

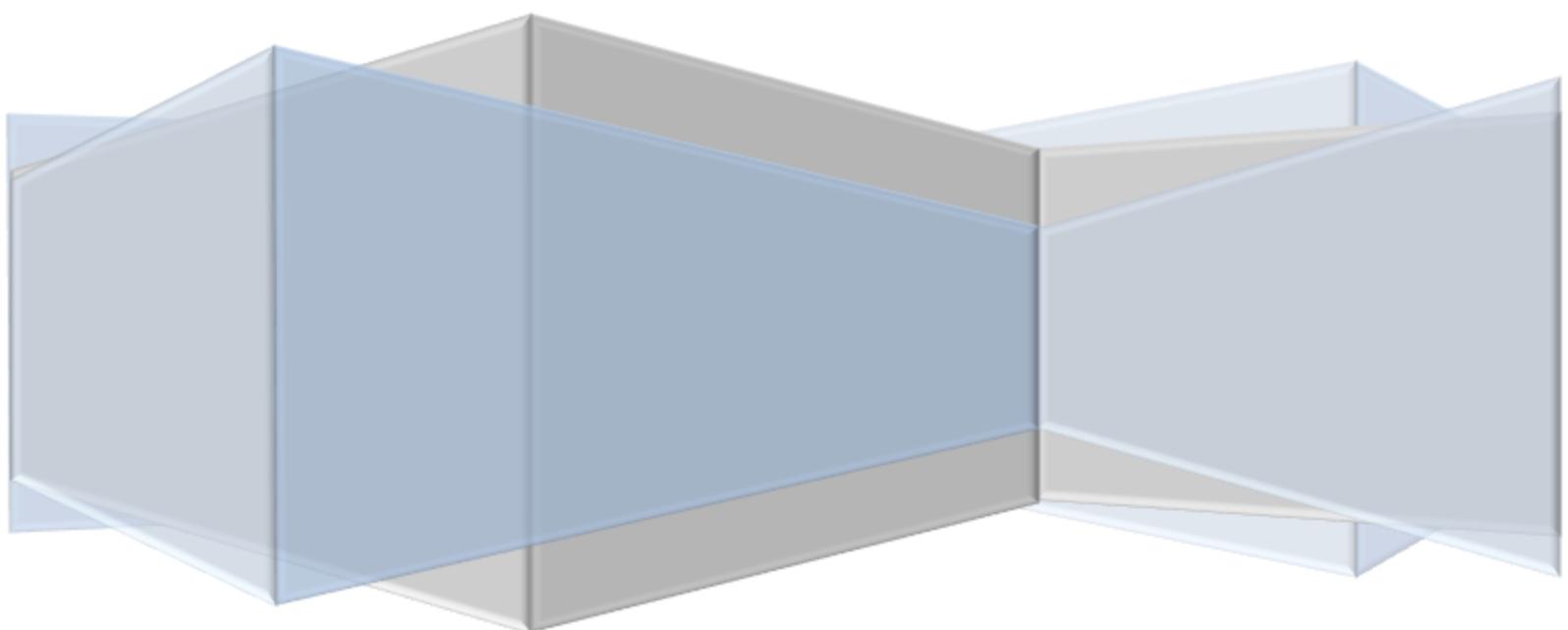


Rapport de Projet 3A:

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

Supervisé par Lucas NUSSBAUM



Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

Sommaire

Introduction :	2
I. Terminologie :	3
II. Contexte :	3
III. Problématique et Besoin :	4
1. Cas d'usage du Grid :	4
2. Problématiques :	4
IV. Solution et Challenge :	5
1. Solution :	5
2. Challenge :	6
3. Script Final :	6
V. Expériences :	7
1. Déploiement par Site Vs. Déploiement par Cluster :	7
a. Démarche expérimentale :	7
b. Résultats :	8
c. Analyse :	9
2. Quels Clusters sont les plus souvent disponibles :	9
a. Démarche expérimentale :	10
b. Résultats :	10
c. Analyse :	12
3. Y a-t-il des Cluster ou des Sites qui sont distingués :	12
a. Démarche expérimentale :	12
b. Résultats :	12
c. Analyse :	14
4. Corrélation entre nombre de Nodes et la performance d'un déploiement :	15
a. Démarche expérimentale :	15
b. Résultats :	15
c. Résultats :	20
Conclusion :	22

Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

Introduction :

Le Grid Computing ou, en Français, la Grille Informatique est un terme qui est apparu au début des années 1990 comme une métaphore qui compare la facilité d'accès à une puissance de calcul informatique à l'accès à la grille électrique. Plus tard ce concept a été présenté plus en détail dans le travail d'Ian Foster et Carl Kesselman : « The Grid: Blueprint for a new computing infrastructure » en 2004.

Le Grid est donc défini comme étant une infrastructure virtuelle constituée d'un ensemble de ressources informatiques potentiellement partagées, distribuées, hétérogènes, délocalisées et autonomes.

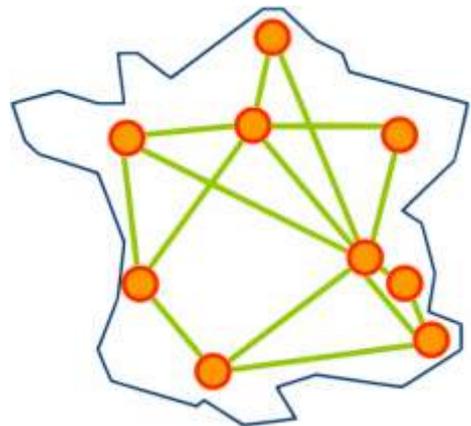
Le Grid est en effet une infrastructure, c'est-à-dire des équipements techniques d'ordres matériel et logiciel. Cette infrastructure est qualifiée de virtuelle car les relations entre les entités qui la composent n'existent pas sur le plan matériel mais d'un point de vue logique.

Elle garantit des qualités de service non triviales, c'est-à-dire qu'elle se distingue des autres infrastructures par son aptitude à répondre adéquatement à des exigences d'accessibilité, de disponibilité et de fiabilité, compte tenu de la puissance de calcul ou de stockage qu'elle peut fournir.

Afin de profiter de cette technologie, l'INRIA et le CNRS ont fondé le Grid'5000, un instrument scientifique pour l'étude des systèmes distribués et parallèles à grande échelle. Il a comme but de fournir une plateforme expérimentale hautement configurable et contrôlable.

Le but initial était d'atteindre 5000 processeurs (en 2003), ensuite ce but a été reformulé en 5000 cœurs au lieu de 5000 processeurs. Ce qui a été atteint durant l'hiver 2008-2009.

L'infrastructure du Grid'5000 est distribuée géographiquement sur 9 sites en France. Les sites sont interconnectés grâce au réseau de recherche RENATER avec un débit de 10Gb/s.



Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

I. Terminologie :

Site : Un Site fait référence à la zone géographique ou le laboratoire qui hôte les machines. La plupart des sites du Grid'5000 hôte plus qu'un Cluster.

Historiquement, les machines du Grid'5000 ont été acquises d'une manière incrémentale en formant un Cluster à chaque acquisition.

Cluster : Un Cluster est un ensemble de machines présentant des propriétés homogènes.

Un Cluster est connecté à une architecture réseau particulière et est présent sur un Site particulier.

Enfin tous les Cluster d'un même Site partagent la même infrastructure réseau.

Node : Tout ordinateur faisant partie d'un Cluster est appelé un Node. On peut donc définir deux types de Nodes :

- Computing Nodes (nœuds de calcul) : est l'élément de base d'un Cluster. Il hôte les calculs.
- Service Nodes (nœuds de service) : ces nœuds ne sont pas destinés à faire tourner les applications des utilisateurs mais à maintenir l'infrastructure du Grid.

Chaque Node peut offrir plusieurs ressources aux utilisateurs. Ces ressources élémentaires sont appelée Cores (Cœurs).

II. Contexte :

Dans le cadre de mon projet de recherche en 3^{ème} année de mon parcours au Mines de Nancy j'ai été amené à choisir un sujet parmi différents sujets que notre département « Systèmes et Information » nous a proposé. Revenant à peine d'une année de césure effectuée autour des technologies de Cloud Computing, mon choix était évident. Voulant compléter mes connaissances des méthodes de calculs à haute performance, j'ai donc choisit de travailler avec M. Lucas NUSSBAUM et son équipe AlGorille, constituée de professeurs chercheurs, d'ingénieurs et d'étudiants doctorants, pour optimiser l'utilisation du Grid'5000 à grande échelle.

L'objectif principal du projet de recherche est d'introduire l'étudiant à ce domaine pour qu'il découvre son fonctionnement, son organisation ainsi que ses buts et motivations. En portant l'habit d'un apprenti chercheur, l'étudiant doit poser une thèse de départ, mener une démarche de recherche scientifique et enfin présenter et discuter ses résultats.

Ayant M. Lucas NUSSBAUM comme superviseur directe au LORIA et M. Pierre-Etienne MOREAU comme interlocuteur au sein des Mines de Nancy, je me suis organisé pour garder un contact continu avec M. NUSSBAUM en essayant de travailler le plus souvent au LORIA à proximité de l'équipe AlGorille, et en contactant M. MOREAU qu'en cas de difficulté.

Bien que je tenais à être présent le plus possible au LORIA, ma collaboration avec l'équipe de recherche était assez réduite et c'était avec M. NUSSBAUM que j'échangeais le plus souvent.

Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

Finalement ce projet est conclu par un rapport et une présentation orale face aux professeurs et à mes camarades de classe, synthétisant la problématique, les démarches adoptés et les résultats obtenus.

