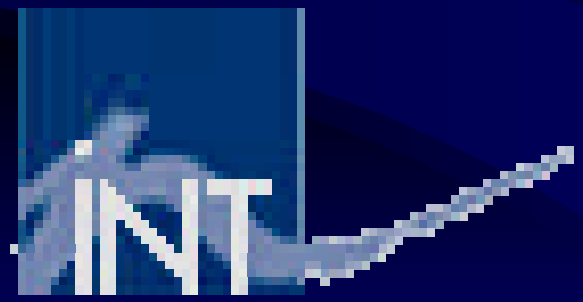
Un système de gestion de tâches pour la machine parallèle MPC



Encadrants :

Alexandre FENYO (LIP6)

Philippe LALEVEE (INT)

Introduction / Sujet

- **Lieu** : LIP6
- **Cadre** : Projet Multi-PC (MPC)
- **Machine parallèle à faible coût**
  - Grappe de PCs reliés par un réseau haut-débit
  - Technologie HSL développée au laboratoire
- **Objet** : réalisation d ’un outil d ’administration pour la machine MPC
- **But** : permettre à plusieurs utilisateurs de bénéficier de cette puissance de calcul

Plan de la Présentation

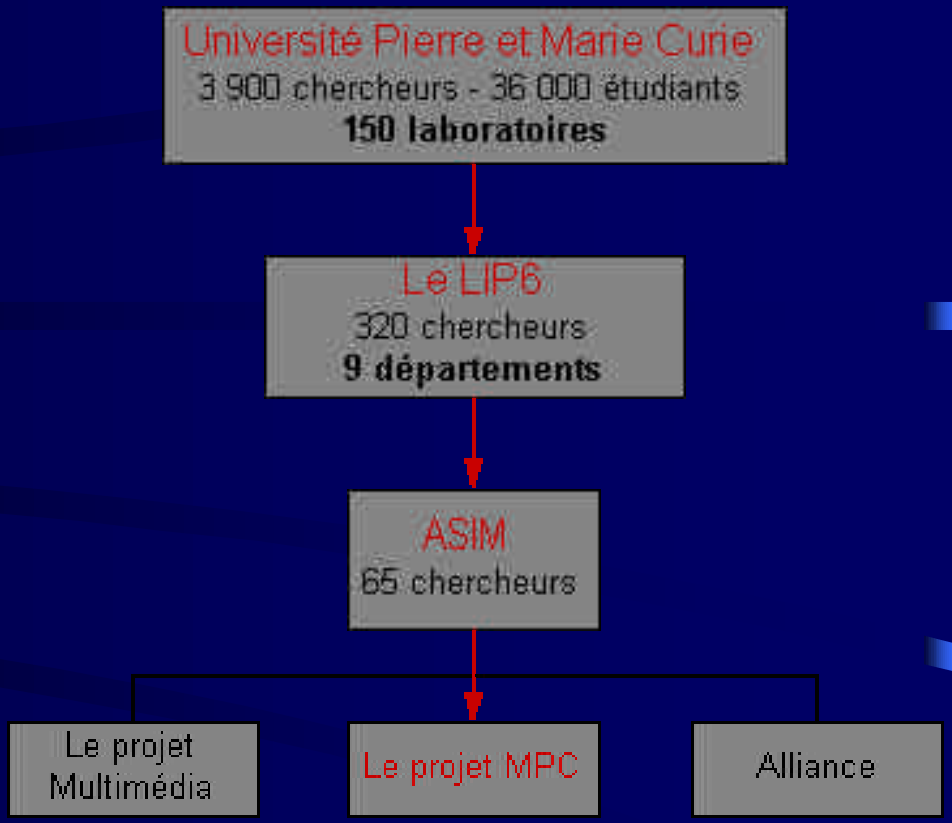
- Introduction / Sujet
- Le Laboratoire d ’Informatique de Paris 6
- La machine MPC
- Le JMS pour la machine MPC
- Conclusion
- Sites Internet

Plan de la Présentation

- Introduction / Sujet
- Le Laboratoire d ’Informatique de Paris 6
- La machine MPC
- Le JMS pour la machine MPC
- Conclusion
- Sites Internet

Le LIP6

- 9 thèmes de recherche (dont ASIM)
- 8 projets transversaux (dont MPC)



