Compas'2022 : Parallélisme / Architecture/ Système
MIS/UPJV - Amiens France, 5-8 juillet 2022

Étude des applications *Bag-of-Tasks* du méso-centre *Gricad*

Quentin Guilloteau, Olivier Richard, Eric Rutten

Univ. Grenoble Alpes, Inria, CNRS, LIG, F-38000 Grenoble France
Prénom.Nom@inria.fr

Résumé
La sous-exploitation des ressources laissées libres dans une grille de calculs représente un manque à gagner non négligeable pour une meilleure utilisation des grappes. Une solution est celle mise en place par *CiGri* dans le méso-centre *Gricad*. *CiGri* soumet des tâches faiblement prioritaires provenant d'applications dites *Bag-of-Tasks* aux ordonnanceurs des différentes grappes de la grille de calculs afin d'utiliser les ressources libres. Dans l'objectif de générer des profils réalistes de telles applications, nous nous intéressons dans cet article aux caractéristiques (quantité et durées d'exécution) des tâches exécutées sur le méso-centre *Gricad* ces 5 dernières années.

Mots-clés : Calculs Hautes Performances, Grille de Calculs, Applications *Bag-of-Tasks*

---

1. Introduction

Tous les scientifiques expérimentaux ont besoin d'exécuter des calculs. Cependant certains de ces calculs sont trop gourmands en termes de ressources (*e.g.*, CPU, mémoire, etc) et ne peuvent pas être réalisés sur une machine personnelle. Ainsi les scientifiques exécutent leurs tâches sur des grappes de calculs. Une grappe est un ensemble de machines de calculs partagées entre utilisateurs. Les machines d'une grappe ont la même configuration matérielle et logicielle. Plusieurs grappes peuvent être reliées entre elles pour former une grille de calculs.
Les utilisateurs de la grille doivent passer par un mécanisme de réservation et soumission pour accéder à une machine d'une des grappes et exécuter leurs tâches. Cependant, ce processus peut amener à la non-utilisation de certaines ressources de la grille. Cette sous-utilisation représente un manque à gagner important et doit être exploitée pour une meilleure utilisation des machines.
Il existe diverses solutions pour utiliser ces ressources libres. L'approche globale à la collecte de ressources consiste à exécuter des tâches courtes, faiblement prioritaires et pouvant être interrompues si nécessaire. Cette stratégie a notamment été appliquée sur des systèmes tels que : *(i)* un réseau de machines personnelles avec le projet *BOINC* [1], *(ii)* des machines de travail avec *Condor* [10], *(iii)* une grappe de calculs haute performance avec des tâches provenant du *Big-Data* [11], ou encore *(iv)* un méso-centre avec *CiGri* [5] et des applications *Bag-of-Tasks*..
Nous nous intéressons dans ce papier à l'étude de ces applications *Bag-of-Tasks* exécutées via l'intergiciel *CiGri*. Le but est de connaître les caractéristiques de ces applications afin de pouvoir en générer de manière réaliste. Ces applications synthétiques serviront ensuite à améliorer l'évaluation de travaux tels que [6], basés sur des modifications de *CiGri*.

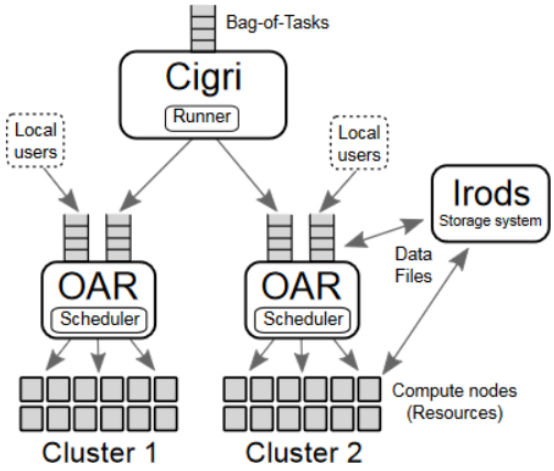


FIGURE 1 – Interactions entre l'intergiciel *CiGri* et les ordonnanceurs *OAR* des différentes grappes de calculs du méso-centre *Gricad*.

2. Contexte

2.1. Le méso-centre *Gricad*

Le méso-centre *Gricad* [1] est une structure qui fournit des infrastructures de calculs et de données aux chercheur·euse·s grenoblois·es. Le méso-centre est composé de plusieurs grappes de calculs (par exemple `dahu` pour le *HPCDA* [2] et la convergence *HPC-Big Data*, `froggy` pour le *HPC* ou encore `luke` pour le traitement de données). Ces grappes sont regroupées en une grille de calculs, cette dernière étant sujette aux problématiques des ressources inutilisées.

