

Evaluation de Systèmes d'Exploitation Temps Réel pour les Applications d'Informatique Industrielle d'EDF

Erwan Becquet * {becquet@cnam.fr}, Laurent Bacon † {laurent.bacon@edf.fr}

Résumé

Cette étude est faite dans le cadre d'une évaluation plus globale d'une solution à base d'ORB temps réel à Qualité de Service. EDF évalue des architectures possibles pour la rénovation de certaines applications d'informatique industrielle. Les besoins des équipes d'intégration font apparaître une nécessité d'utiliser des produits du commerce dans la mise en oeuvre des futures applications. De plus des enquêtes ont permis aussi de mettre en évidence les besoins en terme de fonctionnalités, de performances et d'outils d'ingénierie. Dans cette optique, quatre systèmes d'exploitation temps réel du commerce: ChorusOS, LynxOS, pSOS+ et VxWorks ont été étudiés, on s'intéresse ici aux performances. ChorusOS est en cours d'évaluation fonctionnelle, l'évaluation opérationnelle devant être faite très prochainement. La prochaine étape consistera à évaluer les interactions entre ces ORBs et les systèmes temps réel présents.

1 Introduction

L'étude sur les systèmes d'exploitation temps réel est la première étape de l'étude dont le but est d'évaluer les architectures réparties à Qualité de Service, dont le besoin a été isolé grâce aux enquêtes utilisateurs. Notre étude s'inscrit dans une démarche globale concernant la mise en oeuvre d'application d'informatique industrielle à partir d'une approche orientée "composants du marché sur étagères" afin de réduire les coûts de développement et de maintenance. Parmi les propriétés assurées par les technologies candidates, le déterminisme temporel et la performance occupent un rôle essentiel. Notre étude concerne les systèmes d'exploitation temps réel, plus précisément, les fonctionnalités et les propriétés qui seront utiles à la conception des applications d'informatique industrielle réparties d'EDF. Notre travail vise à faire émerger un certain nombre de points de repères à propos des technologies candidates pour les futurs projets.

La première partie de ce document correspond à une analyse des besoins utilisateurs, analyse caractérisée par un questionnaire [1] et plusieurs enquêtes [2]. Nous poursuivons par une présentation de l'évaluation de performances, au niveau technique, mise en oeuvre et résultats. Enfin nous concluons et évoquons la poursuite de cette étude. Ce travail a donné lieu à une présentation [4] et à des rapports industriels EDF [3][5].

2 Etude des contraintes utilisateurs

La première étape de l'étude a consisté à recueillir les contraintes qui portent sur les systèmes d'exploitation temps réel au moyen d'entretiens réalisés à partir d'un questionnaire [1]. Le questionnaire a eu pour objectif de dégager les besoins en performances et certaines caractéristiques temporelles d'applications d'informatique industrielle, par exemple: temps de traitement, temps d'acheminement des informations, temps de réponse, débit et longueur des données, nature des informations échangées et leurs échéances ou périodes... Ce questionnaire a permis par ailleurs de caractériser les fonctionnalités des systèmes d'exploitation temps réel attendues par ces applications. Il aborde aussi des aspects plus généraux: la facilité d'administration, l'extensibilité du système (composants, serveurs, modules dynamiquement chargeables), les processeurs et bus

* Cnam, Laboratoire Cedric, 292 rue St Martin 75141 Paris Cedex

† E.D.F. - D.R.D. Département C.C.C (Contrôle Commande en Centrales), 6 Quai Watier, 78401 Chatou Cedex

de communication supportés, la capacité de reprise sur panne, les protocoles de communication, l'existence d'une couche de type bus logiciel... C'est ce questionnaire qui a permis de faire le choix des systèmes cibles de notre étude.

Une synthèse globale des entretiens [2] indique, sous forme de recommandations, les points clés à évaluer dans la poursuite de l'étude : sur les systèmes d'exploitation temps réel, et sur les plateformes d'exécution réparties à objets. Par ailleurs, cette synthèse résume les points caractéristiques de chaque enquête et permet l'élaboration d'un profil type d'application industrielle EDF [4]. Les applications d'informatique industrielle en centrale se répartissent sur plusieurs niveaux. Le niveau 0 est chargé de l'acquisition physique de données (majorité des échanges) et des commandes aux équipements (capteurs et actionneurs). Le niveau 1 effectue un traitement temps réel des données acquises et produit des données de synthèse. Le niveau 2 est chargé de traitements de supervision, de l'archivage, du poste opérateur et de l'interface avec les autres fonctions du contrôle de production globales à la centrale. L'architecture logicielle type d'une plate-forme d'informatique industrielle est donnée en figure 1.