- ASIM : Architecture des systèmes intégrés et micro-électroniques
- 3 projets de recherche (MPC, Multimédia, Alliance)

Plan de la présentation

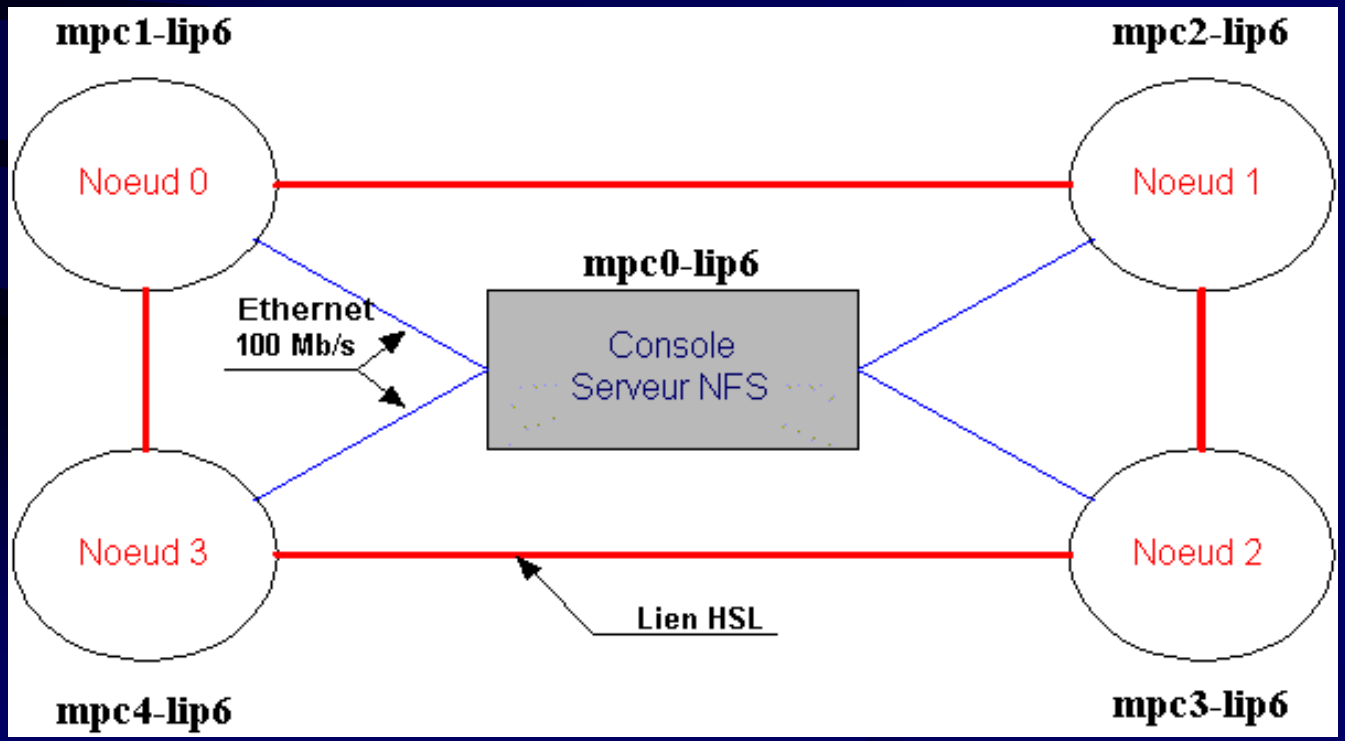
- La machine MPC 
  - Le projet MPC
  - Architecture matérielle
  - Architecture logicielle
  - PVM-MPC

Le projet MPC

- Démarré en janvier 1995 (Alain GREINER)
- Machine parallèle performante à faible coût
- Nœuds de calcul = PCs (Bi-pentium)
  - 4 ou 8 nœuds (4 au LIP6)
  - Réseau HSL, cartes FastHSL, liens HSL
- Couches logicielles (PVM)
- Buts :
  - fournir une puissance de calcul
  - comparaisons avec FastEthernet, Myrinet...

