



CNES

Equipe Fondation Calcul

Direction du Numérique, de l'Exploitation et des Opérations
Sous-direction "Infrastructures numériques, SI scientifique et
applicatif"

Service "Calcul, Ingénierie logicielle et valorisation des
Données"

Edition : 01 Date : 06/01/2022
Révision : 00 Date : 07/10/2021

Réf. : DNO/ISA/CID-2021.0014206

GUIDE DE BON USAGE ET D'INTEGRATION SUR LES SERVICES DU CENTRE DE CALCUL DU CNES

Rédigé par : Annaïg Pedrono Florent Ventimiglia Guillaume Eynard-Bontemps	le : 06/01/2022	
Validé par : Guillaume Eynard-Bontemps	le : 06/01/2022	
Pour application : Eric Morand	le : 06/01/2022	

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01 Date : 07/10/2021 Rév. : 00 Date : 07/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 2

BORDEREAU D'INDEXATION

CONFIDENTIALITE : P		MOTS CLES : Centre de Calcul, Utilisation, Intégration, HPC, Portage, Développement, Migration, codes, algorithmes	
TITRE DU DOCUMENT : Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES			
AUTEUR(S) :			
RESUME :			
DOCUMENTS RATTACHES : Ce document vit seul.			LOCALISATION : DNO/ISA/CID
VOLUME : 1	NBRE TOTAL DE PAGES : 23 DONT PAGES LIMINAIRES : 5 NBRE DE PAGES SUPPL. : 0	DOCUMENT COMPOSITE : N	LANGUE : FR
GESTION DE CONF. : NG		RESP. GEST. CONF. :	
CAUSE D'EVOLUTION : Création du document			
CONTRAT : Néant			
SYSTEME HOTE : Microsoft Word 11.0 (11.0.8345) D:\Donnees\Dot\ModeleGDOC.dot Version GDOC : v4.3.0.0_TW05 Base projet : \\To05res04\GdocBasesPartagees\Projets\CST\GAIA			

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 07/10/2021
	Rév. : 00	Date : 07/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 3	

DIFFUSION INTERNE

Nom	Sigle	Bpi	Observations
DEGUINE Béatrice	DNO/ISA	1321	
VENTIMIGLIA Florent	DNO/ISA/CID	1502	
EYNARD-BONTEMPS Guillaume	DNO/ISA/CID	1502	
PEDRONO Annaïg	DNO/ISA/CID	1502	
MORAND Eric	DNO/ISA/CID	1502	

DIFFUSION EXTERNE

Nom	Sigle	Observations
USSEGLIO Gaëlle	Thales Services	
PICHE Salima	Thales Services	

CNES

**Guide de bon usage et d'intégration sur les services
du Centre de Calcul du CNES**

Edit. : **01**

Date : **07/10/2021**

Rév. : **00**

Date : **07/10/2021**

Référence : **DNO/ISA/CID-2021.0014206**

Page : 4

MODIFICATION

Ed.	Rév.	Date	Référence, Auteur(s), Causes d'évolution
01	00		

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01 Date : 07/10/2021 Rév. : 00 Date : 07/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 5

SOMMAIRE

1.	GENERALITES	1
1.1.	DOCUMENTS APPLICABLES	1
1.2.	DOCUMENTS DE REFERENCE	1
2.	OBJET DU DOCUMENT.....	2
2.1.	CONTEXTE	2
2.2.	QUELQUES NOTIONS SUR LE CALCUL PARALLELE	2
2.3.	VALIDITE DU DOCUMENT	3
3.	PLANIFICATION DE LA VOLUMETRIE.....	4
4.	DISPONIBILITE ET SERVICES.....	5
4.1.	ARRETS DE PRODUCTION.....	5
4.1.1.	Plateformes de calcul.....	5
4.2.	ENGAGEMENTS DE SERVICES	5
4.2.1.	Plateformes de calcul.....	5
4.3.	CANAUX D'ADRESSAGE.....	5
5.	GUIDE ET EXIGENCES.....	7
5.1.	ORDONNANCEUR DE TRAVAUX	7
5.1.1.	E-HPC-ORD0 : Runtime d'exécution parallèle	7
5.1.2.	E-HPC-ORD1 : Réservation de cœurs de calcul	7
5.1.3.	E-HPC-ORD2 : Test de validation	7
5.1.4.	E-HPC-ORD3 : Job Array	7
5.1.5.	E_HPC-ORD4 : Fréquence d'appel de commande Ordonnanceur	7
5.1.6.	E-HPC-ORD5 : Environnement d'exécution	7
5.1.7.	E-HPC-ORD6 : Temps d'exécution minimal	8
5.1.8.	E-HPC-ORD7 : Réservation optimale des ressources.....	8
5.1.9.	E-HPC-ORD8 : Écriture des fichiers de sortie.....	8
5.2.	STOCKAGE ET ACCES AUX DONNEES	9
5.2.1.	E-HPC-DAT-0 : Accès optimisé aux données	9
5.2.2.	E-HPC-DAT1 : Limitation des flux d'IO par nœud.....	9
5.2.3.	E-HPC-DAT2 : Bufferisation des données.....	10
5.2.4.	E-HPC-DAT3 : Workflow de traitement de la donnée optimal	10
5.2.5.	E-HPC-DAT4 : Un filesystem n'est pas une base de donnée.....	11
5.2.6.	E-HPC-DAT5 : Synchronisation des tâches : système d'échange de message.....	11
5.2.7.	E-HPC-DAT6 : Limite sur le nombre de fichiers.....	11
5.2.8.	E-HPC-DAT7 : Limite sur le nombre de sous répertoires	11
5.2.9.	E-HPC-DAT8 : Interdiction de caractères interprétés dans les noms de fichier/répertoires	11
5.2.10.	E-HPC-DAT9 : Type de fichier et volume.....	11