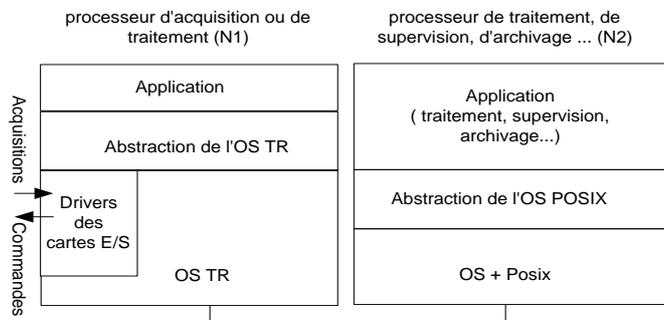


FIG. 1 – Architecture logicielle

3 Evaluation de Performances

3.1 Technique

Le but de ce travail était de développer une méthode générique de banc de tests pour système d'exploitation temps réel, objectif dicté par le fait que nous avons quatre systèmes temps réel à évaluer. Dans les systèmes temps réel, les APIs sont très semblables : mêmes fonctions, au nom et ordre des paramètres près. C'est en partant de ce constat que nous avons développé notre programme de tests. Tous les bancs de tests sont écrits à l'aide de macros dont les définitions vont varier suivant le système cible. Le travail de portage du banc de tests sur un nouveau système revient alors à redéfinir l'ensemble des macros utilisées. Ainsi, le portage sur le deuxième système n'a pris qu'une demi-journée, en comptant l'exécution des tests eux-mêmes et la récupération des résultats.

La solution retenue pour avoir une échelle de temps assez fine pour nos mesures est celle déjà utilisée dans [6]: nous allons exécuter l'appel système à mesurer un certain nombre de fois, suffisamment grand pour obtenir un temps moyen pertinent. Les tests réseaux sont un cas à part puisqu'ils utilisent l'interface des sockets, commune à quasiment tous les systèmes d'exploitation et disponibles sur les quatre systèmes de notre étude. Dans le même ordre d'idée, l'utilisation d'APIs POSIX lorsqu'elles existent permet de les réutiliser pour un autre système les supportant également. La description de ce banc de tests a donné lieu à un rapport EDF [5].

3.2 Mise en oeuvre

Beaucoup des tests mis en oeuvre se résument à exécuter séquentiellement N fois l'appel système en cours d'évaluation, puis à diviser le temps ainsi obtenu pour avoir un temps moyen. Mais certains appels systèmes ne peuvent pas être évalués de cette manière et exigent un traitement spécifique. Prenons pour exemple le calcul du temps du changement de contexte. Comme pour tous les temps, il faut trouver un moyen d'exécuter N changements de contexte sans autres appels systèmes parasites, pour pouvoir diviser ensuite le temps recueilli et en déduire un temps moyen. Pour ce faire, nous utilisons P tâches de même priorité qui vont demander un ré-ordonnement dès qu'elles ont la main, via une macro SWITCH_OP() qui sera redéfinie dans chaque système.

Dans chaque système, nous positionnons l'ordonnement pour que les tâches de même priorité soient gérées en tourniquet. Nous comptons les changements de contexte grâce à un compteur local, partagé par les tâches. Les opérations sur ce compteur (affectation et test) ont été évaluées négligeables en termes de temps d'exécution devant celui du changement de contexte. Nous avons préféré cette méthode à celle décrite dans [6], basée sur le même principe mais qui faisait appel aux tubes nommés pour provoquer les changements de contexte. Nous avons remplacé ceux-ci par un appel système moins invasif et qui perturbe moins les résultats. Dans certains systèmes, une fonction est même préconisée pour effectuer un changement de contexte.

3.3 Résultats

Ci-dessous la matrice des tests qui donne, pour chaque système d'exploitation, les cibles utilisées.

OS	MBX821	MVME2700	Pentium II
LynxOS		X	X
pSOS+	X	X	
VxWorks	X	X	

La figure 2 expose les temps (normalisés) obtenus sous VxWorks sur la carte MBX 821 à base de processeur PowerPC, pour les files de messages. Les temps mesurés ici sont la création, la destruction, l'envoi et la réception avec les variantes quand ces opérations entraînent le réveil d'une tâche (ready), la préemption de la tâche appelante (preempt) ou son blocage (wait). Ces tests ont été de plus effectués avec une taille de message variable, afin de noter une influence possible de la taille des messages sur les résultats. Les temps obtenus sur les trois systèmes d'exploitation (ChorusOS est en cours d'évaluation) sont compatibles avec les contraintes des applications industrielles EDF. Les temps présentés sont un sous-ensemble de ceux obtenus puisque le banc de tests complet fournit une centaine de temps différents par OS.