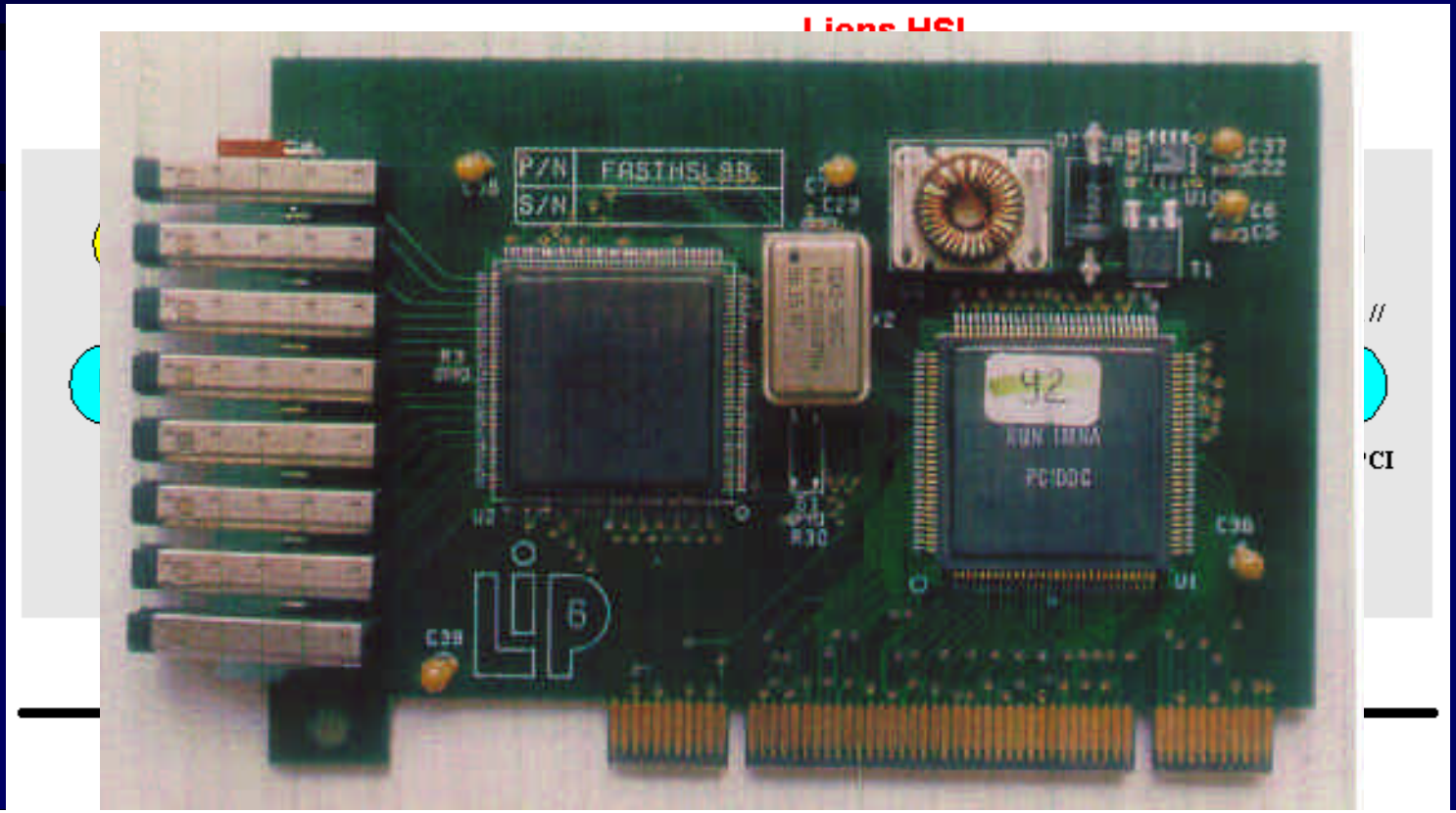
Architecture matérielle

- 4 nœuds de calcul = 4 Bi-pentium
- 1 console pour l ’exploitation de la machine
- Réseaux Ethernet (100 Mb/s) et HSL (1 Gb/s, full duplex)