III. Problématique et Besoin :

1. Cas d'usage du Grid :

De nos jours les recherches scientifiques nécessitent de plus en plus de parallélisme et de puissance de calcul. Le concept du Grid computing est né pour répondre à ces besoins. Plusieurs domaines de recherches y font alors appel et dépendent largement de son efficacité, à titre d'exemple on peut citer les cas d'utilisation suivants :

- e-Science :
 - Un bio-chimiste exploite 10000 ordinateurs pour tester 10000 composés chimiques en une heure.
 - 1000 physiciens dispersés géographiquement combinent leurs ressources informatiques pour analyser des péta-octets de données.
 - Des scientifiques du climat visualisent et annotent des données issues de simulations nécessitant l'analyse de téra-octets de données.
- e-Engineering :
 - Plusieurs ingénieurs de plusieurs équipes au sein d'une ou de plusieurs sociétés collaborent à la conception d'un satellite.
- e-Business :
 - Une compagnie d'assurance analyse des données de sources multiples (plusieurs bases de données) afin de détecter les fraudes.
- Couplage de plusieurs codes de calcul : Fluide-fluide, fluide-structure, structure-thermo, fluide-acoustique-vibration.
- L'exécution simultanée d'un même code de calcul mais avec des paramètres différents :
 - Trouver le meilleur design selon un certain nombre de critères.
- Couplage avec des algorithmes d'optimisation :
 - Algorithmes génétiques.

Tous ces cas d'usage nécessitent une nouvelle approche différente de l'approche traditionnelle qui faisait appel au mieux à un supercalculateur.

2. Problématiques :

Grid'5000 intègre une couche Middleware qui assure la communication entre les différentes machines et les différents sites. Cette couche offre également des services permettant de réserver des nœuds du Grid (Job Reservation) pour ensuite déployer dessus des environnements d'expérimentation (image deployment). Eventuellement, ces environnements permettront plus tard d'exécuter les calculs nécessaires pour satisfaire les besoins évoqués ci-haut.

Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

La plateforme Grid'5000 est partagée par plusieurs équipes de recherche en France, ce qui la rend fortement sollicitée, et rend la première règle d'usage du Grid'5000 plus dure à respecter : « éviter de monopoliser les ressources et utiliser la plateforme de la manière la plus efficiente possible. »

On soulève rapidement donc les problématiques suivantes au sein des utilisateurs du Grid'5000:

- **S'assurer d'obtenir les ressources souhaitées pour ses expériences en minimisant les échecs de déploiement.**
- **Minimiser le temps de déploiement des environnements de travail pour réduire la monopolisation des ressources.**
- **Posséder un Script qui automatise la réservation de ressources et le déploiement d'environnement de travail en respectant les deux critères ci-dessus.**

Il faut noter ici que mon travail concernerait les expériences à large échelle qui nécessitent l'emploi d'un grand nombre de Nodes du Grid éventuellement distribués géographiquement.

Ma mission était donc d'étudier la démarche de réservation de telles ressources et le déploiement des environnements d'expérimentation et de conclure sur la performance des différents sites du Grid. Enfin, si possible, présenter des recommandations pour optimiser le déploiement des environnements d'expérimentation à grande échelle.

IV. Solution et Challenge :

1. Solution :

Afin de réaliser cette étude on a décidé d'utiliser des scripts codés en Ruby. Ces scripts permettent de tester la performance du déploiement sur les Nodes du Grid'5000.

Plusieurs itérations de ces scripts de test ont été employées basés sur un script inclus dans les tutoriels des APIs du Grid'5000. Le script utilise la librairie Restfully du langage Ruby destinée à l'utilisation des APIs RESTful, ce qui est exactement le type d'API qu'expose Grid'5000.

Restfully est une enveloppe légère qui s'impose au-dessus de la librairie RestClient pour abstraire les appels HTTP et se charger d'analyser les réponses.

Le script initial applique l'algorithme suivant :

- Découvrir les sites du Grid'5000.
- Obtenir l'état de chaque site.
- Soumettre un job sur chaque site ayant le nombre voulu de ressources libres.
- Attendre que les jobs commencent.

Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

- Lancer un déploiement sur tous les sites où le job est en état de marche.
- Attendre que tous les déploiements soient finis.
- Se connecter aux nœuds déployés pour lancer une commande, si souhaité.

Toutes les étapes d'attente sont bien sur soumises à un timeout pour ne pas attendre indéfiniment.

En ayant ce script comme point de départ, j'ai essayé donc de le modifier afin d'aboutir au meilleur algorithme de test des performances du Grid.

2. Challenge :

Le premier challenge était bien sûr de faire face à un langage de programmation qui m'était inconnu, Ruby. Il a fallu donc me mettre rapidement à niveau. Pour cela je me suis basé sur les différentes ressources que je pouvais trouver en ligne, mais aussi sur la première partie de l'édition 2010 de « Metaprogramming Ruby » par Paolo PEROTTA.

Prenant conscience du nombre d'expériences qu'il fallait lancer pour obtenir des résultats cohérents et tirer les bonnes conclusions sur la performance du Grid, le deuxième challenge était donc d'automatiser les tests le plus possible.

Il fallait aussi que je me familiarise avec l'architecture et le fonctionnement de Grid'5000 et que j'assimile la technologie du Grid Computing. Ce qui était après tout le but ultime de ce projet.

Un dernier challenge moins grand était de bien repérer les créneaux pendant lesquels je pouvais lancer ces expériences à grande échelle sans déranger les utilisateurs du Grid. Heureusement le réseau semblait assez libre durant les Mardis, mes journées réservées pour ce projet, mais aussi en lançant les séries d'expériences en soirées.

3. Algorithme Final :

J'ai tenu à améliorer les scripts de test de manière à lancer des séries d'expérience et de sauvegarder les données obtenues (temps de déploiement, nombre de ressources obtenues, nombre d'échecs...) pour pouvoir les utiliser plus facilement plus tard.

Au final j'ai obtenu un script pour chaque site, chaque script lance une série de tests de déploiements pour chaque site. Les sources de ces scripts sont sur <https://github.com/marouen19/Grid5000>

L'algorithme final :

- Se connecter au Grid'5000 en utilisant une clé publique.
- On utilise une boucle While, qui permet de terminer le script si on n'arrive pas à obtenir les ressources souhaitées E fois (E=3) :
 - Découvrir l'état du site (chaque script est destiné à un Site)
 - Soumettre un Job sur le site demandant un nombre initial N de Nodes.
 - Si on ne peut pas obtenir N Nodes sur tous les Clusters du Site on incrémente E (initialement égale à 0) de +1 et on recommence la boucle, Sinon :

Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

- On lance une seule commande de déploiement pour l'ensemble des Nodes assignés.
 - On relève le temps qu'a pris le déploiement ainsi que le nombre d'échecs.
 - On réinitialise les Nodes assignés.
 - On lance une commande de déploiement par Cluster.
 - On relève le temps que chaque déploiement a pris ainsi que le nombre d'échecs.
 - On affiche les résultats des deux types de déploiement effectués ci-dessus.
 - On multiplie N par 2 ($N=N*2$)
- Le script se termine quand on atteint N tel qu'aucun Cluster n'a N Nodes libres. On vérifie cela E fois en laissant 4 minutes entre chaque vérification au cas où des ressources se libèrent entre temps.

On essayera donc de collecter les résultats de plusieurs tests utilisant des scripts similaires à celui détaillé ci-dessus pour répondre à des questions sur la performance des différents sites du Grid'5000 face à une exploitation à grande échelle.

V. Expériences :

Nous nous sommes posé une série de questions auxquelles on va essayer de répondre en suivant des démarches scientifiques particulières et en analysant les résultats obtenus. Le souci global étant la performance du Grid'5000, on définit cette performance en fonction de deux paramètres :

- Le taux d'échec : les déploiements lancés sur les Nodes du Grid peuvent ne pas aboutir, j'ai donc veillé à garder un œil sur les échecs de déploiement éventuels afin de détecter d'éventuelles faiblesses au niveau de quelques Sites ou Clusters.
- Le temps de déploiement : ce temps est calculé en secondes, entre le moment où une commande de déploiement est exécutée et le moment où on vérifie que l'ensemble de déploiements se terminent (que cela soit pour les Nodes d'un même site ou les Nodes d'un même Cluster)

1. Déploiement par Site Vs. Déploiement par Cluster :

Après avoir commencé mes premiers tests en utilisant le script initial de déploiement à grande échelle j'ai vite remarqué que quelques déploiements échouaient complètement à cause d'un échec au niveau d'un Cluster. On a donc pensé à effectuer des déploiements par Cluster afin de découvrir la performance des différents Clusters du Grid'5000 et comparer la performance des déploiements par Cluster à des déploiements par Site.

a. Démarche expérimentale :

Afin d'avoir des résultats cohérents j'ai veillé à utiliser le même Job et donc exactement les mêmes Nodes pour un déploiement par Site ensuite pour un déploiement par Cluster. La démarche était donc la suivante :

- Réserver un job avec un Nombre N de Nodes.
- Effectuer un déploiement par Site.
- Relever le temps total et le pourcentage d'échec.
- Réinitialiser les Nodes.

Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

- Attendre 2 minutes, le temps que les Nodes reprennent un fonctionnement naturel.
- Effectuer un déploiement par Cluster.
- Relever le temps de déploiement par Cluster et le pourcentage d'échec.
- Enregistrer les résultats et tracer des graphes comparatifs.

