

IaaS à la sauce Portails

Focus sur

apachecloudstack™

Pierre Aubert

Orange Portails

OF/DMGP/Portails/DOP

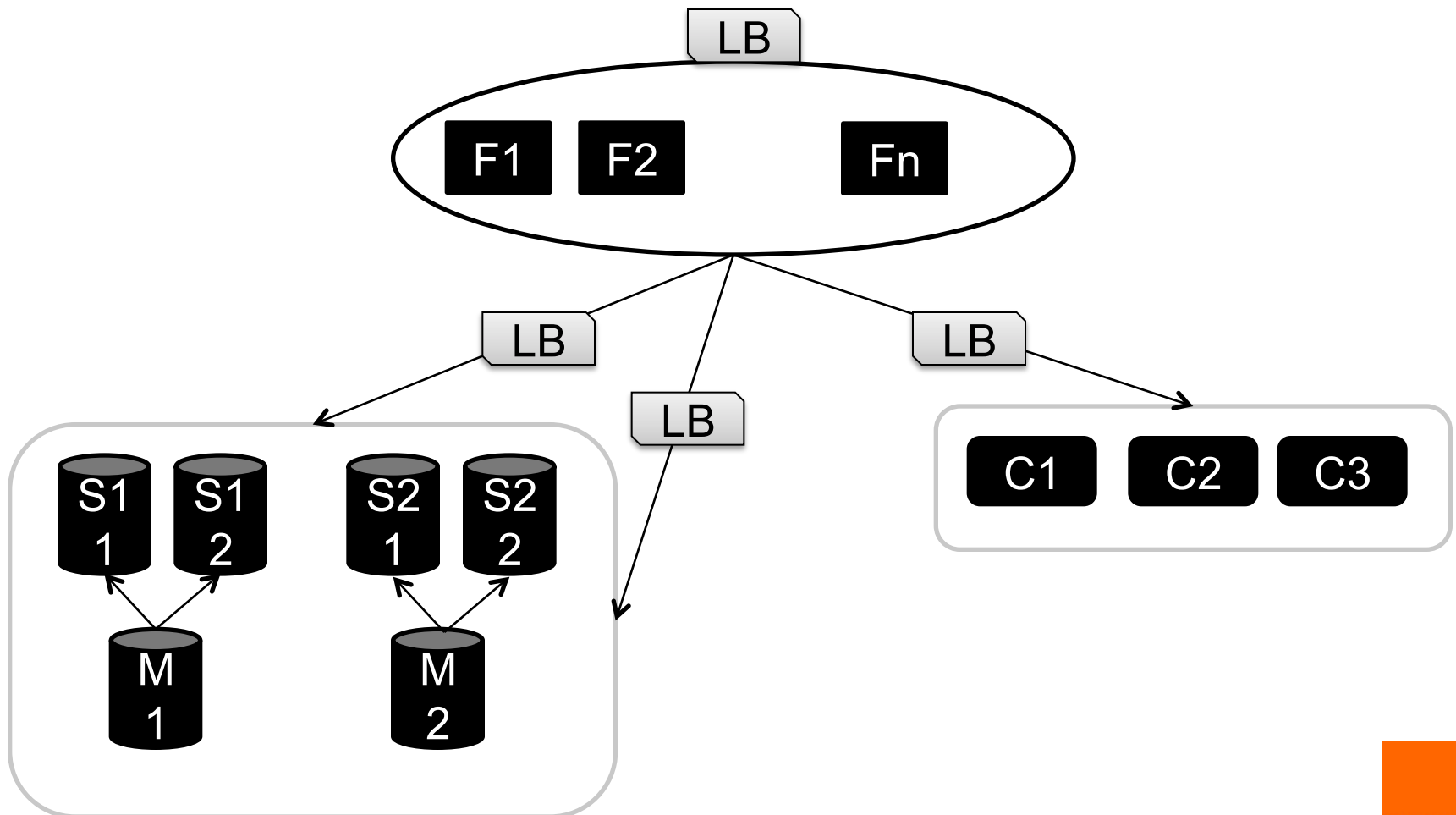
1^{er} Juillet 2013

Notre contexte

- Quelques milliers de serveurs
- Quelques centaines de services et d'applications
- Une historique de startup: le modèle à façon
 - > Des machines physiques choisies et dédiées à l'application
 - > Un OS adapté pour l'application en terme de performance et de QoS
 - > 100% open source
- Mais nous sommes dans Orange donc
 - > Pas d'externalisation des données des clients
 - > Grosse volumétrie: on ne cherche pas à mettre un maximum de VM par host mais au contraire à disperser les VMs du même service sur plusieurs hosts pour assurer la stabilité, la haute disponibilité, la reprise sur erreur ...
 - > et en même temps, les moyens sont faibles et donc l'optimisation des coûts reste un critère principal

Différentes stratégies: les pools

- Le modèle pool de serveurs pour applications de type LAMP par ex
- N-Vhosts par Front; 1 étage de cache, 1 étage de BD



Les pools

- Forces

- > Bonne mutualisation des ressources
- > Capacity planning simple
- > Supervision / Graphage simple
- > Séparation des CPU et des IO
- > Sécurité plus simple car architecture homogène
- > **Simple**

- Faiblesses

- > Isolation des applications (cgroup)
- > Multi Datacenter complexe
- > Evolutions difficiles (comment upgrader un pool?)
- > Architecture relativement figée

Virtualisation

- Choix historique de Xen (version opensource)
 - > KVM mauvais sur les IO (il y a longtemps)
- Modèle IaaS à la main: on place les VMs manuellement sur des hosts
- Forces
 - > On peut mutualiser les patterns par machines (et « boucher » les trous)
 - > Moyenne de 6 VMs par host
 - > On peut migrer les VMs
 - > Bonne maîtrise, degré de confiance élevé
- Faiblesses:
 - > N'a pas changé la structure des coûts de run
 - > La supervision, sécurité etc ... se font aussi bien sur les machines physiques que virtuelles
 - > Reste trop artisanal