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01 Date : 07/10/2021 Rév. : 00 Date : 07/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 6

5.2.11.	E-HPC-DAT10 : Gestion des fichiers	11
5.2.12.	E-HPC-DAT10 : Format de compression de fichiers	12
5.2.13.	E-HPC-TMP1 : Accès à l'espace local des nœuds de calcul	12
5.2.14.	E-HPC-NFS1 : Limitation du protocole NFS	12
5.3.	PLANIFICATION DES CAMPAGNES DE CALCUL	12
5.3.1.	E-HPC-PLAN1.....	12
5.3.2.	E-HPC-PLAN2.....	12
5.4.	ENVIRONNEMENT	13
5.4.1.	E-HPC-OS-1 : Version majeure d'OS	13
5.4.2.	E-HPC-OS-2 : Version mineure d'OS	13
5.5.	DEVELOPPEMENT ET OPTIMISATIONS.....	13
5.5.1.	E-HPC-DEV-1 : Langages de programmation préconisés	13
5.5.2.	E-HPC-DEV-2 : Version langage, compilateur, librairie	14
5.5.3.	E-HPC-DEV-3 : langage compilé	14
5.5.4.	E-HPC-DEV-4 : Compilation.....	14
5.5.5.	E-HPC-DEV-5 : Bibliothèques mathématiques	14
5.5.6.	E-HPC-DEV-6 : Optimisation des performances et validation.....	15
5.5.7.	E-HPC-DEV-7 : Checkpoint/Restart.....	15
5.6.	NŒUDS FRONTAUX	16
5.6.1.	E-HPC-FRT1 : Limitation des nœuds frontaux.....	16
5.7.	NŒUDS DE VISUALISATION	16
5.7.1.	E_HPC_VIS.....	16
	Il est interdit de lancer de lourdes charges de calcul sur les nœuds de visualisation.....	16
5.8.	CRONTAB.....	16
5.8.1.	E-HPC-CRON1.....	16
5.8.2.	E-HPC-CRON2.....	16

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 1	

1. GENERALITES

1.1. DOCUMENTS APPLICABLES

- [DA2] RNC-CNES-Q-HB-80-501 - Règles communes pour l'utilisation des langages de programmation
- [DA3] RNC-CNES-Q-HB-80-505 - Règles pour l'utilisation du langage Fortran 77
- [DA4] RNC-CNES-Q-HB-80-517 - Règles pour l'utilisation du langage Fortran 90
- [DA5] RNC-CNES-Q-HB-80-518 - Recommandations pour l'obtention de la qualité numérique
- [DA6] RNC-CNES-Q-HB-80-527 - Règles pour l'utilisation du langage JAVA
- [DA7] RNC-CNES-Q-HB-80-535 - Règles pour l'utilisation des langages Python
- [DA8] RNC-CNES-Q-HB-80-536 - Règles pour l'utilisation des langages C, C++ et C embarqué

1.2. DOCUMENTS DE REFERENCE

- [DR1] HPC5G-DA-DEV-35-INT – Dossier d'Architecture de HAL – v1.13.

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 2	

2. OBJET DU DOCUMENT

2.1. CONTEXTE

Les plateformes de calcul HPC (Calcul Haute Performance) du CNES, actuellement les clusters HAL et Ktulu du projet HPC5G, sont des environnements partagés. Il sont donc soumis à un référentiel d'exigences, un "code de la route", que tout usager s'engage à respecter. L'objectif est de garantir à tout utilisateur, à tout projet un fonctionnement optimal en termes de performance et de continuité de service.

En effet, le non respect de ces règles peut rapidement conduire à une situation "d'embouteillage" ou "d'accident". C'est pourquoi, il y a sur les clusters des "radars automatiques" permettant à l'équipe support de prendre contact avec les utilisateurs afin de résoudre avec eux leur problème ou les abus.

Sans poursuivre l'analogie jusqu'au "permis à point", le non-respect des règles pourra conduire à la mise en place d'une limitation de ressources.

Il est important de noter que le grand nombre des utilisateurs du cluster implique pour l'équipe Calcul de n'avoir qu'un seul point de contact par projet (ou un point de contact principal) ou le cas échéant par service/structure. Cette personne est désignée comme responsable projet dans la suite du document (en référence au responsable projet figurant sur le formulaire d'ouverture de compte). Cependant, tous les utilisateurs de la plate-forme seront informés des annonces majeures par la liste de diffusion dédiée.

2.2. QUELQUES NOTIONS SUR LE CALCUL PARALLELE

Un cluster de calcul est une plateforme qui consiste à agréger la puissance de calcul d'entités de traitement unitaires pour virtuellement former un supercalculateur.

Dès lors, il incombe au développeur de distribuer efficacement ses traitements sur ces unités de traitement selon la topologie hiérarchique suivante : les serveurs du cluster, les cœurs du serveur et enfin les unités vectorielles des cœurs de calcul. On peut aussi ajouter un dernier niveau consistant en la présence éventuelle d'accélérateurs (GPU, FPGA, etc.).

