Département Informatique et Mathématiques Appliquées

Projet Long 2008

ADMINISTRATION AUTONOME DE SERVEURS SUR LA GRILLE AVEC UNE MACHINE VIRTUELLE

Plan de tests

Responsable : Daniel Hagimont - Professeur INPT/ENSEEIHT - Daniel.Hagimont@enseeiht.fr

Co-encadrant : Laurent Broto – Etudiant en thèse à l'UPS - Laurent.Broto@irit.fr

Superviseur industriel : Emmanuel Murzeau - emmanuel.murzeau@airbus.com

Chef de projet : Ezequiel Geremia - ezequiel.geremia@etu.enseeiht.fr

Etudiants :

- Julien Louisy
- Julien Clariond
- Hery Randriamanamihaga
- Ezequiel Geremia
- Mathieu Giorgino

Sommaire

1 Préliminaires

Le plan de tests se scinde en trois parties complémentaires. La première regroupe les tests tendant à vérifier les propriétés du mécanisme de migration. La deuxième est constituée des tests ayant pour but l'étude des performances du mécanisme de migration. La troisième doit valider la stabilité des propriétés du mécanisme de migration lors de son intégration à l'environnement Tune.

Certains tests, notamment ceux concernant la mesure de performances, amènent à analyser les caractéristiques propres migration, e.g. sa durée ou le délai qu'elle introduit dans le temps d'exécution d'un programme. Les mesures de performances sont expérimentales et le protocole doit donc s'organiser autour de deux cas, le cas témoin et le cas contenant l'expérience elle-même qui seront numérotés *1* et *2* respectivement.

Les notations suivantes seront utilisées dans ce document :

- *dom0* : système d'exploitation hôte associé à Xen s'exécutant sur une machine physique ;
- *domU* : système d'exploitation porté sur Xen s'exécutant sur une machine virtuelle.

Le prérequis *P#* suivant est commun à tous les tests de validation :

(P#): Deux machines physiques sur un même réseau local hébergent respectivement les *dom0s* source et destination, notés respectivement *dom0_source* et *dom0_destination*. Un *domU* s'exécute sur le *dom0* de départ.

2 Tests de validation

2.1 Test des propriétés du mécanisme de migration

T1	*Énoncé du test*	Migration d'un *domU* entre deux *dom0*s.
	Prérequis	***(P#)***
	Déroulement du test	Exécuter la commande `# watch -n1 xm list` sur le *dom0_source* et le *dom0_destination*. Exécuter la commande `# migrate --live <machine virtuelle> <hôte de destination>` sur le dom0_source.

	Résultat attendu	Le *domU* doit s'exécuter sur le *dom0_destination*. Aucun *domU* ne doit s'exécuter sur le *dom0_source*.
	Référence de l'exigence	EF1

T2	*Énoncé du test*	Migration d'un serveur de streaming s'exécutant sur un *domU* entre deux *dom0*s. ***(P#)***
	Prérequis	Un serveur de streaming s'exécute sur un *domU* hébergé sur le *dom0_source*. Un client est connecté sur le serveur de streaming.
	Déroulement du test	Le client accède au contenu du fichier de streaming pendant la migration du *domU* du *dom0_source* vers le *dom0_destination*.
	Résultat attendu	Le client ne doit pas être déconnecté.
	Référence de l'exigence	EF1, EF2

T3	*Énoncé du test*	Programme synchrone écrivant dans un fichier durant s'exécutant sur un *domU* subissant une migration entre deux *dom0*. ***(P#)***
	Prérequis	Le programme $\mathcal{P}$ s'exécute sur *domU*. Le programme $\mathcal{P}$ doit s'exécuter sur une durée finie supérieure à la durée d'une migration.
	Déroulement du test	$\mathcal{P}$ est un programme d'écriture dans un fichier. Toutes les secondes, $\mathcal{P}$ écrit la valeur d'un compteur initialisé à *0* et incrémenté à chaque écriture. *Cas 1 :* le programme $\mathcal{P}$ est entièrement exécuté sur un *domU* témoin s'exécutant sur un même *dom0*. Le fichier de sortie est noté *F1*. *Cas 2 :* le programme $\mathcal{P}$ est exécuté sur un *domU* qui subit une migration du *dom0_source* vers le *dom0_destination* avant la fin de l'exécution de $\mathcal{P}$. Le fichier de sortie est noté *F2*.
	Résultat attendu	Les fichiers *F1* et *F2* sont identiques.
	Référence de l'exigence	EF1, EF3