2.2. *CiGri*

Une approche différente à la récolte de ressources libres est celle prise par *CiGri*[5]. *CiGri* est un intergiciel pour grilles de calculs mis en place sur le méso-centre *Gricad*. Il se place au-dessus d'un ensemble de grappes gérées par des ordonnanceurs *OAR*[3]. Son but est d'utiliser les ressources libres du méso-centre *Gricad*.

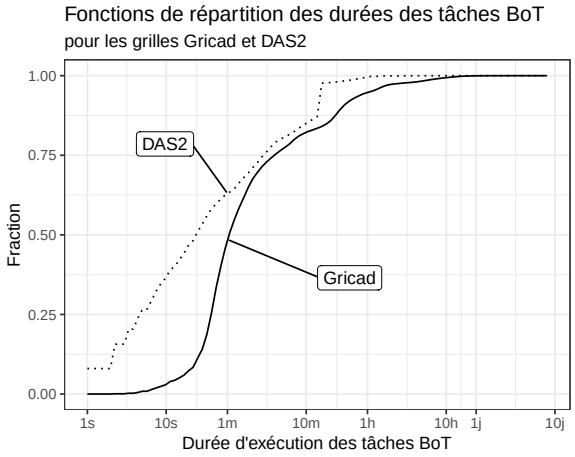
Les utilisateurs de *CiGri* soumettent à l'intergiciel des applications *Bag-of-Tasks*. Ces applications sont composées de nombreuses tâches courtes, indépendantes et ayant des comportements similaires (temps d'exécutions, I/O, etc). Un exemple d'application *Bag-of-Tasks* sont les simulations *Monte-Carlo* consistant à exécuter un grand nombre d'expériences aléatoires courtes et conclure sur les résultats agrégés. Ce genre d'applications, aussi caractérisé de *"embarassingly parallel"*, est ainsi propice à la collecte de ressources libres.

CiGri va ensuite soumettre ces tâches aux différentes grappes avec la priorité la plus faible (*Best-Effort*). Cela permet aux ordonnanceurs des grappes de pouvoir arrêter l'exécution des tâches *CiGri* si des utilisateurs plus prioritaires ont besoin des ressources.

Les applications soumises à *CiGri* appartiennent à des *projets* et sont composées de *campagnes*. Chaque *campagne* comprend un ensemble de tâches à exécuter.

La figure 1 résume les interactions entre l'intergiciel *CiGri* et les ordonnanceurs *OAR* des différentes grappes de la grille de calculs.

1. `https://gricad.univ-grenoble-alpes.fr/`
2. *High Performance Computing-Data Analysis*



(a) Comparaison des fonctions de répartition empiriques des durées des tâches *Bag-of-Tasks* pour les grilles *Gricad* et *DAS2*.

Métriques	Grilles de Calculs	
	Gricad	*DAS2*
Nombre de tâches BoT	$\simeq 3 \times 10^7$	$\simeq 10^5$
Nombre de grappes	5	5
Minimum t_{exec}	1s	1s
Moyenne t_{exec}	19m	3m 51s
Maximum t_{exec}	7j 22h 20m	10h
Médiane t_{exec}	1m 3s	24s
Quantile 75 % t_{exec}	3m 55s	2m 51s
Quantile 95 % t_{exec}	1h 5m 38s	15m 1s
Quantile 99 % t_{exec}	7h 28m 40s	43m 35s

(b) Table récapitulative des données. t_{exec} représente les durées d'exécution des tâches *Bag-of-Tasks* des grilles *Gricad* et *DAS2*.

FIGURE 2 – Fonctions de répartition et tableau récapitulatif pour les données des tâches *Bag-of-Tasks* provenant des grilles *Gricad* et *DAS2*

3. Étude

Cette étude a pour but d'identifier les caractéristiques des applications soumises à *CiGri* et repose sur les tâches soumises par l'intergiciel *CiGri* sur le méso-centre *Gricad* entre le 2 janvier 2017 et le 8 octobre 2021 (58 mois), soit environ 30 millions de tâches. Le jeu de données est disponible sur Zenodo [3] et les scripts d'analyse sur Gitlab [4].