Cette distribution de traitement, ce parallélisme, peut être pris en charge à plusieurs niveaux :

- le premier, consiste à déporter la gestion du parallélisme sur le job scheduler : le code de calcul est séquentiel et se borne à traiter sa donnée d'entrée. La parallélisation consiste à répéter le processus sur autant de données d'entrées que disponibles. Chaque job n'utilise qu'un seul cœur de calcul. Ce type de parallélisme est souvent déconseillé car son passage à l'échelle peut être sous optimal en surconsommant les ressources partagées (le stockage en particulier). Le minimum dans ce cas de figure est d'utiliser la fonctionnalité de job arrays proposée par l'ordonnanceur et d'avoir des jobs individuels suffisamment long.
- le second est une optimisation du premier cas : un parallélisme est implémenté dans le code ou l'une de ses bibliothèques pour utiliser plusieurs cœurs de calcul (OpenMP, pthread, TBB, Python multiprocessing, etc.). Chaque job peut alors utiliser de manière optimale un nœud de calcul entier, ou au moins une portion conséquente de celui-ci, pour chaque job. Là aussi, l'utilisation de job arrays est conseillée.
- enfin le troisième, **conseillé**, consiste à prendre en charge le parallélisme au niveau du code en se

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 3	

basant sur un environnement d'exécution distribué (MPI, Spark, Dask, etc.). Le développeur a la main pour optimiser l'enchaînement des différentes tâches constituant sa chaîne de traitement, optimiser le placement des données et l'utilisation des ressources de calcul. Chaque job utilise dès lors plusieurs serveurs de calcul. Attention néanmoins à la configuration de l'environnement d'exécution pour éviter de retomber dans un cas surconsommant les ressources partagées (des process n'utilisant qu'un cœur).

Il est donc indispensable que chaque projet, chaque utilisateur passe par une phase d'étude approfondie sur le type de parallélisme cible, la décomposition et le placement de ses données, le séquençement des tâches avec la synchronisation adéquate avant de commencer à utiliser un cluster de calcul.

Le pôle HPC du CNES peut vous aider dans cette phase d'analyse. N'hésitez pas à demander du [support](#).

2.3. VALIDITE DU DOCUMENT

La plateforme de calcul étant un moyen de production dynamique, son écosystème évolue régulièrement. Ce document se voulant en être le reflet, il sera régulièrement mis à jour.

Il importe donc de se référer à la dernière version disponible au moment d'une consultation, d'une conception ou d'une mise en exploitation d'un traitement. Afin de s'en assurer, il est possible de contacter L-SIS-poleHPC@cnes.fr.

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 4	

3. PLANIFICATION DE LA VOLUMETRIE

Les moyens de calcul mutualisés sont de fait partagés par l'ensemble des utilisateurs. A ce titre, la qualité de service offerte dépend du niveau d'information remonté par les projets de manière à permettre à l'équipe Calcul d'anticiper et si possible de lisser les pics d'activité. Cela peut également permettre l'allocation de ressources supplémentaires à un projet pendant un temps limité.

Il est donc attendu de chaque responsable de projet de prévoir annuellement la volumétrie de consommation de son projet en terme de :

- profil de jobs (nombre de coeurs et mémoire consommés par job)
- nombre d'heures de calcul
- volume de disque de stockage (NAS ou Capacitif : pour le code lui-même, les scripts et les données à conserver).
- volume de disque de calcul (haute performance, pour les données d'entrées/sortie et temporaires)

Le responsable du projet bénéficiera pour l'aider dans sa tâche d'un relevé semestriel de consommation, ce relevé peut être fourni mensuellement sous forme de rapport.

Afin d'optimiser la charge du cluster, il est demandé au responsable du projet d'indiquer à l'équipe Calcul les périodes de campagnes ou d'études à réaliser et de préciser les ressources nécessaires et s'il y a des contraintes de restitution des résultats. Ces informations permettent de répartir au mieux l'utilisation du cluster entre les différents projets.

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 5	

4. DISPONIBILITE ET SERVICES

4.1. ARRETS DE PRODUCTION

4.1.1. Plateformes de calcul

Afin de garantir un service fiable et performant aux utilisateurs, l'équipe Calcul se réserve une demi-journée par mois (le premier mardi du mois en règle générale, sauf contrainte planning) ou une journée tous les deux mois pour couper l'accès aux services afin d'effectuer des actes de maintenance nécessitant potentiellement de redémarrer ou réinstaller les composants des plateformes. Des réservations seront posées en avance de phase de manière à interdire toute exécution de jobs de calcul ce jour-là (les jobs en attente ne seront pas perdus et démarreront automatiquement une fois la maintenance terminée) sauf sur les ressources projets qui organisent en toute autonomie l'arrêt de leurs chaînes de traitements dans ces créneaux.

Cette action ne sera pas systématique et sera annoncée via la liste de diffusion propre au cluster le cas échéant.

La cible de disponibilité annuelle pour les plateformes de calcul HPC5G est de 98%, mais sans engagement formel de l'industriel en charge de l'exploitation. Ce chiffre a néanmoins été dépassé sur les années 2019, 2020 et 2021, arrêts de production et incidents compris.

4.2. ENGAGEMENTS DE SERVICES

4.2.1. Plateformes de calcul

Ce service est de classe SILVER (exploitation 10h/jour, 5j/7), 8H30-18H30. L'architecture de la plateforme a été conçue pour être hautement disponible : tous les composants critiques sont redondés avec bascule à chaud.