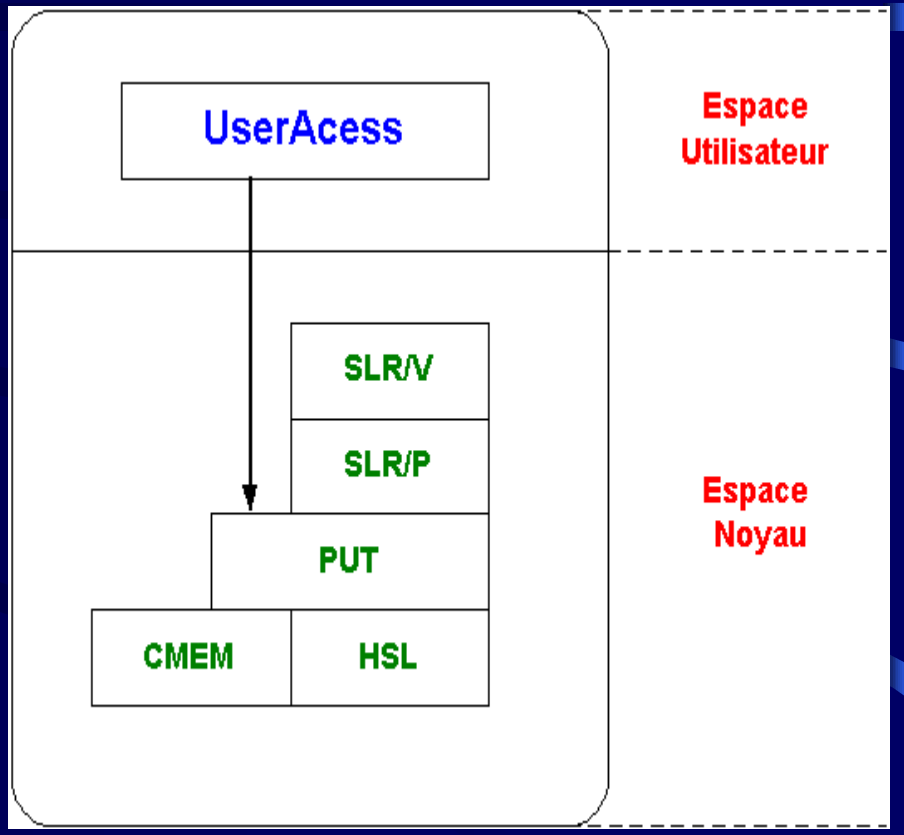
Architecture matérielle

- **PCI-DDC** : contrôleur de bus PCI intelligent
- **RCUBE** : routeur rapide possédant 8 liens HSL à 1 Gbit/s
- Ecriture en mémoire distante



