

Adaptation de Logiciels : du système aux applications [†]

Bertil Folliot

Professeur à l'Université Paris 6

**LIP6 / CNRS, Université Pierre et Marie Curie, Paris 6
et INRIA Rocquencourt - (nouveau) SOR**

*[†] Ce projet est partiellement financé par les projets RNRT Phenix (99S0361),
France Télécom R&D ISA (001B117) et le LIP6 (PLERS 2000)*

Bertil.Folliot@lip6.fr

www-sor.inria.fr/mvv

Ecole Grappes - mai 2001 - 1

Plan d'Alsa

- + **Historique et problématique**
- + **Objectifs du projet « Machine Virtuelle Virtuelle »
(et des MVVs)**
- + **Architecture, réalisations : MVV V0, MVV V1**
 - 1 réseaux actifs, caches webs, satellites
- + **Proposition de programme de recherche CNRS (2001-2005)**
- + **Travaux en cours et progression du projet MVV**

Ecole Grappes - mai 2001 - 2

Historique

- + **début-1997 : ☺ les MVVs, 3 personnes**
- + **1998 : 1 stage DEA**
- + **1999 : 1 thèse, 1 DEA, 1/2 ingénieur**
- + **2000 : 2 thèses, 3 DEA, 1 ingénieur**
- + **Collaborations avec : 2 Maître de Conférences, 1 Chercheur INRIA, + 2 thèses**
- + **fin 2001 : >16 (?)**

Ecole Grappes - mai 2001 - 3

Introduction (autour du « système »)

- + **Multitudes de normes et de standards (de facto ou de jure)**
 - 1 ODP, CORBA, JAVA, DCOM, JINI, BLUETOOTH...
 - 1 Technologies logicielles rigides (!)
- + **Multiplication des cas particuliers**
 - 1 Qualité de Service, Tolérance aux fautes, performances, taille...
- + **Multiplication des « domaines informatiques »**
 - 1 travail coopératif, télétravail, multimédia, mondes virtuels, réseaux actifs, domotique, informatique « embarqué », GRID :) ...
- + **Évolution rapide du matériel**
 - 1 loi de Moore valide pour les X prochaines années (?)
 - 1 PC, PDA, Cartes à puces, étiquettes électroniques

Ecole Grappes - mai 2001 - 4

Où est le problème ?

- + **La recherche en système n'est plus de re-faire un système**
 - 1 trop long, trop rigide + cf. évolution du matériel
- + **Les besoins des « domaines informatiques » (applications émergentes) changent les objectifs des logiciels (performances) :**
 - 1 répartition géographique des utilisateurs et des ressources,
 - 1 forte contrainte de sécurité (données sensibles, copyright, droits d'accès),
 - 1 modèles de données complexes et réparties,
 - 1 configurations dynamiques de partenaires hétérogènes,
 - 1 conditions d'interactions inconnues lors de la conception.

Ecole Grappes - mai 2001 - 5

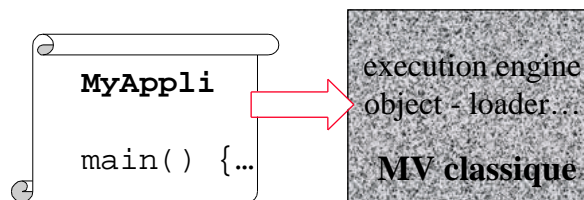
Pourquoi est-ce un problème ?

- + **Chaque « domaine » à un environnement d'exécution adapté (langage, API, OS ou « colle OS »)**
 - 1 ex. impression - postscript / dvips - « imprimer » / printcap - sélect. d'impr.
 - 1 réseaux actifs - langage dédié (bytecode) / send-receive / OS ?
- + **Le nombre de domaines est en forte augmentation**
 - 1 fonction du matériel (PDA)
 - 1 ou du logiciel - QdS (réalité virtuelle)
- + **Les besoins changent/évoluent au cours de l'exécution**
 - 1 cf. Internet (latence, bande passante, « hot spot of the week »)
 - 1 spécifications initiales (matériel, QdS)
- + **NOUVEAUX langages et/ou API et/ou OS**
- + **MAIS figé et/ou non interopérable et/ou difficile à réutiliser**