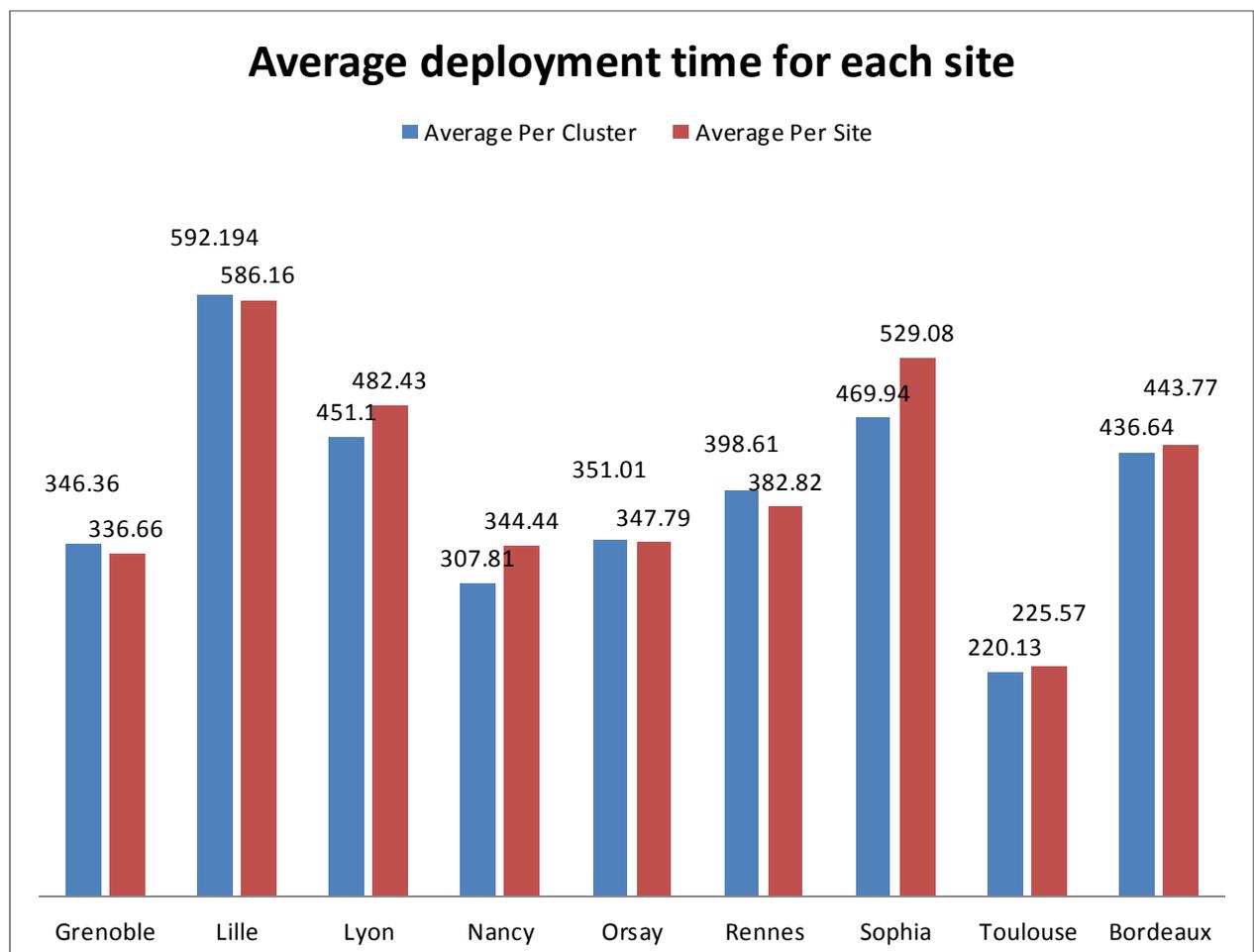
b. Résultats :

Les graphes ci-dessous comparent les performances des déploiements par Site face aux déploiements par Cluster.

Il faut noter tout d'abord que la marge d'erreur des temps de déploiement calculés est de +/- 3 secondes.

On note aussi que ces moyennes sont obtenues sur une moyenne de 20 expériences par site.

Le graphe ci-dessous compare donc la moyenne des temps de déploiement par Site et par Cluster pour chaque site du Grid'5000.



Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

c. Analyse :

Dans le tableau suivant on catégorise les résultats et on essaye d'expliquer quelques cas particuliers :

Résultat	Sites concernés	Constatations
Le déploiement par Site est plus rapide	Lille Rennes Grenoble	Sur ces trois sites on constate qu'il y a des clusters en particulier qui ont des performances très inférieures au reste du site (i.e. Chuque et Edel) ou des Clusters qui ont des performances très supérieures au reste des Clusters (i.e. Parapide) Donc en général les performances des Cluster de chacun de ces sites ne semblent pas homogènes.
Le déploiement par Cluster est plus rapide	Sophia Nancy Lyon Bordeaux	A l'exception de Helios sur Sophia les Clusters des différents sites semblent avoir des performances similaires.
Les deux méthodes de déploiements ont des résultats similaires	Orsay Toulouse	La caractéristique de ces deux sites est que pendant la majorité des expériences un seul Cluster était disponible sur chaque site.

Comme « règle » initiale on constate que sur les Sites où un Cluster possède des performances plus inférieures au reste des Clusters, ce Cluster se transforme en maillon faible et affecte la performance moyenne des déploiements par Cluster et rend un déploiement par Site plus intéressant, si on arrive à obtenir l'ensemble des ressources souhaitées sur les autres Clusters « normaux » du Site.

Quant à la fréquence d'échec, on déduit tout de suite qu'un déploiement par Cluster est le choix à adopter, puisque les échecs au cours d'un déploiement par Site résultent en 0 Nodes utilisables, alors que lors d'un déploiement par Cluster, si un Cluster échoue l'utilisateur peut toujours finir avec quelques environnements de travail déployés sur le reste des Clusters du Site.

2. La disponibilité des Clusters :

On tente ici d'identifier les Clusters qui offrent le plus de Nodes libres. Puisque nos expériences ont été réalisées sur des journées différentes on peut utiliser les proportions de Nodes allouées à nos Jobs comme une moyenne sur la disponibilité de chaque Cluster.

Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

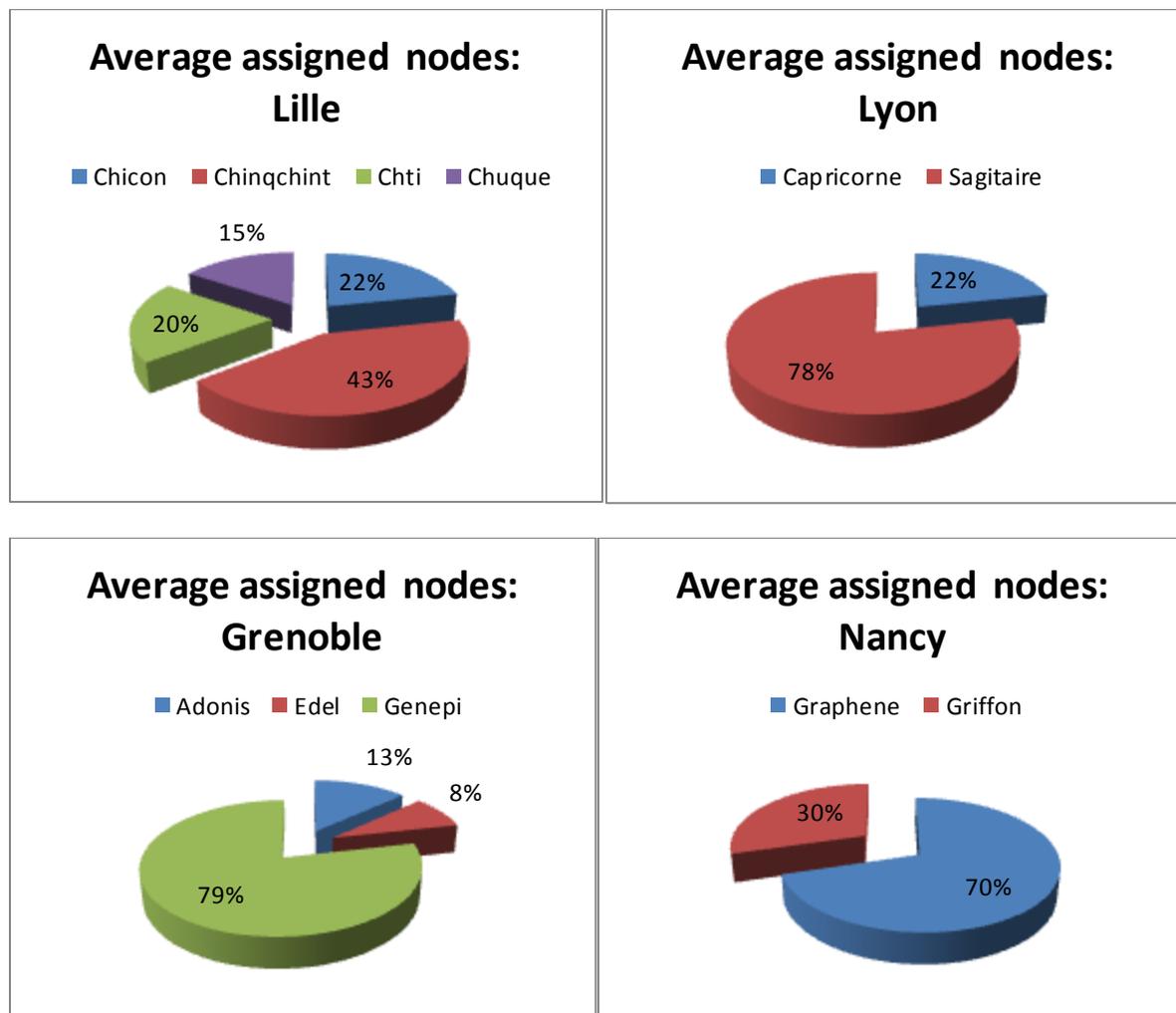
Marouen ZELLEG

a. Démarche expérimentale :

En demandant un nombre N de Nodes quand on soumet un Job, on laisse à la plateforme Grid'5000 la tâche de choisir quels Nodes nous associer en fonction des disponibilités des Clusters. On note donc à chaque fois le nombre de Nodes de chaque Cluster qui nous ont été alloué pour enfin effectuer une moyenne.

b. Résultats :

Les graphes suivants représentent en moyenne la composition des ressources allouées sur chaque Site.