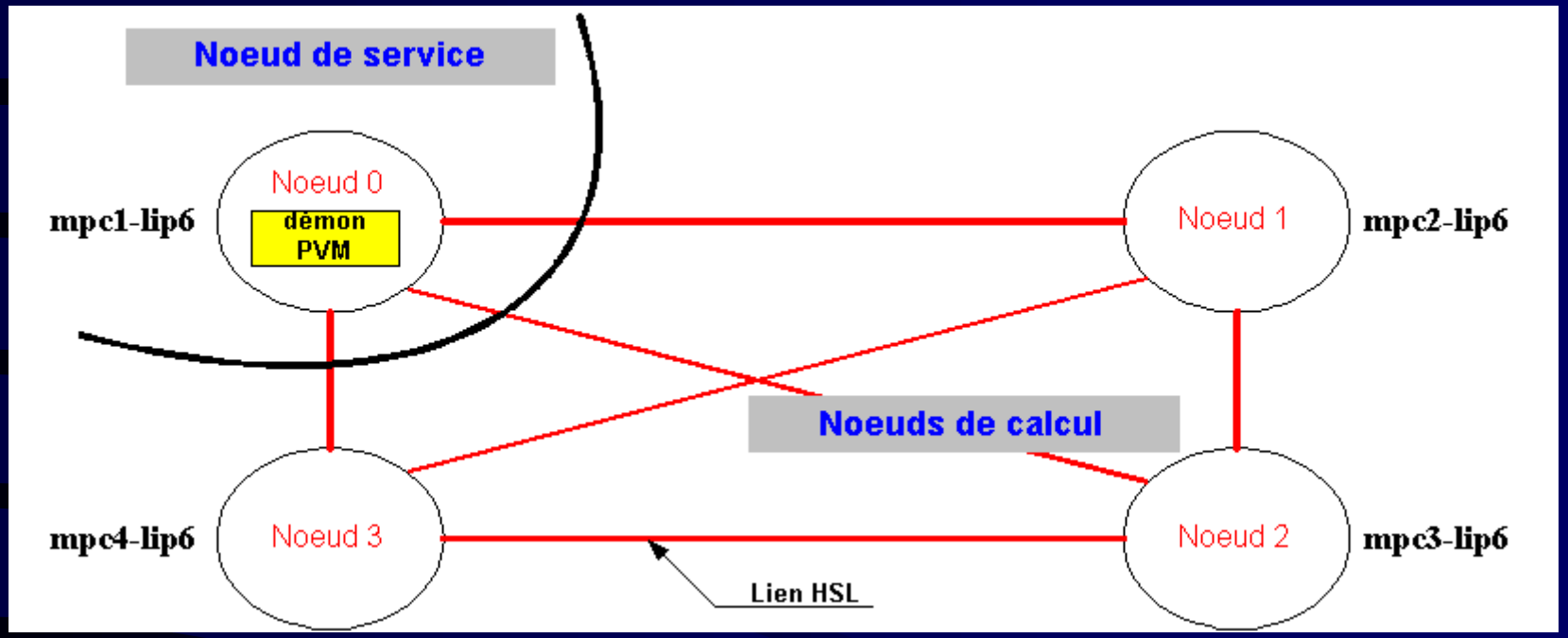
Architecture logicielle

- Communiquer avec la carte FastHSL à moindre coût
- Différents services
- Mode Remote Write
- 2 drivers ou pilotes
  - CMEM
  - HSL
- 2 démons



Portage sur Linux de PUT

Architecture logicielle / PVM-MPC



- PVM 3.3 (soit Ethernet, soit MPC)
- Architecture à démon unique
- Lancement des tâches esclaves
- 1 driver + le démon PVM

Plan de la présentation

- Le JMS pour la machine MPC 
  - Généralités sur les JMS
  - Un JMS-MPC
  - Les composants du JMS-MPC
  - Les queues / Le calendrier
  - L ’exécutif
  - Les fonctionnalités
  - Architecture logicielle
  - Difficultés techniques

Généralités sur les JMS

JMS = Job Management System

- Une interface utilisateur
  - Exécution de tâches locales ou distantes par l'intermédiaire de files d'attente
  - Spécifier les ressources
  - Connaître l'état des files
  - Suppression, suspension / reprise, [illegible]
  - Ligne de commande ou environnement graphique
- Un scheduler
  - Politique de lancement des tâches
  - Priorité dépendante de :
    - temps d'attente dans les files
    - ressources nécessaires
    - type de la tâche
    - identité de l'utilisateur
  - Priorités : statiques ou dynamiques
  - Exemples : FIFO, périodes utilisateurs
- Un gestionnaire de ressources
  - Allocation des ressources aux tâches
  - Gérer l'état des ressources
  - Collecter des informations sur l'exécution des tâches
- Un environnement sécurisé
- Un système de rapatriement des logs

Autres fonctionnalités :
- Migration de tâches
- Reprise d'exécution

Généralités sur les JMS

- Trois modes d ’utilisation :
  - Dedicated mode
  - Space sharing
  - Time sharing