Engagements sur les demandes de services :

- Traitement des demandes sous 4 h pour les demandes opérationnelles (comptes ...)
- Traitement des demandes standards en moins de 48 h

4.3. CANAUX D'ADRESSAGE

Les ouvertures d'incidents ou de demande de support sont possibles via :

- Ma vie Numérique :
 - <https://mavienumerique.cnes.fr/index.php>

- Téléphone :
 - Le "40" (interne CNES)
 - Le "05 82 28 63 44" (extérieur CNES)

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 7	

5. GUIDE ET EXIGENCES

5.1. ORDONNANCEUR DE TRAVAUX

5.1.1. E-HPC-ORD0 : Runtime d'exécution parallèle

Un ordonnanceur de travaux n'est pas en premier lieu un runtime d'exécution parallèle. Le parallélisme de l'application ou de la campagne de traitement **doit être pris en charge par un framework dédié** et optimisé pour cette tâche (MPI, OpenMP, Dask, Spark, etc.) ou a minima en utilisant la fonctionnalité job arrays. Le comportement attendu est de réserver à travers l'ordonnanceur (PBS ou Slurm), XX nœuds/ressources de calcul, puis de gérer la parallélisation sur les nœuds et sur les cœurs avec le framework le plus adapté à votre contexte.

5.1.2. E-HPC-ORD1 : Réserve de cœurs de calcul

La granularité minimale de réservation doit être idéalement le nœud de calcul (24 cœurs en génération 2016, 40 en génération 2019). Cela présuppose que le code soit multithreadé ou multiprocessus, ce qui doit être la norme sur le cluster de calcul. Si par contrainte forte, cela n'est pas possible, il faut utiliser un nombre de cœurs multiple de 2 (idéalement 4, 8 ou 12).

5.1.3. E-HPC-ORD2 : Test de validation

Avant chaque campagne importante de calcul, des tests de validation doivent être effectués sur un échantillon réduit. En particulier, toute mise en œuvre de Job Array doit être testée sur un nombre limité de sous-jobs (10 ou 100). Il est fortement recommandé d'effectuer ces tests sous la supervision de l'équipe support.

5.1.4. E-HPC-ORD3 : Job Array

Dans le cas de campagne de calcul (retraitement, études paramétriques, etc.) nécessitant la soumission d'un nombre important de jobs de calcul, il est demandé de mettre en œuvre un job array (cf. wiki Calcul).

5.1.5. E_HPC-ORD4 : Fréquence d'appel de commande Ordonnanceur

La fréquence d'utilisation des commandes de l'ordonnanceur ne doit pas être inférieure à la minute. Par exemple, il est interdit de monitorer la fin d'exécution de la fin d'un jobs avec une boucle `qstat`. Veuillez utiliser pour ce faire l'envoi de mail automatique par l'ordonnancier (cf. wiki) ou bien un mécanisme de "broker" de message (RabbitMq, ZeroMQ, etc.) ou équivalent.

5.1.6. E-HPC-ORD5 : Environnement d'exécution

L'environnement d'exécution des jobs doit être spécifique aux traitements effectués. Il ne faut pas charger

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 8	

l'environnement utilisateur `.bashrc` dans les Jobs. Souvent l'environnement utilisateur charge des modules (tous les N jobs) qui sont inutiles pour le traitement des jobs et surcharge inutilement l'environnement. Il faut se référer au Wiki Calcul pour configurer correctement son `.bashrc`.

5.1.7. E-HPC-ORD6 : Temps d'exécution minimal

Le **temps d'exécution minimum admis pour un job est de 10 minutes**. Nous préconisons néanmoins une durée minimale de 30min et dans l'idéal 1h. Un job de calcul s'exécute en mode asynchrone sur des ressources de calcul partagées. Chaque job de calcul est géré indépendamment par l'ordonnanceur de travaux (même si dans un JobArray). Ainsi la multiplication de jobs de très courte durée a un impact négatif (surcharge) sur les performances globales de l'ordonnanceur de travaux et donc sur tous les utilisateurs.

De plus, sachant que les cycles d'ordonnancement peuvent aller jusqu'à 20 min et que les temps d'attente en file peuvent atteindre plusieurs heures, **le lancement en batch d'un job très court est inefficace**. Si l'algorithme de traitement à des temps d'exécution très courts, il est attendu un regroupement de plusieurs exécutions au sein d'un job de calcul. Le traitement de 10 000 jobs de 100 minutes sera plus rapide que celui de 100 000 jobs de 10 minutes.

5.1.8. E-HPC-ORD7 : Réservation optimale des ressources

L'objectif est d'éviter la fragmentation du cluster (une partie des ressources non utilisables sur les nœuds de calcul) et de minimiser les effets de bord entre différents jobs. Comme rappelé plus haut, l'unité de réservation atomique idéale est 1 nœud complet, soit 24c + 120G de mémoire pour un nœud de génération 2016, ou 40c + 180G de mémoire pour un nœud de génération 2019. Mais tous les codes de calcul ne sont pas suffisamment parallèles pour tirer partie de ces ressources. Dans ce cas, il est demandé de suivre les règles suivantes :

- Veillez à ce que les ressources demandées soient réellement utilisées par le code. Dans la négative, veuillez réserver moins de ressource.
- Les jobs doivent réserver des capacités qui sont une racine de la capacité totale d'un nœud. Il est donc important de respecter le ratio cœurs/mémoire suivant:
 - 1 cœur pour 5 Go de RAM (g2016)
 - 1 cœur pour 4.6 Go de RAM (g2019)
- Les réservations optimales après la réservation de nœud complet sont 8c + 36GB, 4c + 18GB ou 12c + 54GB.
- La réservation mémoire par cœur doit être comprise entre 2GB et 10GB maximum.