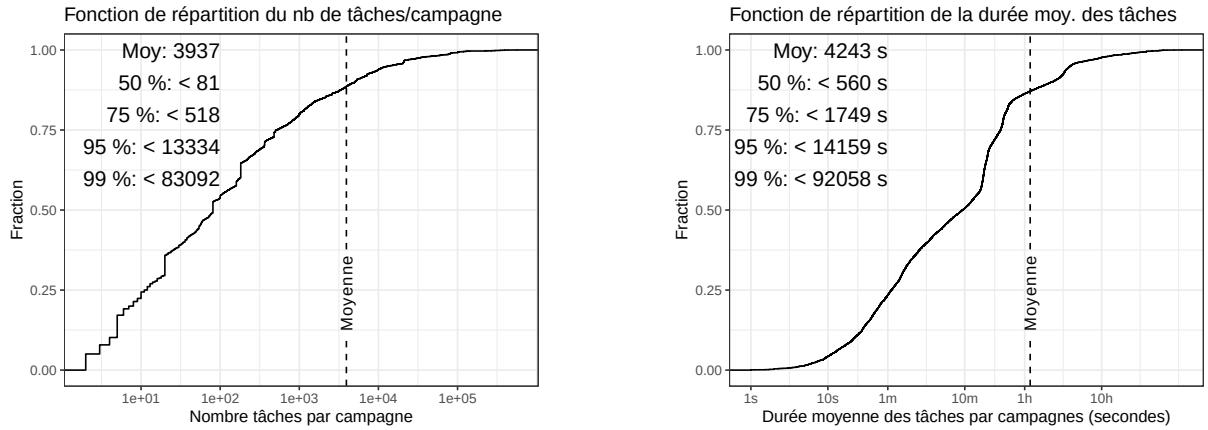
3.1. Caractéristiques globales

Dans cette section, nous nous intéresserons en particulier au temps d'exécution de ces tâches.
La figure 2a compare les fonctions de répartitions des durées des tâches provenant d'applications *Bag-of-Tasks* pour la grille *Gricad* et *DAS2*. *Distributed ASCI Supercomputer 2 (DAS2)* [5] est la deuxième itération d'une grille de calculs entre universités hollandaises. La table 2b donne pour ces deux grilles les ordres de grandeurs pour plusieurs métriques. Nous constatons que la majorité des tâches ont un faible temps d'exécution. En effet, la moitié des tâches durent moins d'une minute pour *Gricad* et moins de 30 secondes pour *DAS2*. Cependant, en moyenne, une tâche provenant de *CiGri* s'exécute pendant environ 19 minutes. Nous remarquons qu'il y a plus de tâches *Bag-of-Tasks* courtes pour *DAS2* que pour *Gricad*. Dans les deux cas, environ 75 % des tâches ont une durée d'exécution de l'ordre de quelques minutes. Cela nous conforte sur le fait que la majorité des tâches *Bag-of-Tasks* sont courtes.

3.2. Caractéristiques par campagne

Toujours dans l'optique de modéliser de manière réaliste les campagnes des taches des applications *Bag-of-Tasks* de *CiGri*, nous cherchons dans cette section à représenter la "campagne moyenne" en nombre de tâches et temps d'exécution.
La figure 3 représente les fonctions de répartition empiriques pour le nombre de tâches dans

3. https://zenodo.org/record/6787030
4. https://gitlab.inria.fr/cigri-ctrl/compas22_etude_bot_gricad
5. https://www.cs.vu.nl/das5/home.shtml



(a) Fonction de répartition empirique du nombre de tâches par campagne

(b) Fonction de répartition empirique de la durée moyenne des tâches par campagne

FIGURE 3 – Fonctions de répartition empiriques pour le nombre de tâches et leur durée moyenne par campagne

une campagne (figure 3a) et pour la durée moyenne des tâches dans une campagne (figure 3b). Ainsi, 50 % des campagnes comprennent moins de 100 tâches et 50 % des campagnes ont des tâches s'exécutant en moyenne en moins de 560 secondes (9 minutes et 20 secondes).
La campagne moyenne possède environ 4000 tâches s'exécutant en moyenne en une heure et 10 minutes.

3.3. Caractéristiques par projet

Nous allons maintenant nous intéresser aux distributions des temps d'exécution par projet.
La figure 4 montre les durées des tâches appartenant aux 10 projets ayant le plus de tâches. La majorité des distributions sont similaires : un pic autour d'une valeur et une "queue lourde".