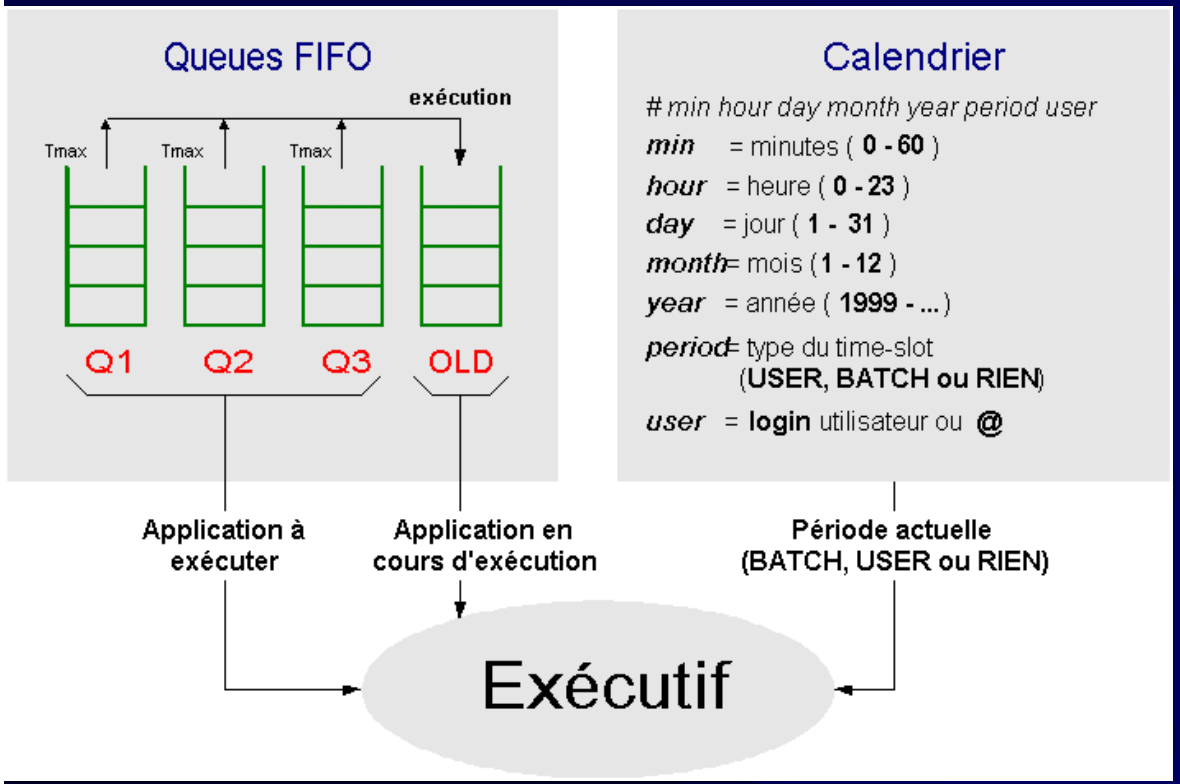
- Systèmes existants :
  - DQS et Codine
  - LSF
  - NQE
  - Condor et NQS
  - GNU Queue

- Systèmes existants :
  - Clusters hétérogènes (UNIX)
  - Pas de migration dynamique

JMS pour la machine MPC

- Pourquoi un JMS ?
  - Automatiser la gestion du réseau HSL et de PVM-MPC
    - automatiser la gestion des drivers CMEM et HSL
    - automatiser le démarrage des démons *hslclient* et *hslserver*
    - automatiser le chargement du driver PVM pour MPC
    - automatiser le démarrage du démon PVM
  - Permettre à plusieurs utilisateurs de lancer leur applications PVM pour MPC
- Pourquoi un JMS spécifique ?

Les composants du JMS



- Dedicated mode
- Architecture LIP6
- Nombre de nœuds paramétrable
- UNIX FreeBSD
- 2 interfaces
  - ligne de commande
  - CGI (Web)

- Les queues : lancer des applications PVM-MPC
- Le calendrier : différents types de périodes
- L ’exécutif : applique les règles de priorités

Les queues / Le calendrier

- 3 queues FIFO
- Paramètre *Tmax*

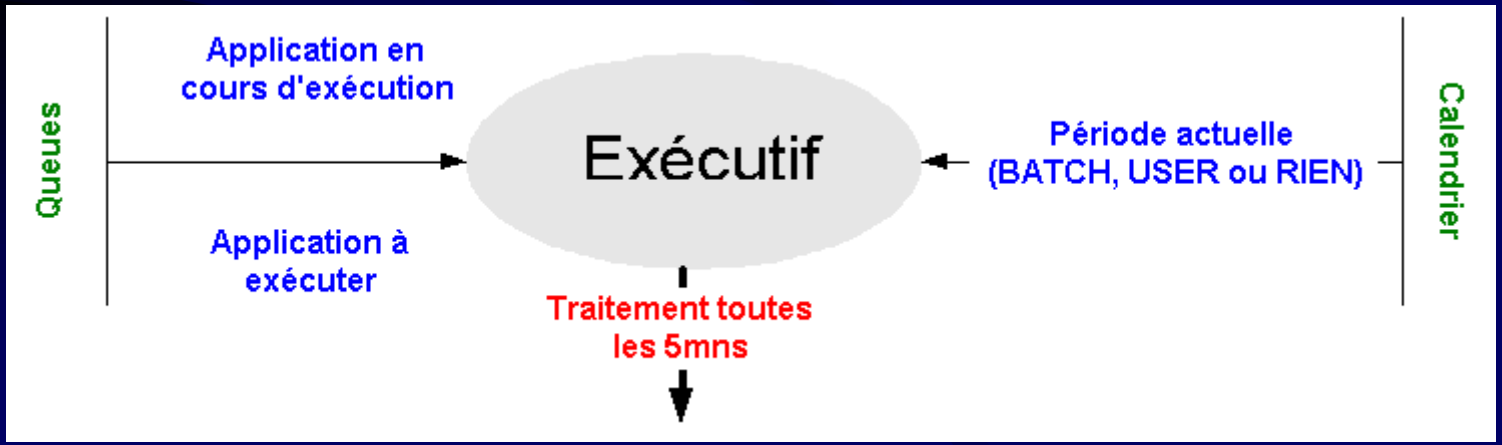
$\mathbf{Tmax_{Q1} < Tmax_{Q2} < Tmax_{Q3}}$