T4	*Énoncé du test*	Migration d'un domU d'un *dom0_source* vers un *dom0_destination* à différents niveaux de charge processeur du *dom0_source*.
	Prérequis	***(P#)***

	Déroulement du test	Le programme $\mathcal{P}_k$ décrit l'ensemble des applications suivantes : $\mathcal{P}_1$ est un serveur de streaming, $\mathcal{P}_2$ est un compteur synchrone écrivant ses résultats dans un fichier. *Cas 1 :* seul le programme $\mathcal{P}_k$ s'exécute sur le *domU* hébergé par le *dom0_source*. Une migration du *domU* est opérée vers le *dom0_destination*. *Cas 2 :* le programme $\mathcal{P}_k$ s'exécute sur le *domU* hébergé par le *dom0_source*. Le *dom0_source* héberge des composants logiciels consommateurs de temps processeur à une hauteur minimale qui vaut respectivement 50%, 70% et 90%. Une migration du *domU* est opérée vers le *dom0_destination*.
	Résultat attendu	Les critères de réussite dépendent du $\mathcal{P}_k$ choisi : pour $\mathcal{P}_1$ aucun client connecté au serveur ne doit être déconnecté, pour $\mathcal{P}_2$ les fichiers produits dans chacun des deux cas doivent être identiques.
	Référence de l'exigence	EF1, EF2, EF3, EF4

2.2 Tests de performances du processus de migration

T5	*Énoncé du test*	Mesure comparative de la durée de la migration d'un *domU*. *(P#)*
	Prérequis	Le programme $\mathcal{T}$ capable d'accéder au temps processeur d'un *domU*. Le programme $\mathcal{P}$ capable d'effectuer des appels RMI successifs depuis *dom0_source* vers le *dom0_destination*.
	Déroulement du test	*Cas 1 :* mesure de la durée de *n* appels RMI (Remote Method Invocation) depuis le *dom0_source* vers le *dom0_destination* utilisant $\mathcal{P}$, par deux appels à $\mathcal{T}$, l'un avant l'appel de $\mathcal{P}$ l'autre à son retour. *Cas 2 :* mesure de la durée totale de la migration d'un *domU* du *dom0_source* vers le *dom0_destination* suivie de la migration du même *domU* du *dom0_destination* vers le *dom0_source*. La durée d'escale du *domU* sur le *dom0_destination* doit être minimisé. De même que dans le premier cas, $\mathcal{T}$ sera appelé à deux reprises sur le *dom0_source* lors du départ du *domU* puis lors de son retour.
	Résultat attendu	La durée de deux migrations successive exprimée en fonction de la durée d'un appel RMI.
	Référence de l'exigence	EF1, EF5

T6	*Énoncé du test*	Calcul du délai introduit par la migration dans l'exécution d'un programme s'exécutant sur un *domU*.
	Prérequis	*(P#)*, T5 Le programme $\mathcal{T}$ capable d'accéder au temps processeur d'un *domU*. Le programme $\mathcal{P}$ doit s'exécuter sur une durée finie supérieure à la durée de deux migration successives.
	Déroulement du test	*Cas 1 :* mesure de la durée d'exécution du programme $\mathcal{P}$ sur le *dom0_source*, par deux appels à $\mathcal{T}$, l'un avant l'appel de $\mathcal{P}$ l'autre à son retour. *Cas 2 :* mesure de la durée d'exécution de $\mathcal{P}$ s'exécutant sur un *domU*, qui subit deux migrations successives, l'une du *dom0_source* vers le *dom0_destination* et l'autre du *dom0_destination* vers le *dom0_source*. La durée d'escale du *domU* sur le *dom0_destination* doit être minimisé. De même que dans le premier cas, $\mathcal{T}$ sera appelé à deux reprises sur le *domU* lors de l'appel de $\mathcal{P}$ puis à la fin de son exécution.
	Résultat attendu	Le délai introduit par la migration exprimé en fonction de la durée d'une migration.
	Référence de l'exigence	EF1, EF6