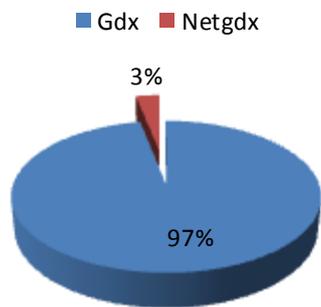


Rapport de Projet 3A

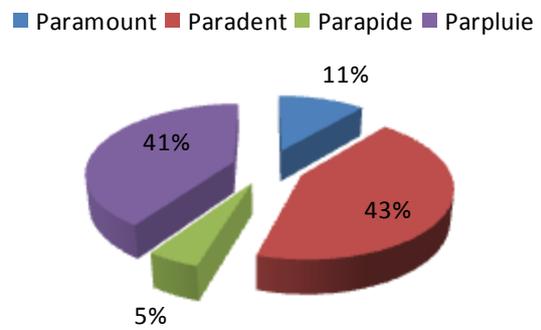
Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

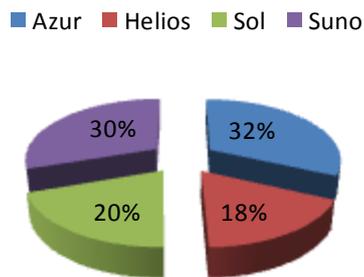
Average assigned nodes: Orsay



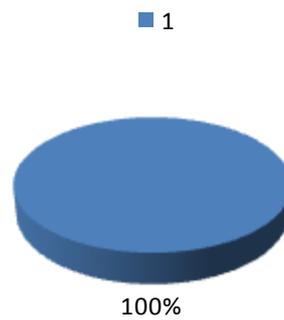
Average assigned nodes: Rennes



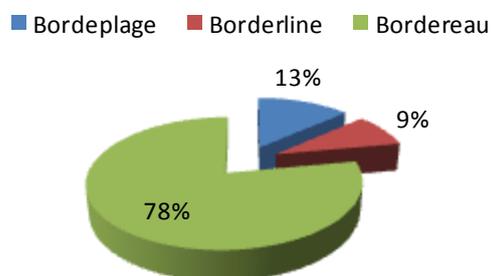
Average assigned nodes: Sophia



Average assigned nodes: Toulouse



Average assigned nodes: Bordeaux



Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

c. Analyse :

Comme les Sites du Grid présentent des propriétés hétérogènes, on note ici les particularités de chaque Site :

Nancy : Si j'ai réussi à obtenir plus de Nodes sur Graphene c'est parce que le Cluster venait d'être installé et que personne ne l'utilisait.

Toulouse : un seul Cluster était disponible à chaque fois.

Orsay : Netgdx était rarement disponible et quand il l'était, n'offrait que très peu de Nodes.

Lyon : Capricorne était rarement disponible et quand il l'était, n'offrait que très peu de Nodes.

Sophia mis à part, on constate que les Sites ne présentent pas des disponibilités similaires de leurs Clusters. Certains Clusters sont beaucoup plus utilisés que d'autres, et quelques Clusters souffrent sont même rarement en ligne.

3. Irrégularité des performances des Clusters :

Dans cette partie j'ai essayé de détecter s'il y avait des Sites ou des Clusters qui affichent des performances bien différentes de celles des autres.

a. Démarche expérimentale :

En utilisant les moyennes des temps de déploiement pour chaque déploiement par Site et chaque déploiement par Cluster on trace des graphes comparant les performances des Clusters et des Sites.

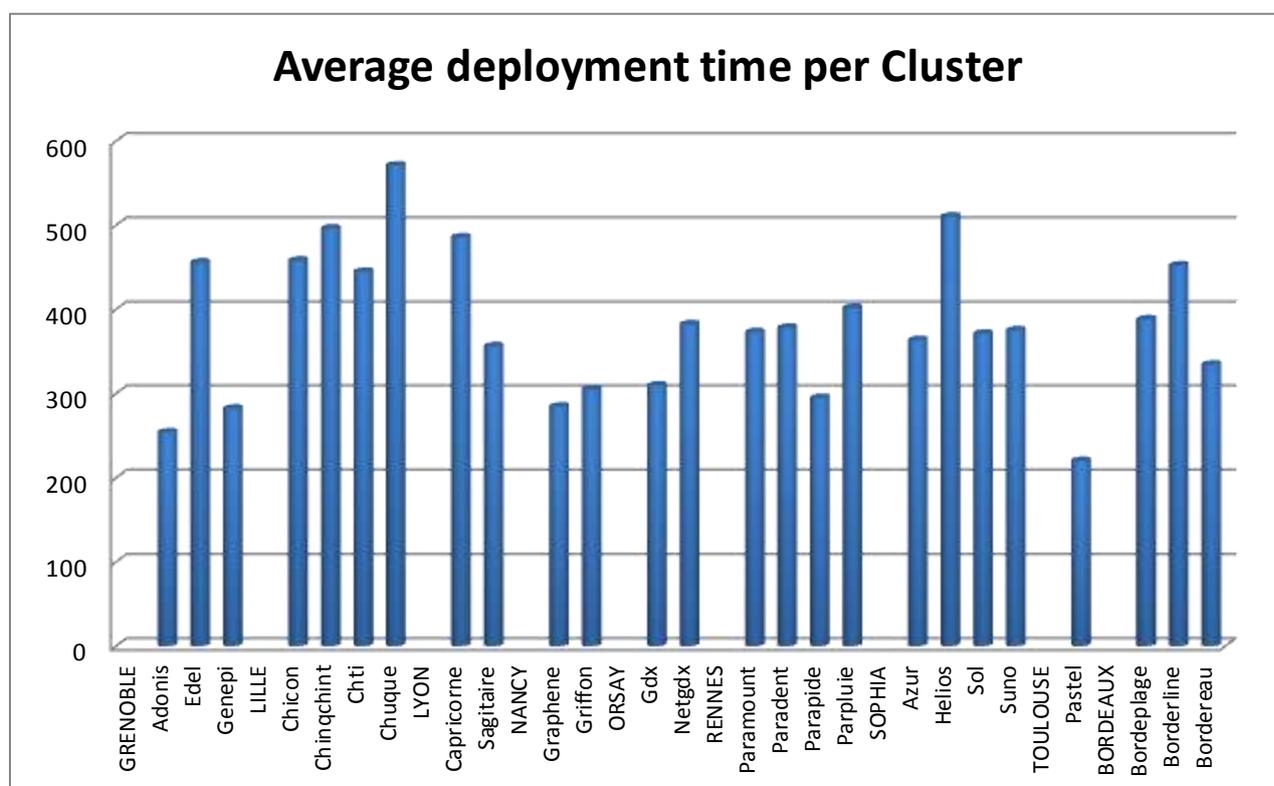
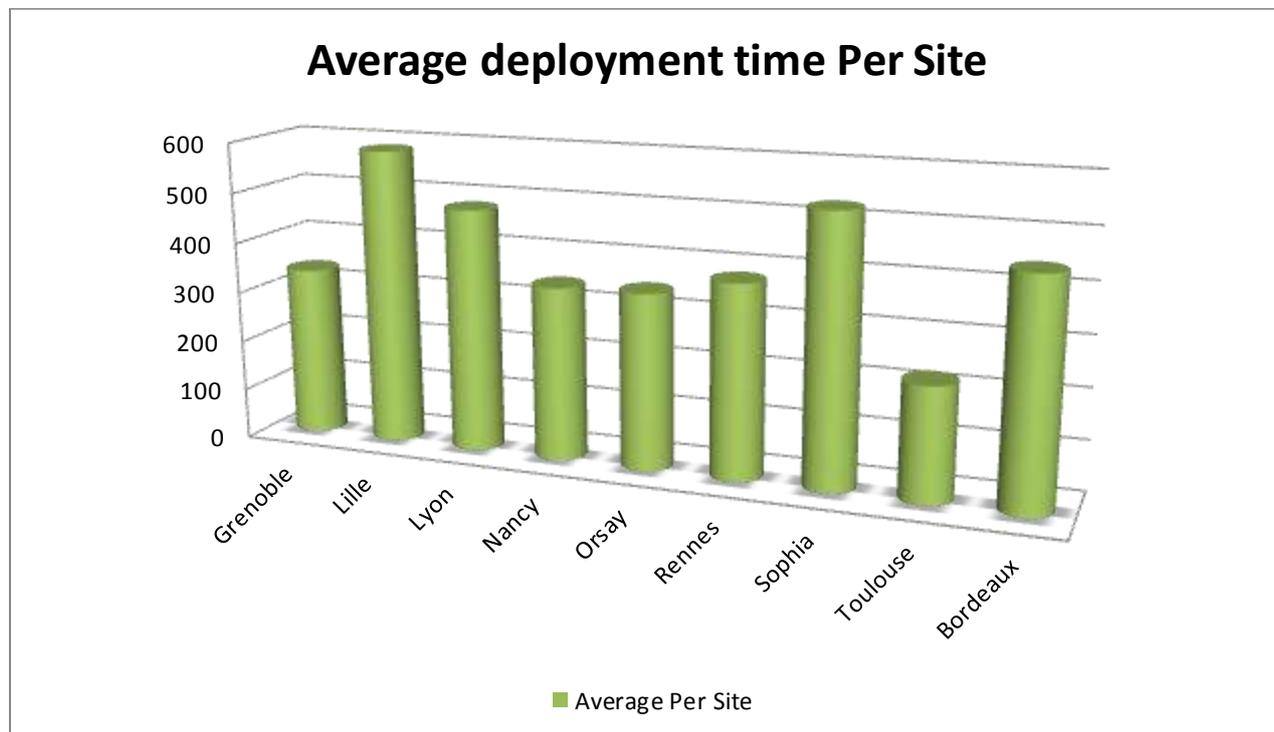
b. Résultats :

Les graphes comparatifs sont tracés ci-dessous :

Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG



Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

c. Analyse:

En définissant un seuil de 350 secondes au-dessus duquel un déploiement pourrait être considéré lent, on peut séparer les Clusters en Clusters lents et Clusters rapides. Résumés dans le tableau suivant.