5.1.9. E-HPC-ORD8 : Écriture des fichiers de sortie

Il ne faut pas écrire les fichiers de sortie au même endroit (même chemin absolu) pour plusieurs jobs ou sous-jobs. Typiquement, il est tentant d'écrire les fichiers de sortie au même endroit pour tous les sous-jobs d'un job array. Cela entraînera des conflits d'écriture et des erreurs à grande échelle qui auront un impact non négligeable sur l'ordonnanceur, pouvant aller jusqu'à un incident majeur. Il faut impérativement vérifier avant la soumission d'un grand nombre de jobs qu'il n'y ait aucun risque d'écriture sur les espaces de destination (existence du dossier, accessibilité, aucun risque de conflit d'écriture, marge sur l'espace disponible, ...).

Si nécessaire, l'option `-k` peut éviter de récupérer les fichiers de sortie ou permettre d'écrire directement sur

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 9	

l'espace de destination (plus d'erreurs de copie dans ce cas mais à réaliser en dernier recours) :

- -k e pour éviter de récupérer le fichier erreur
- -k o pour le fichier de sortie
- -k oe pour le deux
- -k d pour écrire directement les fichiers de sortie sur l'espace de destination

5.2. STOCKAGE ET ACCES AUX DONNEES

"A supercomputer is a device for turning compute-bound problems into I/O-bound problems." Ken Batcher

Dans ce chapitre, I/O signifie opération d'entrée/sortie sur disque (un simple accès de lecture à un fichier se décompose en plusieurs IO au niveau de la couche matérielle).

5.2.1. E-HPC-DAT-0 : Accès optimisé aux données

- *Type de fichier* : Un fichier hébergé sur l'espace `/work` devrait être un fichier de données. Un fichier de données devrait être stocké sous forme binaire (pas de fichier de données textuel, ou alors a minima compressé). Les meta-données devraient être stockées dans une base de données.
- *La lecture d'un fichier* sur le `/work` doit se faire en une seule étape : le fichier ne doit être accédé qu'une seule fois au cours du calcul. Si de multiples accès doivent être réalisés, ils doivent se faire en mémoire. Si cela n'est pas possible (multiple accès disque en lecture), le fichier doit préalablement avoir été copié sur l'espace local au nœud de calcul (`$TMPDIR`) afin de limiter la fréquence d'accès à la baie de stockage haute performance.

5.2.2. E-HPC-DAT1 : Limitation des flux d'IO par nœud.

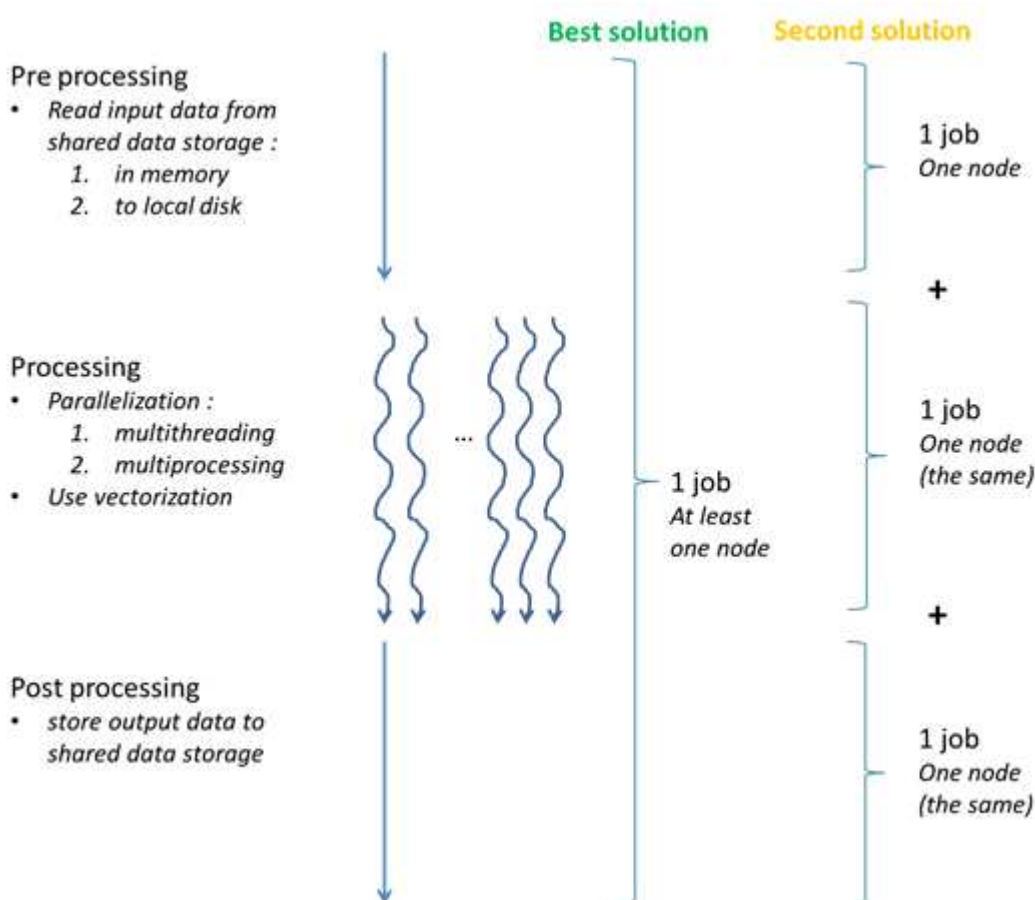
La règle idéale étant 1 job = 1 nœud = 1 flux d'IO. Néanmoins, il est souvent difficile, voir contre performant de suivre cette règle. Si l'accès aux données respecte les bonnes pratiques (lecture séquentielle par blocs suffisamment gros) une telle limitation n'est pas nécessaire. Il est demandé de faire le maximum pour suivre les règles suivantes :