Protocole

En nous inspirant de [9] et [2], nous avons réalisé plusieurs tests de qualité d'ajustement (*"Goodness of fit"*) avec différentes potentielles distributions à queues longues (*Normale*, *Log-Normale*, *Weibull*, *Frechet*, *Gamma*) et donné pour chaque projet celle qui semble la plus appropriée.
Nous nous plaçons dans le cas des 10 projets avec le plus de tâches présentés en figure 4. Pour chacune des campagnes de ces projets, nous choisissons aléatoirement 50 tâches. Comme le comportement des tâches peut varier selon la grappe (*e.g.,* accélérateur disponible ou non), nous considérons les distributions par couple $(\texttt{projet}, \texttt{grappe})$. Pour chacun de ces couples, et pour chacune des distributions sélectionnées, nous performons un test de Cramer-von Mises [4]. Ce test calcule la distance entre la fonction de répartition empirique de la loi théorique et celle de la distribution à tester. Plus la distance est faible, meilleur est l'ajustement. Comme le test de Cramer-von Mises génère aléatoirement la fonction de répartition empirique de la loi théorique, nous répétons ce procédé 30 fois et prenons la moyenne des distances. La table 1 présente pour chaque couple $(\texttt{projet}, \texttt{grappe})$ la distribution s'ajustant le mieux aux données.

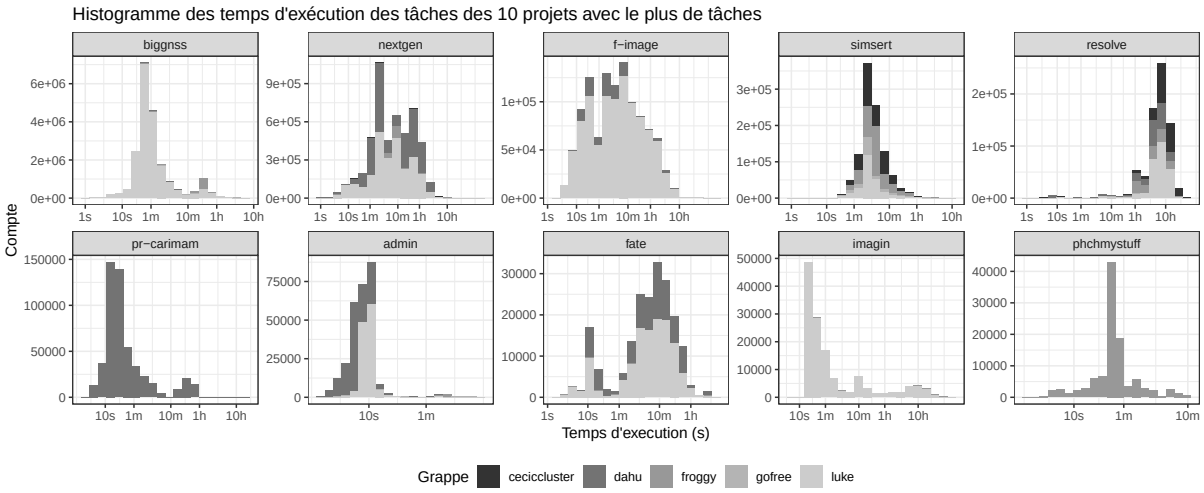


FIGURE 4 – Histogramme des temps d'exécution des tâches des 10 projets avec le plus de tâches

Projet	Grappe	Distribution
admin	Dahu	*Log-Normale*
	Froggy	*Frechet*
	Gofree	*Gamma*
	Luke	*Log-Normale*
biggnss	Dahu	*Normale*
	Froggy	✗
	Gofree	✗
	Luke	*Frechet*
f-image	Dahu	*Frechet*
	Luke	*Log-Normale*
fate	Dahu	*Weibull*
	Luke	*Weibull*
imagin	Froggy	*Normale*
	Luke	*Weibull*

Projet	Grappe	Distribution
nextgen	Ceci	*Frechet*
	Dahu	*Weibull*
	Froggy	*Weibull*
	Luke	*Gamma*
phcmystuff	Froggy	*Log-Normale*
pr-carimam	Dahu	*Frechet*
resolve	Ceci	*Gamma*
	Dahu	*Normale*
	Froggy	*Normale*
	Luke	*Normale*
simsert	Ceci	*Frechet*
	Dahu	*Frechet*
	Froggy	*Frechet*
	Luke	*Frechet*

TABLE 1 – Distributions s'ajustant le mieux aux données de temps d'exécution pou chaque couple (projet, grappe)

Analyse

Globalement, les distributions les plus adaptées sont les lois *Log-Normale*, *Frechet* et *Weibull* avec une qualité quasi identique. Ceci est en accord avec les résultats de [7] pour d'autres grilles de calculs et applications *Bag-of-Tasks*. Nous pouvons également voir sur la table 1 que certains projets ont la même distribution peu importe la grappe (*e.g.*, `simsert`, `fate`). Ce qui n'est pas le cas pour d'autres, comme pour le projet `admin` par exemple, ou encore pour `biggnss`, où aucune distribution ne semble s'ajuster suffisamment.