  - queue rapide, moyenne, lente
- La queue *OLD*
- Un élément = 1 fichier
  - username
  - tâche à exécuter
  - nœud
  - assurance vie
  - Tav
  - reboot avant exécution
  - mail

- 3 types de périodes :
  - BATCH = tous les utilisateurs sont équivalents
  - USER = privilégier un utilisateur
  - RIEN = geler l ’exécution des applications
- Consultable par tous les utilisateurs
- Mis à jour par l ’admin.

L ’exécutif

- Organe central du JMS
- Exécution toutes les 5 minutes
- Avant chaque exécution, les démons PVM et MPC sont relancés
- Les priorités :
  - dans une même queue : FIFO
  - Q1 > Q2 > Q3
  - période de type USER



Les fonctionnalités

L 'utilisateur

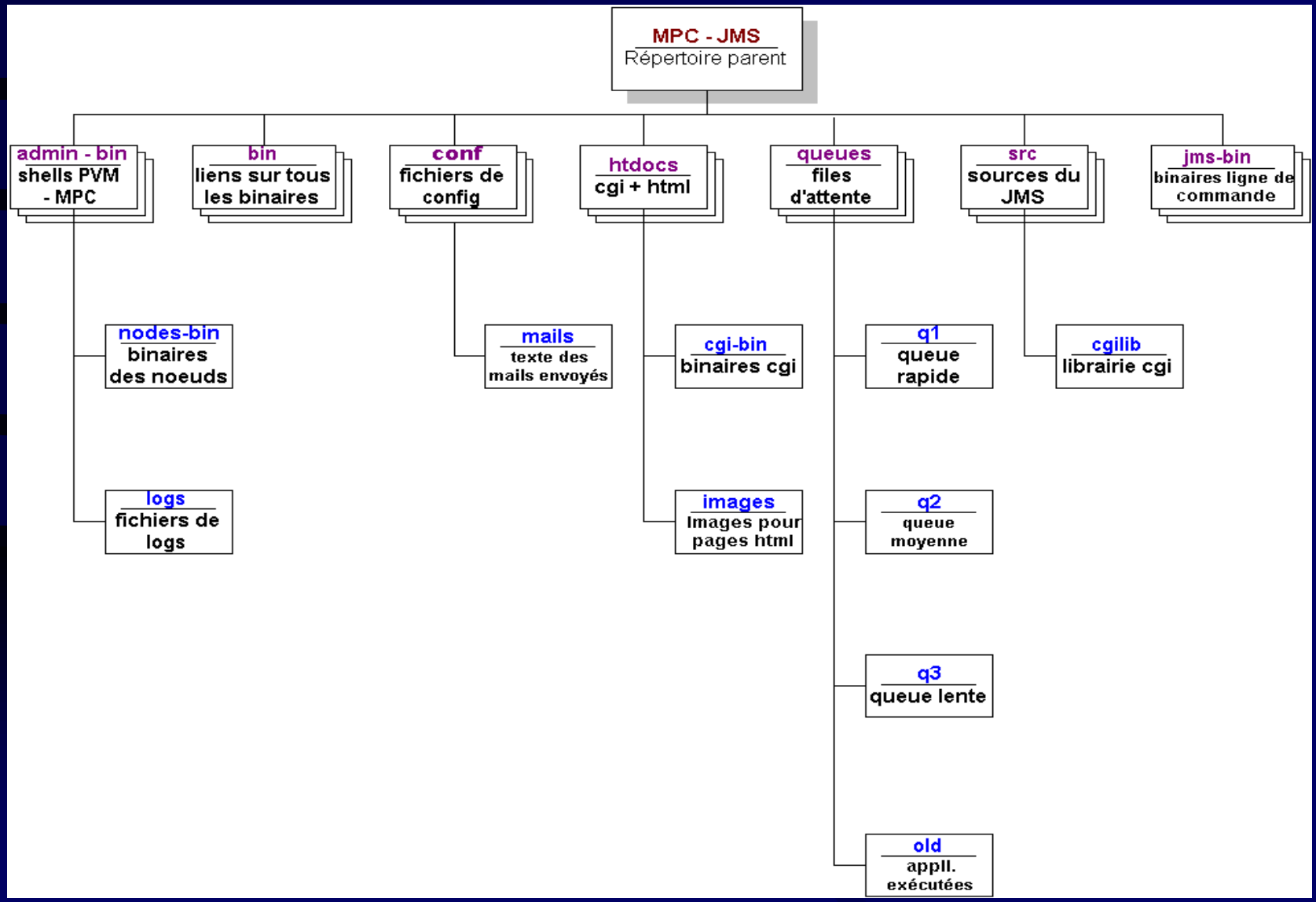
- Mettre une appli. en queue
- Consulter les queues
- Supprimer une appli.
- Appli. en cours d 'exécution (voir / tuer)
- Consulter le calendrier
- Utilisateur privilégié
- Tester l 'appli en cours
- Liste des démons / drivers
- Liste processus utilisateur
- Redémarrage des nœuds
- Rapatrier fichiers de log