- Un seul flux d'IO / nœud de calcul dans le cas d'un code mal optimisé. Cela signifie que depuis un nœud, il ne peut pas exister plus d'un processus accédant aux données présentes sur `/work`. Si besoin, cela peut être effectué via copie préalable des données dans l'espace temporaire local au nœud sur lequel le calcul est réalisé.
- Limitation des flux d'IO par nœud : un flux d'IO par job, avec des jobs qui respectent les règles précédemment établies.
- Dans tous les cas, un flux d'IO maximum par cœur de calcul, et cela n'est possible qu'avec un code qui fait des accès correctement optimisés à la donnée.

5.2.3. E-HPC-DAT2 : Bufferisation des données.

Penser à bufferiser un maximum de données en mémoire (ie charger l'intégralité des données pour des accès ultérieurs). Par exemple un fichier de données ne doit être accédé qu'une seule fois sur disque au cours du calcul (les autres accès étant effectués en mémoire). Cela permet de réduire le nombre d'accès au stockage partagé.

5.2.4. E-HPC-DAT3 : Workflow de traitement de la donnée optimal



Idéalement le chronogramme IO attendu pour un job de calcul est le suivant :

1. Accès en lecture aux données sur /work
2. Phase de calcul en mémoire (la plus longue)
3. Écriture des résultats sur disque sur /work

Si la bufferisation en mémoire ne peut pas être totale, la phase 2 peut accéder au disque local du nœud de calcul (\$TMPDIR).

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 11	

5.2.5. E-HPC-DAT4 : Un filesystem n'est pas une base de donnée.

Tous les produits d'un projet, de même que leurs méta-données doivent être indexés dans une base de données.

Toutes les recherches/parsage de données doivent être faits dans la base de données. En particulier, il est formellement interdit de faire des opérations de recherche de type `find/grep` sur des fichiers/répertoires *au cours des traitements* sur le filesystem partagé `/work`.

5.2.6. E-HPC-DAT5 : Synchronisation des tâches : système d'échange de message.

Il est formellement interdit de synchroniser des tâches sur des conditions d'existence et d'état de fichier. De même il est interdit d'échanger de l'information entre deux serveurs en utilisant le filesystem. Privilégiez l'utilisation du réseau, de bibliothèques de type MPI/Dask ou des outils de type broker de message (ZeroMQ, RabbitMQ, etc.).

5.2.7. E-HPC-DAT6 : Limite sur le nombre de fichiers

Un répertoire ne doit pas contenir plus de 10 000 fichiers afin de permettre de conserver des performances d'accès à la donnée optimale. Au dessus de cette valeur, vous pourriez observer des ralentissements dans vos traitements.

5.2.8. E-HPC-DAT7 : Limite sur le nombre de sous répertoires

Un répertoire ne doit pas contenir plus de 10 000 sous répertoires afin de permettre de conserver des performances d'accès à la donnée optimale. Au dessus de cette valeur, vous pourriez observer des ralentissements dans vos traitements.

5.2.9. E-HPC-DAT8 : Interdiction de caractères interprétés dans les noms de fichier/répertoires

Les noms de fichiers ou répertoires ne doivent pas contenir les caractères spéciaux suivants `?, $, \, %, ^M`.

5.2.10. E-HPC-DAT9 : Type de fichier et volume

Le `/work` ne doit héberger que des **fichiers de données binaires** de **taille supérieure à 1Mo**. Essayez de suivre au maximum cette règle. En particulier limitez au maximum tous les fichiers textuels, et les petits fichiers. Si c'est inévitable, il est donc exigé de procéder à leur compression/archivage automatiquement (1/jour ou /semaine en fonction du contexte).

5.2.11. E-HPC-DAT10 : Gestion des fichiers

- *fichiers temporaires* : les fichiers temporaires (ex : `.log`, `.err`, `.out`, etc.) doivent être régulièrement nettoyés.

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 12	

- *fichiers anciens* : les fichiers qui ne sont plus accédés depuis plusieurs mois ne doivent pas *confisquer* du volume de stockage sur un espace orienté traitement. Ils doivent à minima être triés (*sont-ils toujours nécessaires ?*), compressés (.zip), et enfin transférer sur une infrastructure de stockage long terme.

5.2.12. E-HPC-DAT10 : Format de compression de fichiers

Éviter l'utilisation des anciens formats de compression orientés bandes : tar, tar.gz. Privilégiez l'utilisation de format de type zip. En effet, ce format est indexé. Dans le cas de la compression/archivage de plusieurs fichiers, il est possible d'accéder par la suite directement au fichier souhaité sans décompresser l'ensemble de l'archive.

L'outil pigz/unpigz (qui utilise le format gzip) est disponible sur le cluster pour faire de la compression en multithreading. Faire "man pigz" pour avoir les options, par défaut la commande pigz en batch utilise le nombre de cores réservés pour le job.

5.2.13. E-HPC-TMP1 : Accès à l'espace local des nœuds de calcul

Sur les nœuds de calcul, l'accès direct à l'espace `/tmp` est interdit. Vous devez utiliser la variable PBS `$TMPDIR` qui pointe vers un sous répertoire temporaire du `/tmp` géré par l'ordonnanceur (et nettoyé en fin de job). Cela évite la saturation de cet espace et le plantage des prochains jobs.