Ecole Grappes - mai 2001 - 6

Un exemple : les Machines virtuelles

- + **Java VM**
 - 1 En utilisation croissante pour résoudre les problèmes systèmes
 - 1 Applications “portables”, code compact
 - 1 “sure”, (un peu) spécialisable
- + **Chargement de bytecode, interprétation, JIT-C,**



Ecole Grappes - mai 2001 - 7

Limitations des machines virtuelles

- + Le « domaine » de la JVM suppose : beaucoup de mémoire, pas d'accès à l'OS, pas (beaucoup) de QdS
- + f(**matériel**) ? carte à puce (JavaCard), téléphone (KVM)
- + f(**propriétés**) ? persistance (Pjama), temps réel (RT-Java)
- + f(**temps qui passe**) ? gestion de crise (FT-Java)
- + **Les MVs sont une étape dans la bonne direction, MAIS :**
 - 1 peu flexible (nouveau domaine => nouveau langage + nouvelle MV)
 - 1 peu adaptable
 - 1 difficilement interopérable

Ecole Grappes - mai 2001 - 8

Plan

- + Historique et problématique
- + Objectifs du projet « Machine Virtuelle Virtuelle » (et des MVVs)
- + Architecture, réalisations : MVV V0, MVV V1
 - 1 réseaux actifs, caches webs, satellites
- + Programme de recherche CNRS (2001-2005)
- + Travaux en cours et progression du projet MVV

Ecole Grappes - mai 2001 - 9

Notre solution : les MVUs

- + **Virtualiser la machine virtuelle = MVV**
 - 1 **MVV** = [MV]V est une plate-forme d'exécution (**MV**) dans laquelle on construit son environnement d'exécution (appelé **MVlet**) : langage, API, modules systèmes...
- + Une **MVlet** correspond à un domaine d'application
 - 1 Spécification exécutable de haut-niveau (écrite en langage VVM)
 - 1 Chargeable/déchargeable dynamiquement
 - 1 Vérification des propriétés
- + Un mécanisme d'exécution au sein de la MVV pour toutes les MVlets
 - 1 Favorise l'interopérabilité et la réutilisation de code existant
 - 1 Partage les ressources physiques entre les MVlets
 - 1 Permet d'incorporer des optimisations agressives

Ecole Grappes - mai 2001 - 10

Pourquoi la MVU résout le problème

- + **Adaptabilité/spécialisation** du support d'exécution aux divers domaines - ou aux applications elles-mêmes (OS + langage)
- + **Flexibilité/extensibilité** pour ajouter/modifier des fonctionnalités, en fonction de l'application, de son utilisation ou des évolutions matérielles (configuration/re-configuration + déploiement)
- + **Interopérabilité** pour l'échange de données entre applications, pour la mobilité (code et/ou donnée) et pour favoriser la réutilisation
- + Prise en compte de la **logique métier** (expérience PLERS)
 - 1 tout ne peut/ne doit être adaptable/flexible/interopérable

Ecole Grappes - mai 2001 - 11

Ce que les MVU ne font pas

- + Aucune utilité pour un grand nombre de « Domaines informatiques » : Systèmes statiques et/ou prédictibles et/ou sans QdS
- + Pas de solution particulière pour les autres - aucun algorithme (réseaux actifs, caches flexibles, etc.) ou langage (DSL, etc.) - mais par effet de bord : adapté, flexible, interopérable...
- + Complémentaires (orthogonal ?) aux travaux « chauds » - AOP, EJB, CCM- composition, vérification de comportement, sécurité, répartition... ou Meta-Computing (Legion, Globus) ou systèmes à larges échelles (Globe, Perdis)