T7	*Énoncé du test*	Mesure du durée de l'interruption de service produite par une migration de *domU* et doit s'exécuter sur une durée supérieure à la durée d'une migration.
	Prérequis	*(P#)* Le commande `ping` paramétrable en fréquence. Le programme $\mathcal{P}$ qui utilise la commande `ping` pour solliciter un troisième *dom0* noté *dom0_référence*.
	Déroulement du test	$\mathcal{P}$ s'exécute sur un *domU* hébergé sur le *dom0_source*. Le *domU* en question subit une migration de *dom0_source* vers *dom0_destination*. Pendant toute la durée de son exécution $\mathcal{P}$ sollicite dom0_référence de manière synchrone grâce à la commande `ping`. La durée de l'interruption de service introduit par une migration de *domU* est la différence entre le nombre de requêtes `ping` envoyées par et le nombre de requêtes `ping` reçues par le *dom0_référence*.
	Résultat attendu	La durée de l'interruption de service introduit par une migration de *domU* sera exprimée en fonction de la durée d'une migration.

	Référence de l'exigence	EF1, EF7

2.3 Test de l'utilisation du mécanisme de migration dans Tune

T8	*Énoncé du test*	
	Prérequis	
	Déroulement du test	Pour chaque exigence, refaire chacun des tests de satisfaction des exigences EF1 à EF8 dans une architecture autonome administrée par Tune.
	Résultat attendu	Chacun des tests effectué pour chaque exigence devra se dérouler correctement.
	Référence de l'exigence	EF1, EF8

T9	*Énoncé du test*	
	Prérequis	
	Déroulement du test	Pour chaque exigence, refaire chacun des tests de satisfaction des exigences EF1 à EF8 dans une architecture autonome administrée par Tune.
	Résultat attendu	Chacun des tests effectué pour chaque exigence devra se dérouler correctement.
	Référence de l'exigence	EF1, EF9

T10	*Énoncé du test*	
	Prérequis	
	Déroulement du test	Pour chaque exigence, refaire chacun des tests de satisfaction des exigences EF1 à EF8 dans une architecture autonome administrée par Tune.
	Résultat attendu	Chacun des tests effectué pour chaque exigence devra se dérouler correctement.
	Référence de l'exigence	EF1, EF10

T11	*Énoncé du test*	
	Prérequis	

	Déroulement du test	Pour chaque exigence, refaire chacun des tests de satisfaction des exigences EF1 à EF8 dans une architecture autonome administrée par Tune.
	Résultat attendu	Chacun des tests effectué pour chaque exigence devra se dérouler correctement.
	Référence de l'exigence	EF1, EF11

T12	*Énoncé du test*	
	Prérequis	Le programme de test sera une boucle s'exécutant continuellement pendant une durée prédéterminée.
	Déroulement du test	Un certain nombre de machines virtuelles hébergeant le programme de test sont ajoutées.
	Résultat attendu	L'environnement intégré dans Tune doit évoluer dans le sens d'un équilibrage des charges CPU.
	Référence de l'exigence	EF1, EF12

3 Matrice de couverture

La couverture des exigences par les tests est résumée dans la matrice de couverture ci-après. Les points sont situés à l'intersection de la ligne du test considéré et de la colonne de l'exigence qu'il permet de valider.

		Exigences											
		EF1	EF2	EF3	EF4	EF5	EF6	EF7	EF8	EF9	EF10	EF11	EF12
Tests	T1	●											
	T2	●	●										
	T3	●		●									
	T4	●	●	●	●								
	T5	●				●							
	T6	●					●						
	T7	●						●					
	T8	●	●						●				
	T9	●		●						●			
	T10	●			●						●		
	T11	●										●	
	T12	●											●