5.2.14. E-HPC-NFS1 : Limitation du protocole NFS

Le protocole NFS ne doit pas être utilisé pour faire de l'accès intensif au stockage. Il est mis à disposition des projets sur des VMs ou serveurs dédiés projet afin d'accéder aux données mais pas pour les manipuler. Toutes lectures/écritures intensives doivent être réalisées dans un job sur le cluster de calcul afin d'utiliser le protocole natif de GPFS.

5.3. PLANIFICATION DES CAMPAGNES DE CALCUL

5.3.1. E-HPC-PLAN1

Toute utilisation intensive de la plateforme de calcul (campagne de rattrapage, de retraitement, importante simulation numérique, benchmark) doit être analysée et validée au préalable par le supportHPC.

5.3.2. E-HPC-PLAN2

Toute utilisation intensive de la plateforme de calcul (campagne de rattrapage, de retraitement, importante simulation numérique, benchmark) doit être communiquée un mois à l'avance à l'équipe de supportHPC.

L'objectif est d'arriver à déphaser les pics de charge entre les projets (en prenant en compte les contraintes calendaires de chacun).

Tous les utilisateurs et projets ont la même priorité. Si la planification se fait dans les temps impartis, il est possible de demander une priorité plus élevée ou un volume de ressources supérieur pour maximiser la

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 13	

bande passante de calcul du projet. Toutefois, cette élévation de priorité ne pourra être mise en place que pour les chaînes ayant une utilisation nominale des ressources du cluster (IO, calcul) et ayant été analysées/validées au préalable par le supportHPC.

5.4. ENVIRONNEMENT

5.4.1. E-HPC-OS-1 : Version majeure d'OS

L'OS cible des plateformes de calcul HAL et Ktulu (HPC5G) est CentOS7. L'OS cible de la plateforme HPC6G (fin 2022) sera RedHat 8.

5.4.2. E-HPC-OS-2 : Version mineure d'OS

Les codes de calcul doivent être compatibles avec les évolutions mineures de la version de l'OS (7.X ou 8.X).

Ainsi, l'installation ou l'exécution d'un logiciel ne doit pas être impactée par une montée de version mineure comme par exemple 7.6 à 7.8. A noter qu'il n'est pas donné d'indication sur la version mineure de la version de Red Hat installée sur le cluster. L'équipe Calcul se doit de maintenir ses systèmes à jour afin de pallier à tout risque de faille de sécurité ou pour combler tout risque de bug système. L'interface entre le code et le système d'exploitation devra donc se faire au moyen d'appels système standards, se baser sur des API standards à des bibliothèques systèmes communes.

Dans le cas contraire, il incombe au projet de packager et d'embarquer les bibliothèques spécifiques au niveau de son logiciel, pour un déploiement local dans son espace projet/utilisateur ou bien de réserver des jours de TMA spécifiques pour prendre en charge les éventuelles corrections à apporter au code ainsi que les tests de non régression.

5.5. DEVELOPPEMENT ET OPTIMISATIONS

5.5.1. E-HPC-DEV-1 : Langages de programmation préconisés

Les langages préconisés pour les codes de calcul sont :

- C
- C++
- Fortran
- Python
- Julia

Les versions à jour de ces langages sont disponibles dans le /softs. Avant de démarrer un nouveau développement, il faut vérifier les versions présentes. Il s'agit de sélectionner une version déjà installée ou alors de constituer son propre environnement logiciel si cela est possible.

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 14	

5.5.2. E-HPC-DEV-2 : Version langage, compilateur, librairie

Dans le cas d'un nouveau développement, il faut s'orienter vers la dernière version installée. Si les versions installées ne conviennent pas, toute demande d'installation d'une nouvelle version devra être justifiée pour une installation dans l'espace mutualisé. Dans le cas contraire, le projet devra gérer l'installation en local dans son espace projet/utilisateur.

5.5.3. E-HPC-DEV-3 : langage compilé

Les cœurs de calcul, attendus comme les plus consommateurs en temps d'exécution, doivent être développés ou reposer sur un langage dont la compilation pourra générer un code assembleur optimisé pour l'architecture du CPU.

Les langages interprétés sont réservés aux couches hautes du programme (manipulation/préparation des données, interface utilisateur par ex.).

Des librairies de langage interprétés mais avec un cœur en langage compilé comme Numpy ou Pandas conviennent pour les cœurs de calcul.

5.5.4. E-HPC-DEV-4 : Compilation

Pour les langages C, C++ et Fortran, il est fortement recommandé d'utiliser l'un des deux compilateurs suivants dans la dernière version disponible sur le cluster :

- compilateur GNU
- compilateur Intel

Les développements doivent être compatibles avec la suite de compilateurs GNU et/ou de la suite de compilateurs Intel.

La version minimale de la suite GNU GCC à prendre en compte est la version GCC native de la version Red Hat utilisée (/usr/bin/gcc --version). Il est possible d'opter pour des versions plus récentes de cette suite de compilateurs dans le répertoire /softs. Le choix de cette suite de compilateurs est fait vis-à-vis du choix d'une solution open source et non commerciale.

L'autre suite de compilateur à utiliser est la suite de compilateurs Intel. Cette suite de compilateur nécessite une licence à la compilation mais pas à l'exécution. Elle peut donc être utilisée sous réserve de la disponibilité de cette suite commerciale sur le site de développement ou sous condition que le CNES puisse donner un accès à ses ressources Linux proposant cette suite de compilateurs. Le choix de cette suite de compilateurs est fait vis-à-vis de son positionnement en termes de performance sur les infrastructures HPC du CNES. Les versions à jour sont disponibles dans le /softs.