Ecole Grappes - mai 2001 - 12

Par rapport à l'existant

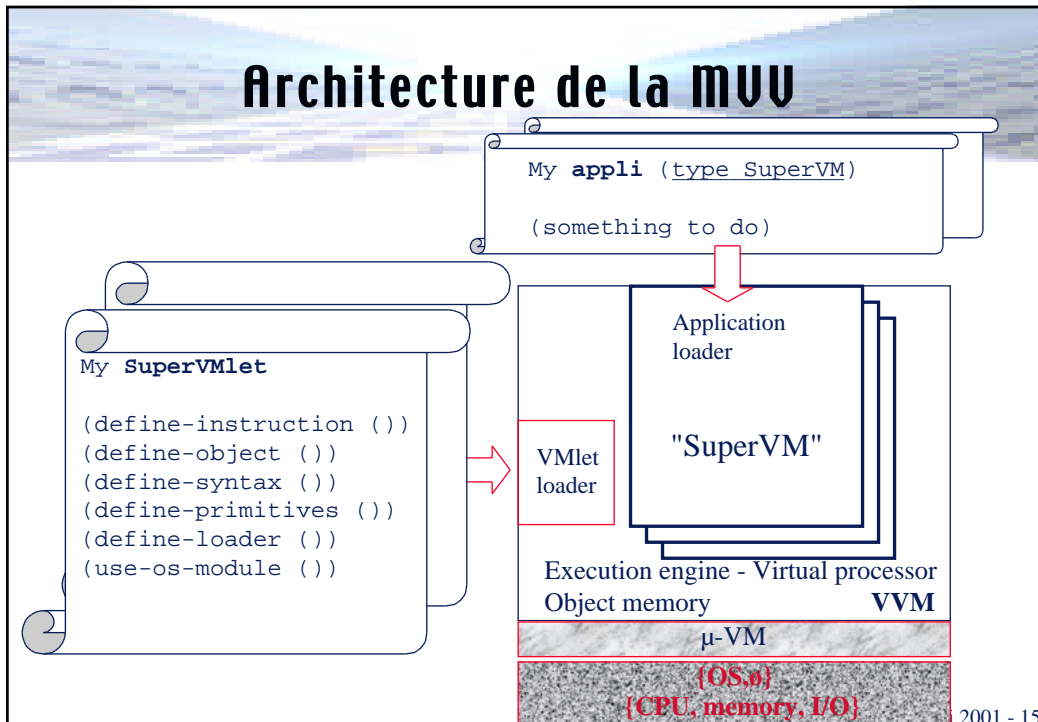
- + **Machines virtuelles spécialisables**
 - 1 génération d'une MV en fonction des performances, de la taille du code, du domaine, etc. (JavaCard, PLAN, Harissa) : peu/pas extensible, le même travail doit être refait pour chaque domaine
- + **Système d'exploitation flexible / MOP**
 - 1 changement de fonctionnalités et d'interfaces (SPIN, ExoKernel, Fluke) / AperiOS : un unique langage dédié
- + **Systemes embarqués**
 - 1 MultOS, Windows CE, SCP, μ CLinux, KVM (Sun - réutilisation de CCG-LIP6) : différences avec la problématique ?
- + **Interopérabilité des langages**
 - 1 exécution de plusieurs langages (Java, Smalltalk, VisualBasic pour la Machine Virtuelle Universelle d'IBM) : basé sur une MV existante (Smalltalk), pas extensible

Ecole Grappes - mai 2001 - 13

Plan

- + Historique et problématique
- + Objectifs du projet « Machine Virtuelle Virtuelle » (et des MVVs)
- + Architecture, réalisations : MVV V0, MVV V1
 - 1 réseaux actifs, caches webs, satellites
- + Programme de recherche CNRS (2001-2005)
- + Travaux en cours et progression du projet MVV

Ecole Grappes - mai 2001 - 14



Réalisation (1) : Machine Virtuelle Récursive (MVU VO)

- + **Machine virtuelle à la Scheme**
 - 1 Manipulations sur la syntaxe
 - 1 Manipulations sur la sémantique :
 - accès direct aux objets d'implantation (méthodes, contextes, fermetures...)
 - accès directs aux primitives d'exécution (ajout d'instructions et de primitives, utilisation du moteur d'exécution)
 - réalisation au niveau utilisateur : classes (avec héritage et passage de messages), synchronisation, processus légers... (~15 % du C)
- + **Réalisations de deux environnements pour réseaux actifs (PLAN et ANTS)**
 - 1 ~500 lignes de services (API), ~50 lignes pour le protocole lui-même

Ecole Grappes - mai 2001 - 16

Réalisation (2) : μ-Machine Virtuelle (MVU V1)

