclusion

- Running operating system for large scale grids
- Low level services built on the top of P2P overlays
- Self-healing and self-organizing operating system
- Efficient high level services
  - application management
  - memory sharing
  - resource allocation
Future works

- New resource discovery protocols
- New resource allocation policies
- Run real applications
- Co-scheduling (Boris Daix)
  - Code coupling
- Application control (Master student)
- Security and restricted network access
Thank you!