5.5.5. E-HPC-DEV-5 : Bibliothèques mathématiques

Les noyaux de calcul présents dans le code devront s'appuyer sur les bibliothèques mathématiques admises comme standards. Il faut utiliser, dans la mesure du possible, la dernière version des bibliothèques installées sur le système ou sous l'espace des logiciels.

Un ensemble de bibliothèques mathématiques sont disponibles dans le /softs. Parmi ces bibliothèques, il y a :

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 15	

- LAPACK : une bibliothèque reposant sur BLAS proposant des routines de résolution de systèmes linéaires, de problèmes de moindre carré, de valeurs propres, de décomposition en valeurs singulières et de factorisation de matrice.
- BLAS : bibliothèque incluant des routines de calcul d'algèbre linéaire de base (opérations entre matrices et/ou vecteurs).
- Intel MKL : bibliothèque développée par Intel (BLAS, LAPACK, FFT, Statistique, ...)
- FFTW : bibliothèque de calcul des transformées de fourrier (2D, 3D).
- Atlas (Automatically Tuned Linear Algebra Software) : Bibliothèque d'algèbre linéaire.

5.5.6. E-HPC-DEV-6 : Optimisation des performances et validation

Les développements doivent prendre en compte la problématique d'optimisation des performances. Il s'agit de développer un logiciel dans le but d'exploiter au mieux les ressources matérielles mises à disposition.

Pour tout code de calcul lancé en mode batch, une analyse de profiling doit être réalisée sur le logiciel pour identifier les portions de code les plus consommatrices.

Une liste d'outils de contrôle de code préconisés est disponible dans l'espace des logiciels :

- Gprof
- Valgrind
- Maqao
- Vtune

Lors de la livraison, une analyse sur la consommation des ressources matérielles doit être fournie. Une analyse détaillée des entrées sorties devra être livrée présentant un graphe du volume de données lues et écrites en fonction du temps.

Lors des phases de livraison de versions majeures du code de calcul, une analyse de consommation des ressources CPU / mémoire / stockage devra être fournie pour les cas tests de référence. En particulier tout profil d'exécution anormal (forte charge I/O, sous-utilisation du CPU, mauvaise scalabilité, etc.) devra être justifié.

5.5.7. E-HPC-DEV-7 : Checkpoint/Restart

En cas de temps d'exécution d'une durée de plus de 3 jours, les développements doivent intégrer des procédés de sauvegarde et de reprise d'exécution (checkpoint/restart) sur disque. Ces procédés permettent de ne pas perdre l'avancée de travaux en cas de dysfonctionnements/pannes matériels.

Les procédés de sauvegarde/reprise implémentés au niveau applicatif permettent de pallier aux pannes matérielles dont la probabilité augmente avec la durée de temps de traitement ou avec l'utilisation d'un nombre croissant de serveurs de calcul. De plus, cela est rendu nécessaire dans le cadre des opérations de maintenance mensuelle.

CNES Guide de bon usage et d'intégration sur les services du Centre de Calcul du CNES	Edit. : 01	Date : 25/10/2021
	Rév. : 00	Date : 25/10/2021
Référence : DNO/ISA/CID-2021.0014206	Page : 16	

5.6. NŒUDS FRONTAUX

Les nœuds frontaux sont les portes d'entrée du cluster. Ces derniers sont donc plus sécurisés que les autres nœuds du cluster et portent un ensemble très réduit de service :

- rebond : A travers le menu de connexion, ils permettent aux utilisateurs de rebondir vers les nœuds interactifs/visualisation du cluster
- transfert : Ils hébergent un serveur sftp permettant aux utilisateur de transférer des données sur les espaces de stockage du cluster.

5.6.1. E-HPC-FRT1 : Limitation des nœuds frontaux

Toute autre utilisation que les fonctions de rebond ou de transfert est interdite sur les frontaux du cluster, en particulier via des exécution de script via ssh (ssh -c).

5.7. NŒUDS DE VISUALISATION

5.7.1. E_HPC_VIS

Il est **interdit de lancer de lourdes charges de calcul sur les nœuds de visualisation.**

En effet ces nœuds sont partagés, leur rôle principal est d'héberger des activités de développements, tests légers, soumission de travaux. Tout calcul prolongé (>1h) doit être lancé dans un job de calcul batch. Nous vous référons aux instructions de batch et aux exemples de scripts disponibles dans la documentation utilisateur.

Si vous devez lancer un processus de calcul lourd en mode interactif, nous vous conseillons d'ouvrir une session interactive sur un nœud de calcul

5.8. CRONTAB

5.8.1. E-HPC-CRON1

Toute chaîne lancée par `crontab` doit débuter par un test d'existence d'une éventuelle précédente exécution qui ne serait pas terminée.

L'objectif est d'éviter un phénomène d'empilement de processus (précédente chaîne n'ayant pas eu le temps de se terminer ou précédente chaîne étant partie en erreur).

5.8.2. E-HPC-CRON2

Des machines virtuelles sont dédiées aux besoins de mise en place de CRON. Au préalable demander l'accès à ces VMs au support calcul.

CNES

**Guide de bon usage et d'intégration sur les services
du Centre de Calcul du CNES**

Edit. : **01**

Date : 25/10/2021

Rév. : **00**

Date : 25/10/2021

Référence : **DNO/ISA/CID-2021.0014206**

Page : 17

*** * * FIN DU DOCUMENT * * ***