Site	Performance	Cluster	Performance
Grenoble	Rapide	Adonis	Rapide
		Edel	Lent
		Genepi	Rapide
Lille	Lent	Chicon	Lent
		Chinqchint	Lent
		Chti	Lent
		Chuque	Lent
Lyon	Lent	Capricorne	Lent
		Sagitaire	Lent
Nancy	Rapide	Graphene	Rapide
		Griffon	Rapide
Orsay	Rapide	Gdx	Rapide
		Netgdx	Lent
Rennes	Lent	Paramount	Lent
		Paradent	Lent
		Parapide	Rapide
		Parapluie	Lent
Sophia	Lent	Azure	Lent
		Helios	Lent
		Sol	Lent
		Suno	Lent
Toulouse	Rapide	Pastel	Rapide
Bordeaux	Lent	Bordeplage	Lent
		Borderline	Lent
		Bordereau	Rapide

On note également que certains Cluster sont assez lents à déployer nos images d'environnement de travail. Notamment Chuque et Helios, qui présentent d'ailleurs à eux deux un taux d'échec assez élevé.

Au niveau des sites, Lyon et Orsay sont des Sites qui m'ont particulièrement donné du mal pour déployer les environnements avec ces échecs répétitifs et une très haute indisponibilité

Sophia et Rennes eux présentaient un autre genre d'anomalie. Ces sites, après avoir reçu la commande de déploiement, devaient renvoyer une variable contenant la liste de tous les nœuds qui ont reçu cette commande. Mais à plusieurs reprises ces deux sites renvoyaient une variable de valeur NULL.

Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

Avec un peu plus d'analyse il s'est avéré qu'à cause d'une erreur dans la communication HTTP, ces deux sites échouaient à nous renvoyer la liste des nœuds qui reçoivent les commandes de déploiement quand on la demande. Cette liste quand elle est vide, mène à la terminaison du script alors que les déploiements peuvent toujours être actifs au niveau des Nodes. Je recommanderais donc d'effectuer plus d'investigation sur ce sujet.

4. Corrélation entre nombre de Nodes et performance du déploiement :

Comme notre souci initial est la performance de l'utilisation du Grid'5000 à grande échelle, il est naturel qu'on se pose des questions quant à l'effet du nombre des Nodes auxquels on fait appel pour un déploiement sur sa performance.

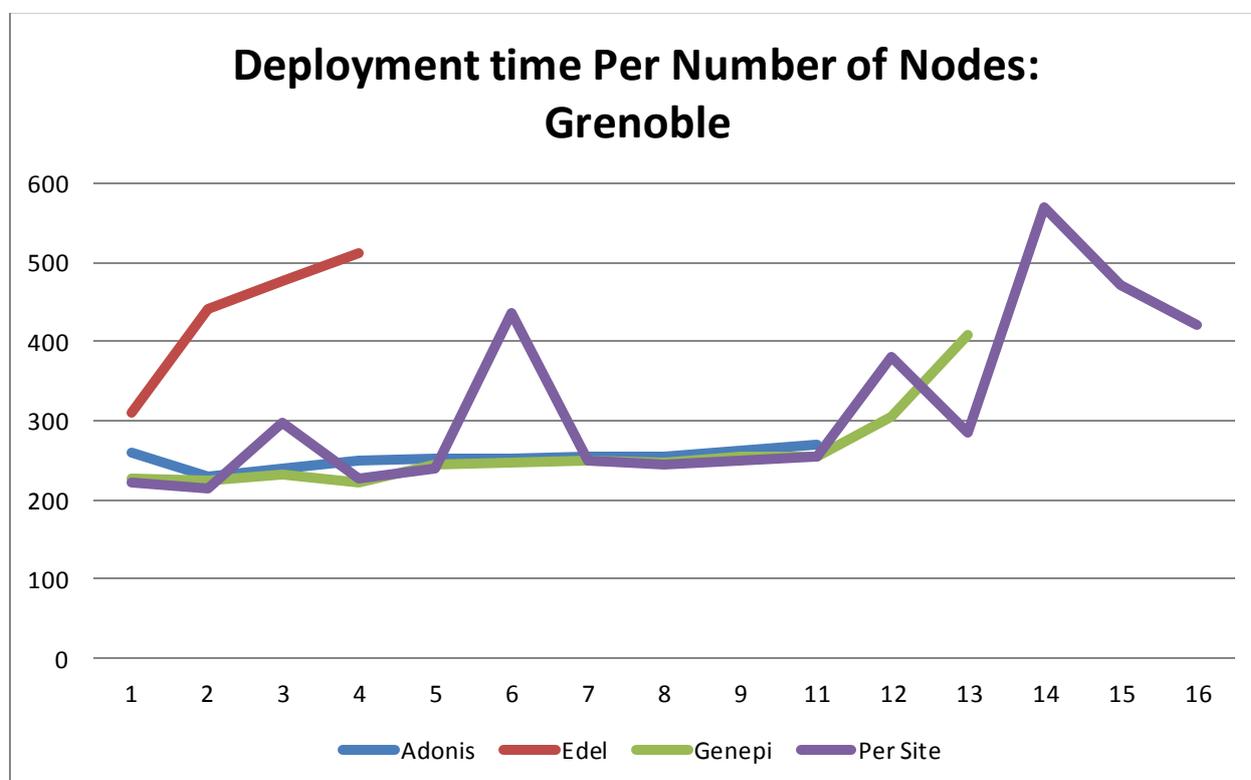
a. Démarche expérimentale :

Pour répondre à cette question j'ai refait la série d'expériences utilisées ci-dessus et qui consiste à soumettre des Jobs en variant à chaque fois le nombre de Nodes demandés, partant d'un seul Node jusqu'au maximum des Nodes disponibles (Nodes=BEST).

On relève à chaque fois le temps de déploiement ainsi que le pourcentage d'échecs. Toutes les informations obtenues sont groupées dans l'annexe « Grid Experiment.xls »

b. Résultats :