L 'administrateur

- Configurer le calendrier
- Configurer les queues
- Arrêter / redémarrer l 'exécutif
- Vider toutes les queues
- Redémarrer tous les nœuds de calcul
- Devenir un simple utilisateur

- Période de type « USER @ »
- Assurance vie
- Les mails

Architecture logicielle (I)



Architecture logicielle (II)

- 25 shells UNIX
- 10 fichiers hypertext (HTML)
- 11 binaires pour la ligne de commande
- 27 exécutables CGI
- 5 fichiers de configurations
  - fichier de configuration générale
  - files d ’attente (*Tmax*)
  - calendrier
- Les sources (dont 3 librairies et un *Makefile*)
- Les documentations

Difficultés techniques

- Les problèmes de lock
  - un fichier pour tout le JMS
- L ’interface CGI (réalisation)
- Des droits de super-utilisateur pour les utilisateurs (bit SUID)
- Sécurité et interface CGI
  - utilisation de Apache
  - login + mot de passe

Job Management System for MPC

• **Jobs**	Put a job in a queue	Click
	Remove a job from a queue	Click
	Kill the job in execution	Click
• **Status**	State of queues	Click
	State of calendar	Click
	Become the priority user	Click
	See the job in execution	Click
	Test of the job in execution	Click
	List of MPC-PVM deamons and drivers	Click
	List of all your processes on all nodes	Click
• **Logs**	Logs go back	Click

Server design : Olivier Gluck

$Date: 1999/05/10 15:00:13 $

Conclusion

- Différentes phases du stage
- Evolutions du JMS
  - portage sur d ’autres systèmes UNIX
  - autres types d ’applications (MPI)
- Documentations
  - Le rapport
  - Manuel d ’installation
  - Guide d ’utilisation
- Remerciements

Sites INTERNET

- Site de l'Université de **Paris 6** http://www.admp6.jussieu.fr/
- Site du **LIP6** http://www.lip6.fr/
- Site du département **ASIM** http://www-asim.lip6.fr/
- Site de la machine **MPC** du LIP6 http://mpc.lip6.fr/

- Site du JMS **CODINE** http://www.genias.de/welcome.html
- Site du JMS **NQE** http://www.sgi.com/software/nqe/
- Site du JMS **GNU Queue** http://bioinfo.mbb.yale.edu/fom/cache/1.html
- Site du JMS **LSF** http://www.lerc.nasa.gov/WWW/LSF/lsf_homepage.html
- Site du JMS **Condor** http://www.cs.wisc.edu/condor/
- Site du JMS **DQS** http://www.msi.umn.edu/bscl/info/dqs/

- Site de la librairie **CGIC** http://www.boutell.com/cgic/
- Site du serveur HTTP **Apache** http://www.apache.org/