- + **Générateur dynamique de code (de MV)**
 - 1 Interface, schéma ou C (~100% du C)
 - 1 Chargement dynamique de bibliothèques systèmes (dlsym)
 - 1 Compiler Compiler Generator - CCG (Réutilisé par SUN pour un proto. de KVM)
- + **Exemple : cache Web en micro-VM**
 - 1 Une nouvelle politique s'écrit en ~10 minutes
 - 1 Se charge et se compile dans la micro-VM en 50ms
 - 1 Se change (si elle était déjà compilée) en 0,5 ms
 - 1 Benchmarks : ajout/changement de politique, observation à la volée, mode trace...
- + **Le langage de programmation du cache **EST** le langage d'implantation du cache.**
 - 1 Programmation par aspects manuel (?)

Ecole Grappes - mai 2001 - 17

Plan

- + Historique et problématique
- + Objectifs du projet « Machine Virtuelle Virtuelle » (et des MVVs)
- + Architecture, réalisations : MVV V0, MVV V1
 - 1 réseaux actifs, caches webs, satellites
- + Proposition de programme de recherche CNRS (2001-2005)
- + Travaux en cours et progression du projet MVV

Ecole Grappes - mai 2001 - 18

Adaptation dynamique de Logiciels - du Système aux Applications

Jean-Pierre Briot, Bertil Folliot (février 2001)

- + **Objectif : concevoir, réaliser et expérimenter de nouvelles solutions, s'adaptant/anticipant les besoins de la prochaine génération de l'informatique.**
 - 1 Applications potentielles dans : télécommunications (Internet nouvelle génération, mobilité) multi-média, travail collaboratif, mondes virtuels, télé-formation, enseignement à distance, support pour le médical, espaces actifs, community-ware, domotique, etc.
- + **Intérêt : réunir les différentes communautés du « logiciel » : système, interstitiels ("middleware"), langages (à objets), architectures logicielles, prototypage, agents logiciels + langage et vérification**

Ecole Grappes - mai 2001 - 19

Plan

- + Historique et problématique
- + Objectifs du projet « Machine Virtuelle Virtuelle » (et des MVVs)
- + Architecture, réalisations : MVV V0, MVV V1
 - 1 réseaux actifs, caches webs, satellites
- + Programme de recherche CNRS (2001-2005)
- + Travaux en cours et progression du projet MVV

Ecole Grappes - mai 2001 - 20

Travaux en cours

- + **Axe noyau MVV : (construire « son » OS)**
 - 1 Intégration de la YNVM dans Think (THink is Not a Kernel - FT R&D / RNRT Phenix)
 - 1 Réutilisation de micro-noyaux flexibles existants
- + **Axe applicatif : (instancier chaque étape de la VVM)**
 - 1 VMlet de réseaux actifs « actifs » (généralisation du concept des réseaux actifs) (CTI FT R&D ISA)
 - 1 Cache Web réactifs (cache web flexible + piles d'observation réseau Pandora)
 - 1 PLERS : reconfiguration logiciel pour satellite (projet LIP6 - proposition CNES)
- + **Projet DEAL (INRIA) - déploiement d'applications à large échelle**

Ecole Grappes - mai 2001 - 21

Bientôt...

- + **Fusion noyau et applicatif :**
 - 1 Sécurité et sûreté de fonctionnement (à la composants CORBA ou EJB)
 - 1 Documents actifs (généralisation des réseaux actifs actifs)
 - 1 Configuration/reconfiguration de logiciels contraints
 - 1 Tisseurs d'aspects - VMlet d'aspects ?
 - 1 VMlets pour cartes à puces, VMlets Java
 - 1 Observation réactive
 - 1 Tolérance aux fautes et réplication adaptables

Ecole Grappes - mai 2001 - 22

Conclusion

- + **Le chemin est long vers les MVVs (5 ans ?)
mais un potentiel de recherche immense
et très approprié pour attaquer beaucoup de problèmes
actuels et (surtout) futurs**

Ecole Grappes - mai 2001 - 23

Questions ?

- + **PUB :**
**plusieurs postes de post-doc disponibles sur le projet VVM
(autour des logiciels flexibles pour satellites,
des composants flexibles,
de l'observation de BD flexible,
ou de la MVV elle-même (langage, interprétation,
compilation, etc.))**

Ecole Grappes - mai 2001 - 24