4. Conclusion

Nous avons vu qu'il existe plusieurs types de comportements pour les applications *Bag-of-Tasks* de *Gricad*. La majorité des campagnes contient un faible nombre de tâches qui s'exécutent en quelques minutes. Ces durées d'exécution peuvent être modélisées par des distributions de probabilités usuelles (*Weibull*, *Frechet* ou *Log-Normale*). Avec la figure 3 et la table 1, nous sommes maintenant capables de représenter la *"campagne moyenne"* en nombre de tâches, durées d'exécution et distribution de ces dernières.
Il reste cependant des questions ouvertes, et notamment celle des raisons de ces "queues lourdes". Pour répondre à cela, il faudrait certainement s'intéresser à l'état global de la plateforme (*e.g.*, quantité de tâches sur les grappes, charge du système de fichiers) lors de l'exécution des tâches de *CiGri*.

Remerciements

Merci au méso-centre *Gricad*, et notamment à Bruno Bzeznik, pour l'accès aux données, et à GWA[8] pour celles de *DAS2*.

Bibliographie

1. Anderson (D.). – BOINC : A System for Public-Resource Computing and Storage. – In *Fifth IEEE/ACM International Workshop on Grid Computing*, pp. 4–10, Pittsburgh, PA, USA, 2004. IEEE.
2. Brevik (J.), Nurmi (D.) et Wolski (R.). – *Quantifying machine availability in networked and desktop grid systems*. – Rapport technique, Technical Report CS2003-37, Dept. of Computer Science and Engineering . . ., 2003.
3. Capit (N.), Da Costa (G.), Georgiou (Y.), Huard (G.), Martin (C.), Mounie (G.), Neyron (P.) et Richard (O.). – A batch scheduler with high level components. – In *CCGrid 2005. IEEE International Symposium on Cluster Computing and the Grid, 2005.*, pp. 776–783 Vol. 2, Cardiff, Wales, UK, 2005. IEEE.
4. Darling (D. A.). – The kolmogorov-smirnov, cramer-von mises tests. *The Annals of Mathematical Statistics*, vol. 28, n4, 1957, pp. 823–838.
5. Georgiou (Y.), Richard (O.) et Capit (N.). – Evaluations of the Lightweight Grid CIGRI upon the Grid5000 Platform. – In *Third IEEE International Conference on e-Science and Grid Computing (e-Science 2007)*, pp. 279–286, Bangalore, India, 2007. IEEE.
6. Guilloteau (Q.) et al. – Controlling the Injection of Best-Effort Tasks to Harvest Idle Computing Grid Resources. – In *ICSTCC 2021 - 25th International Conference on System Theory, Control and Computing*, pp. 1–6, Iași, Romania, octobre 2021.
7. Iosup (A.) et Epema (D.). – Grid computing workloads. *IEEE Internet Computing*, vol. 15, n 2, 2010, pp. 19–26.

8. Iosup (A.), Li (H.), Jan (M.), Anoep (S.), Dumitrescu (C.), Wolters (L.) et Epema (D. H.). – The grid workloads archive. *Future Generation Computer Systems*, vol. 24, n7, 2008, pp. 672–686.
9. Javadi (B.), Kondo (D.), Vincent (J.-M.) et Anderson (D. P.). – Mining for statistical models of availability in large-scale distributed systems : An empirical study of seti@ home. – In *2009 IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems*, pp. 1–10. IEEE, 2009.
10. Litzkow (M. J.), Livny (M.) et Mutka (M. W.). – *Condor-a hunter of idle workstations*. – Rapport technique, University of Wisconsin-Madison Department of Computer Sciences, 1987.
11. Mercier (M.), Glesser (D.), Georgiou (Y.) et Richard (O.). – Big data and HPC collocation : Using HPC idle resources for Big Data analytics. – In *2017 IEEE International Conference on Big Data (Big Data)*, pp. 347–352, Boston, MA, décembre 2017. IEEE.