4 Bilan et perspectives

Du point de vue des performances, les systèmes d'exploitation évalués respectent les contraintes des applications d'EDF visées. En construisant une chaîne de traitement du capteur au niveau 0 jusqu'au poste opérateur en niveau 2, nous observons que les temps passés en gestion système (envois/réceptions en files, commutations de tâches, communications réseaux avec passage par les sockets, utilisation de mémoire partagée et de sémaphores) quelle que soit la plate-forme ne sont pas prohibitifs. Ceci représente moins de 10% de la contrainte de 500ms de bout en bout, contrainte correspondant à une application type définie en figure 1. Notre étude mérite d'être poursuivie pour une évaluation plus fine des plates-formes ci-dessus. En effet, nous n'avons testé que l'API dans un mode idéal, c'est à dire sans aucune perturbation due à une application qui s'exécute en concurrence. Il faudrait donc faire des mesures avec des machines chargées par une application perturbatrice (avec des entrées sorties capteurs, actionneurs ou réseau). Il est intéressant de noter qu'en développant ces programmes de mesures à partir des macros de notre banc de test, un seul développement suffira pour tous les systèmes supportés, modulo de simples recompilations. Dans un second temps, il faut évaluer les systèmes d'exploitation avec des applications possédant

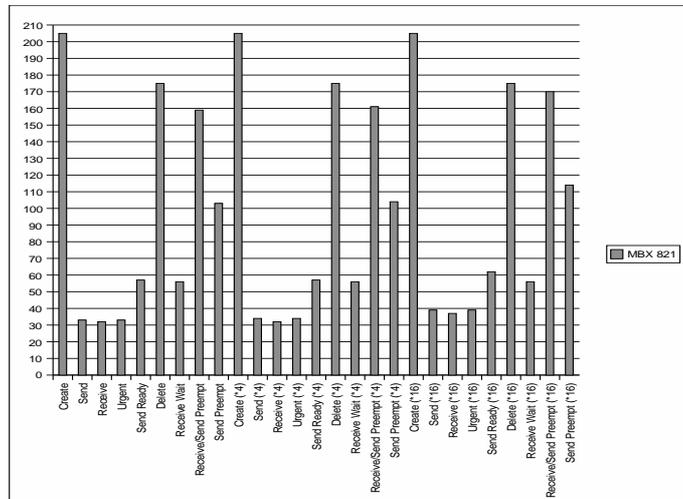


FIG. 2 – Résultats de l'API file de messages sous VxWorks

les profils EDF, par exemple avec différents régimes : régime permanent avec ou sans occurrence d'avalanches. Ceci sera fait dans la suite de l'étude, pour l'évaluation d'ORBs temps réel.

Les travaux qui vont suivre ont pour but d'évaluer les solutions de répartition disponibles dans le contexte de ces systèmes d'exploitation temps réel. Nous traiterons les ORBs temps réel est en cours, et en particulier CORBA [10] et son extension temps réel [9], sur la technologie Java et son extension temps réel [7][8] et sur DCOM [11]. Nous choisirons alors une solution et l'évaluerons au moyen d'un prototype représentatif d'une application d'informatique industrielle EDF.

Bibliographie

1. L. Bacon, E. Gressier-Soudan. *Etude des contraintes utilisateur pour les systèmes d'exploitation temps réel - Questionnaire*. HP-32/98/095/A. Septembre 1998.
2. L. Bacon, E. Gressier-Soudan. *SETR : Systèmes d'Exploitation Temps Réel - Etude des contraintes utilisateurs Synthèse entretiens*. HP-32/98/079/A. 1999.
3. L. Bacon, E. Becquet, E. Gressier-Soudan. *SETR : Systèmes d'Exploitation Temps Réel - Etude des contraintes utilisateurs - Critères d'évaluation*. HP-32/98/078/A. Mars 1999.
4. L. Bacon, E. Becquet, E. Gressier-Soudan. *Etude comparative de micro-noyaux temps réel du commerce*. 7ème Séminaire du Laboratoire d'Ingénierie de la Sécurité de fonctionnement (LIS). Bordeaux. 16-17 Mars 1999.
5. L. Bacon, E. Becquet. *SETR : Systèmes d'Exploitation Temps Réel - Description de EDFBench*. HP-32/99/035/A. Juillet 1999.
6. L. McVoy, C. Staelin. *Lmbench : Portable tools for performance analysis*. Dans Proc. Winter 1996 USENIX, San Diego, CA, pp. 279-284. Janvier 1996.
7. J consortium. *Draft International J Consortium Specification*. Septembre 1999.
8. NIST. *Report from the Requirements Group for Real-Time Extensions For the Java Platform*. Septembre 1999.
9. Object Management Group. *Real-Time CORBA*. Joint Revised Submission. Mars 1999.
10. Object Management Group. *The Common Object Request Broker : Architecture and Specification*. Revision 2.3. Juin 1999.
11. R. Grimes. *Professional DCOM Programming, A guide to creating practical applications with Microsoft's Distributed Component Object Model*. WROX Press Limited. 1997.