Les IaaS

AWS, openNebula et apache CloudStack

- OpenNebula en production depuis 2 ans, héberge
 - > les environnements de développement
 - > les bacs à sable
 - > adapté pour ces patterns
- AWS
 - > utilisé pour apprendre
 - > pour les plateformes de tests et de développement
 - > coûts 100% opex
 - > parfois moins cher que l'interne parfois beaucoup plus cher
- Apache CloudStack
 - > une version 2 déjà en production depuis 2 ans chez Orange (OBS)
 - > Gagne l'étude de marché interne mi-2012
 - entre openstack, nimbula, eucalyptus et open nebula

Apache Cloudstack

- Dans un premier temps, on veut:
 - > installer Cassandra
 - > installer des frontaux élastiques avec du PHP et du C++ derrière des serveurs apaches
 - > dans 2 datacenters avec une QoS au moins égale à l'actuelle i.e. 99,9%
 - > que cela coute significativement moins cher à la cible que dans l'environnement actuel
- Comment on est parti:
 - > Cloudstack 4 (beta), XenServer 6 (beta) sur des machines HP G8 (toute fraiche) avec des cartes Fusion-IO, des hosts en CentOS et des VM en ubuntu
 - > En //, on se pose la question de comment installer C* et les autres applications (notre historique est 100% CFengine 2)
 - On choisit Chef de OpsCode (entre CFengine3, Puppet et Chef) et CFengine3

Apache Cloudstack

ce que l'on a fait

- On a réussi à trouver la bonne combinaison entre
 - > cloudstack/xen server/G8 et fusion-io
 - > quelques patches nécessaires sur cloudstack
- On a installé la base de donnée de cloudstack en bi-site et validé les bascules de site et les reprise
- On tient compte de notre réseau compliqué
- On a déployé les VM cassandra à la main (aie) parce que l'on n'a pas su écrire un allocateur adapté
 - > C* a plusieurs copies de la même donnée mais il faut être sûr que les vm correspondantes ne sont pas sur la même machine physique
- On a prié très fort, fait pleins de réunions de GO/NOGO et on est parti en production (sans toutes les docs, sans expérience, sans)

Bilan après quelques mois

- Cloudstack
 - > Nb de ticket depuis la mise en ligne 0 avec impact
 - > Temps de réponse de déploiement d'une VM (5 minutes)
 - > Automatisation de l'installation d'un serveur physique OS xcp (30 min par serveur)
- Cassandra sur Cloudstack
 - > Nb de ticket depuis la mise en ligne 1 avec impact
 - > Nb de sollicitation de l'astreinte niveau 2: 0
 - > Temps d'écriture : 10x plus rapide, <1ms
 - > QoS: >99,98% malgré les pertes de serveurs, de VM ou de coupures du réseau entre les datacenters

Re-Aie

Cloudstack en panne

- Un incident de production (cause inconnu) nous force à arrêter cloudstack (i.e. le manager)
- Bonne nouvelle: la production fonctionne mais on ne plus ajouter de VM
- En cours: tentative de migration sur la Cloudstack 4.1 réponse bientôt
- Morale:
 - > toujours avoir une plateforme de pre-production pour pouvoir tester son cloud et faire ses upgrades
 - > toujours avoir des développeurs et des sysadmins sous la main qui connaissent le code pour faire des patches si besoin
 - > si pas de support Citrix (ou shapeblue ou autre), pour son support interne, il faut des développeurs et des sysadmins disponibles
 - > Rester Zen coté management

Apache Cloudstack

la suite

- Valider la migration en 4.1
- Déploiement de Hive au dessus de Cassandra
- Déploiement des frontaux (pas élastique puis élastique à base de haproxy)
- L'orage passé:
 - > se reposer la question Cloudstack/OpenStack/OpenNebula/ZZZ
 - > se reposer la question Xen/KVM/LCX/YYYY versus Vmware
 - > se reposer la question de simplifier le réseau
 - VXLAN, VLAN pleins ...
 - Switch/Router Juniper SDN compliant

Bilan technique

- CloudStack

- > c'est complexe et pas encore stable pour la production
- > il faut bien dimensionner la taille de l'équipe support aussi bien en développement qu'aux opérations
 - le code bouge très vite
- > il faut anticiper les migrations et les patchs sur le cloud lui-même
- > on ne peut pas se passer de plateformes de performance et de pre production, pas plus que de swing kit
- > il faut mettre en place de la non-régression sur le cloud via des tests automatiques d'infrastructures
- > il est trop tôt pour valider les autres parties de Cloudstack (stockage hors baies nas ou san (ceph?), LB élastique, ...)
- > Les gains de cout sur la construction des plateformes et sur le run ne sont pas encore prouvés

Bilan humain

- CloudStack, pour votre prochain projet:
 - > Ne pas démarrer avec des bases de données
 - > Privilégier les frontaux et les parties non critiques par exemple la production cachée pour gagner en compétence et en confiance
 - > Le savoir faire n'existe pas et il se construit au fil de l'eau
 - Par ex: pas de documentation d'exploitation pour un niveau 1
 - > Ne pas se faire peur à la première difficulté et bien préparer les plans d'actions en avance de phase
 - > Accompagner les équipes au changement et à la prise de risque. Il ne faut pas négliger l'impact du changement d'esprit que la mise en place d'un IaaS implique. Ca n'est pas un projet comme les autres.

On est en PRODUCTION et on continue