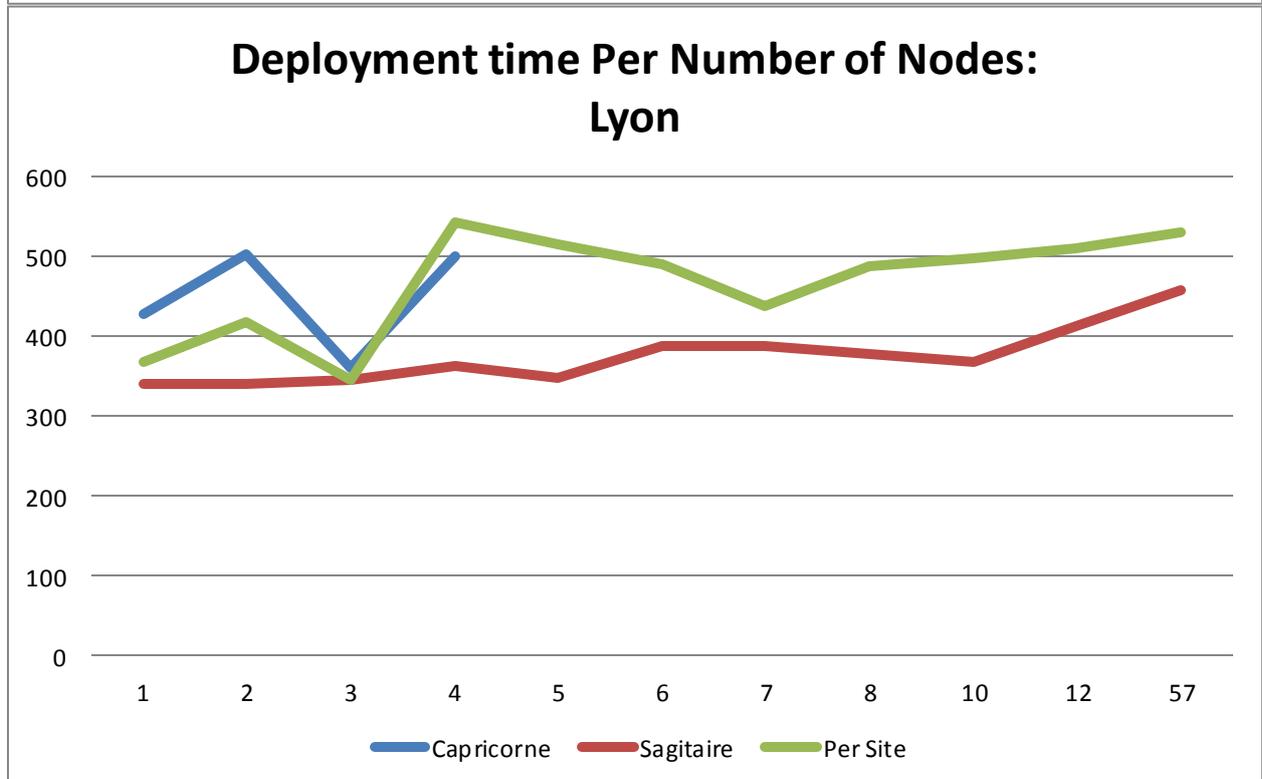
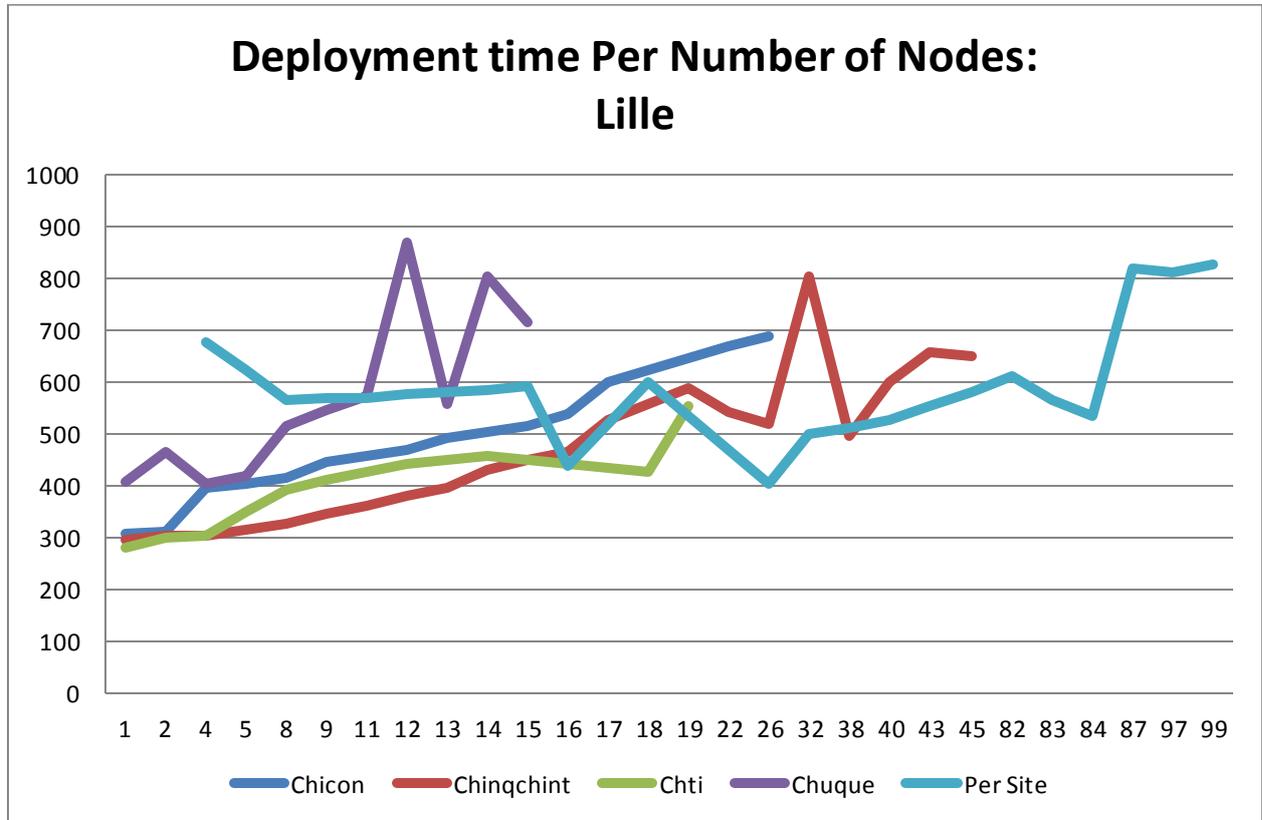
Dans ces graphes j'ai présenté la variation du temps de déploiement moyen en fonction du nombre de Nodes utilisés.



Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

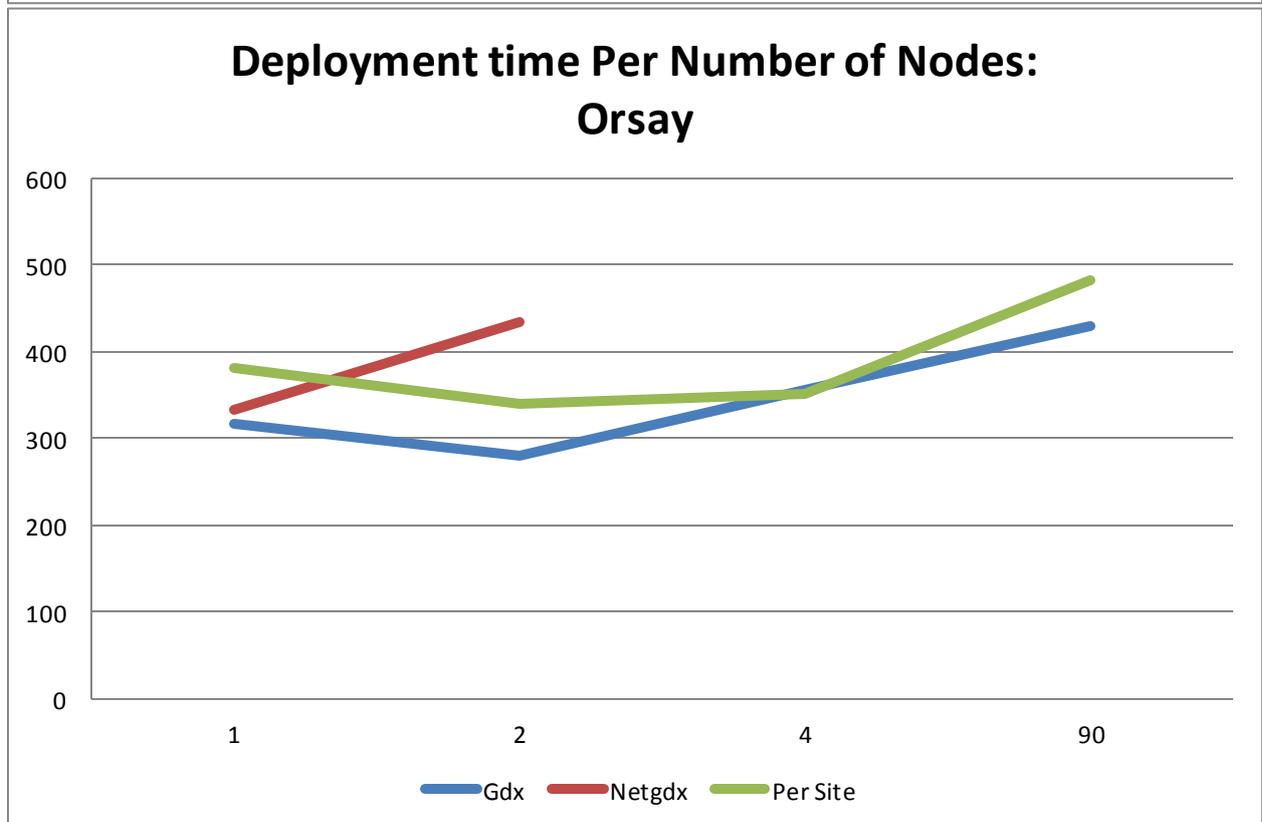
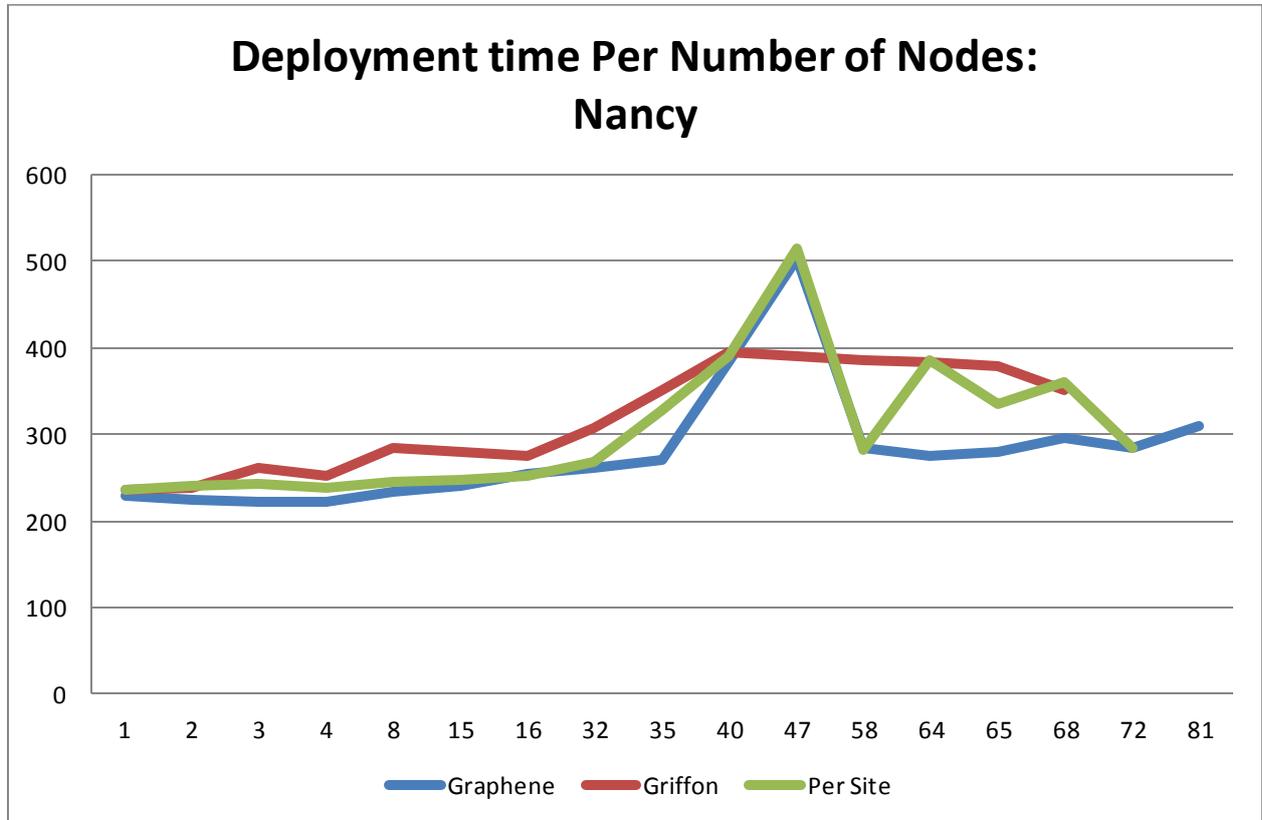
Marouen ZELLEG



Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

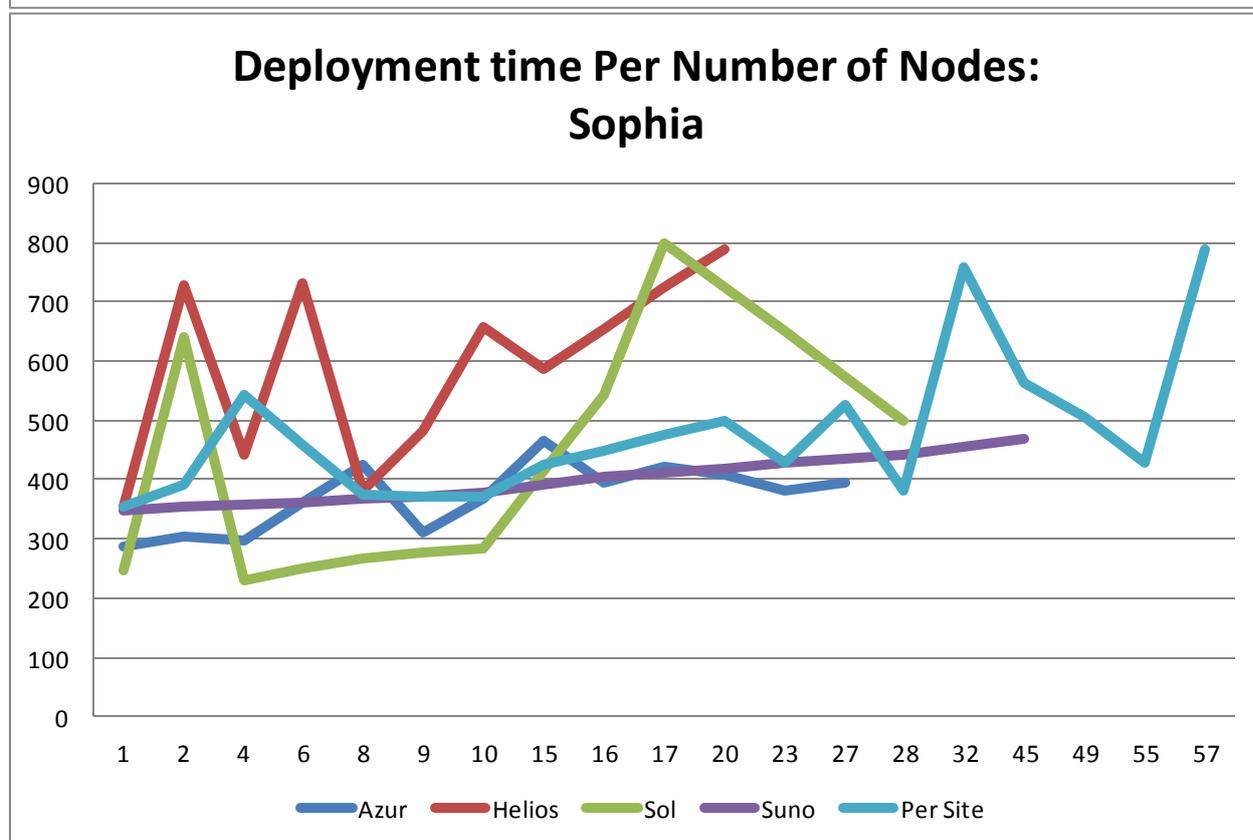
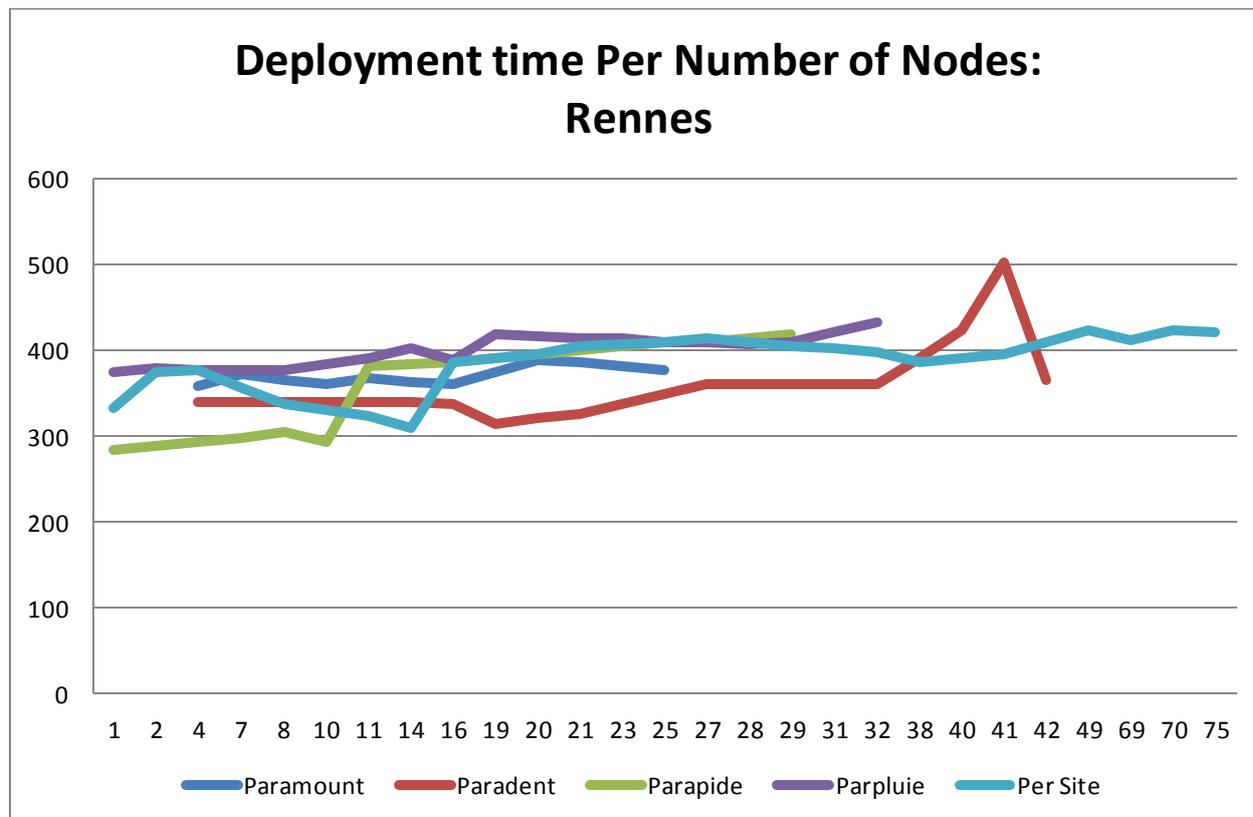
Marouen ZELLEG



Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

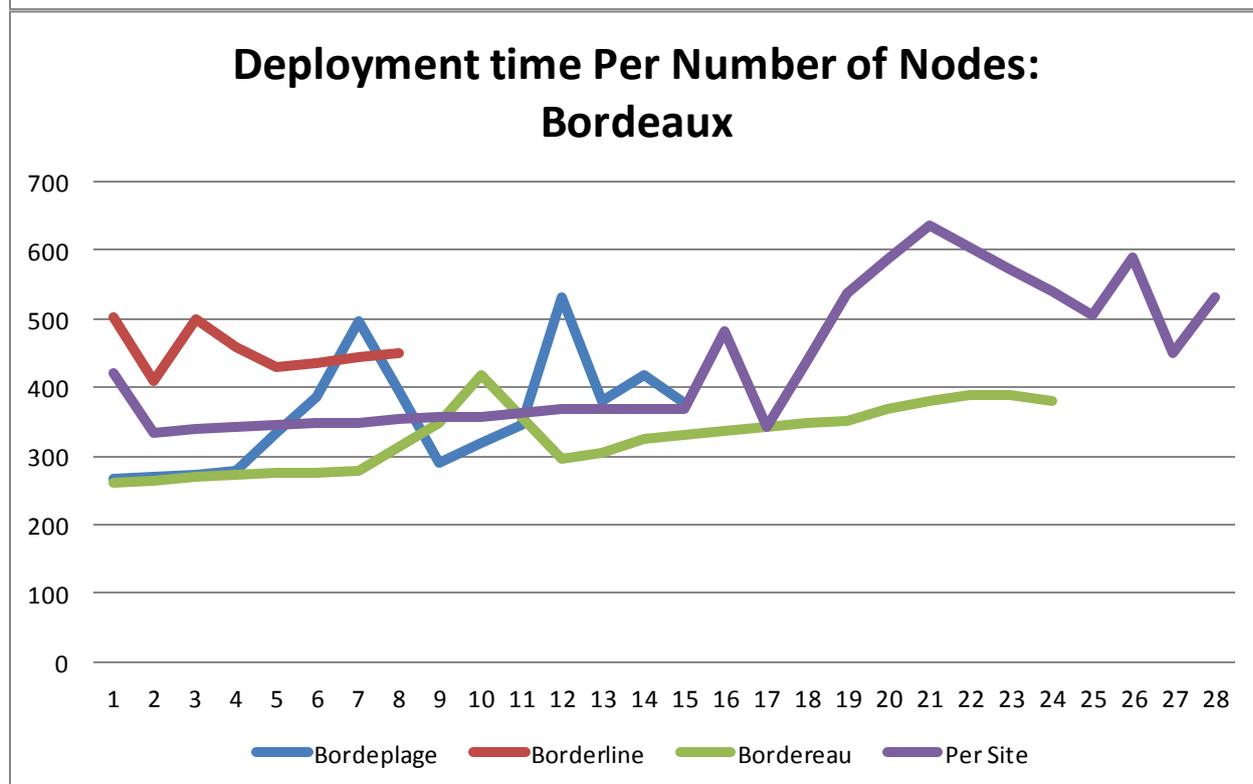
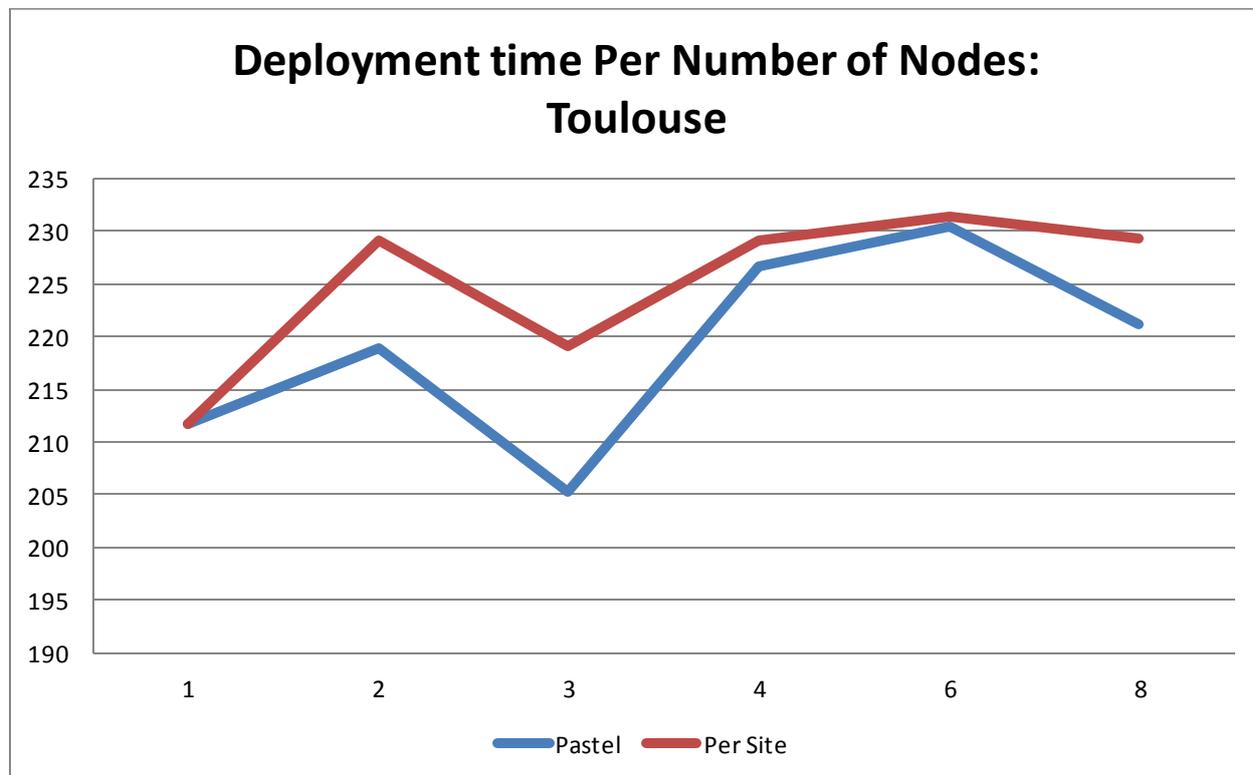
Marouen ZELLEG



Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG



Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

c. Analyse :

Site	La Performance :	Cluster	La Performance :
Grenoble	Souffre de la chute de performance d'Edel quand il s'agit de peu de Nodes et ensuite de la chute de performance de Genepi quand on dépasse la dizaine de Nodes.	Adonis	Insensible à la variation du nombre de Nodes
		Edel	Varie de manière critique face à la moindre augmentation du nombre des Nodes
		Genepi	Baisse légèrement quand le nombre de Nodes augmente au-delà d'une dizaine
Lille	Suit une moyenne entre les performances des différents Clusters du Site.	Chicon	Chute proportionnellement à l'augmentation du nombre de Nodes
		Chingchint	
		Chti	
		Chuque	Chute de manière critique quand on dépasse les 10 Nodes
Lyon	Plutôt stable grâce à l'absence fréquente du Cluster Capricorne.	Capricorne	Plutôt médiocre et de toute façon ce Cluster a été souvent absent.
		Sagitaire	Stable
Nancy	Bonne, conditionnée par la performance de Griffon	Graphene	Très bonne, et plutôt insensible à l'augmentation du nombre des Nodes.
		Griffon	Bonne pour un nombre réduit de Nodes ensuite chute légèrement quand on dépasse les 30 Nodes.
Orsay	Plutôt insensible à l'augmentation du nombre de Nodes.	Gdx	Plutôt insensible à l'augmentation du nombre de Nodes
		Netgdx	Peu de données pour la déduire. Cluster rarement disponible.

Rapport de Projet 3A

Exploitation Efficace à Grande Echelle de Grid'5000

Marouen ZELLEG

Rennes	Conditionnée par les variations de Paradent à grand nombre de Nodes et Parapide à faible nombre de Nodes.	Paramount	Plutôt insensible à l'augmentation du nombre de Nodes
		Paradent	Bonne, mais chute légèrement quand on atteint 40 Nodes
		Parapide	Bonne, mais chute légèrement quand on atteint une dizaine de Nodes
		Parapluie	Plutôt insensible à l'augmentation du nombre de Nodes
Sophia	Connait des variations importantes qui dépendent de la participation ou non des Clusters Helios et Sol dans le déploiement	Azure	Plutôt insensible à l'augmentation du nombre de Nodes
		Helios	Médiocre et assez instable, les déploiements sur ces Cluster échouent souvent.
		Sol	
		Suno	Plutôt insensible à l'augmentation du nombre de Nodes
Toulouse	Bonne et stable, mais on n'a pas eu la chance de la tester avec un grand nombre de Nodes (qui dépasse les 10)	Pastel	Bonne et stable, mais on n'a pas eu la chance de la tester avec un grand nombre de Nodes (qui dépasse les 10)
Bordeaux	Souffre de la mauvaise performance de Bordeplage et Bordeline	Bordeplage	Très oscillante mais généralement moyenne.
		Borderline	mauvaise
		Bordereau	Bonne et plutôt insensible à l'augmentation du nombre de Nodes

Conclusion :

En conclusion générale on remarque pour la plupart des Sites que la présence d'un Cluster pas très performant ou de performance instable affecte la performance globale d'un site lors d'un déploiement par site.

Je conseillerais donc les chercheurs d'effectuer des déploiements par Cluster et si possible de viser seulement les Clusters les plus performants pour profiter d'un déploiement à grande échelle rapide et avec un bas taux d'échec.

Lors de mon projet j'ai eu la chance d'assister à la mise en marche du Cluster Graphene sur le Site Nancy. Et je n'ai pu que remarquer l'impact que cet investissement a eu sur la performance de ce Site. J'espère donc qu'avec cette étude l'équipe de maintenance de Grid'5000 pourrait cibler les Clusters qui réduisent la performance globale de leurs Sites et ainsi apporter les modifications nécessaires pour avoir des ressources homogènes tout au long du Grid. Cela pourrait réduire la charge sur quelques sites et ainsi encourager les chercheurs à compter sur les services de l'ensemble du réseau.

Au niveau des Sites même j'ai remarqué que certains Clusters sont plus sollicités que d'autres, hormis les cas où un utilisateur demande un Cluster particulier pour ses Jobs, une répartition plus égale de la charge de travail sur les Nodes et Cluster du Site pourrait être bénéfique pour les machines du Grid.

La recherche scientifique est un domaine passionnant grâce aux larges horizons de travail qu'il offre. Rejoindre une équipe particulière de recherche, certes définit votre domaine de recherche, mais vous laisse tout de même une infinité de choix de sujets et d'axes de travail.

Cela peut être pourtant une arme à double tranchant, car le jeune chercheur risque de se perdre entre les différents sujets à traiter et ainsi diluer son énergie sur des aspects différents de son domaine de recherche ce qui peut facilement compromettre la pertinence de son travail.

Cela a affecté en quelque sorte le début de mon projet. Ne maîtrisant pas les différentes composantes de mon environnement de travail : le langage Ruby, l'usage des API de Grid'5000 et les commandes du shell linux, mes efforts ont été distribués entre la maîtrise de ces différents outils. Après un bout de temps j'ai commencé à me sentir confortable avec l'utilisation du Grid et la programmation en Ruby, j'ai donc veillé à échanger avec mon encadrant pour bien cibler mon travail et définir des buts précis à atteindre et qui se sont conclu par le rapport ci-présent.