



Bilan Campagne Performance DILE - REU Février – Mars 2019

DILA
DILE – REU

Sommaire

1	Introduction	3
1.1	OBJET DU DOCUMENT	3
1.2	VERSION	3
2	Rappel des principes d'injection de la charge	4
3	Comportement du front SP/PSL	5
3.1	DEBIT RESTITUE	5
3.2	TRAFIC RESEAU AVEC LA BASE DE DONNEES	6
3.3	ANOMALIE DES PJ PDF	8
3.4	PAGE DE DELESTAGE	8
4	Comportement M6G : GS et ORCHESTRA	9
4.1	PERFORMANCES BRUTES GS	9
4.2	STABILITE D'ORCHESTRA	10
4.3	PERFORMANCES D'ORCHESTRA ET FLUX « RETOUR »	11
4.4	PATCH ORCHESTRA	13
4.5	BASES DE DONNEES MYSQL GS & ORCHESTRA	14
4.6	PURGE DU NAS GS	14
4.7	MISE EN PLACE DES VIP HAPROXY	15
5	Comportement PEC	16
6	Synthèse	17

1 Introduction

1.1 Objet du document

Ce document présente les travaux menés lors de la campagne de performances DILE – M6G – PEC - REU réalisée entre le 4 Février 2019 et le 8 Mars 2019. L'objectif de cette campagne est d'évaluer au mieux les points de contention de l'ensemble de la chaîne, de bout en bout, depuis un dépôt de TD utilisateur sur la PSL jusqu'à l'envoi à l'INSEE (avec et sans flux retour), et de proposer des axes d'amélioration en collaboration avec tous les acteurs (DILA, TMA, EXPLOITATION) autant sur les axes applicatifs que sur l'infrastructure.

L'objectif est de favoriser les meilleures performances du système en vue des élections européennes à venir et des élections futures.

La philosophie du document est de proposer une synthèse pertinente du comportement de chaque sous système et ne présente pas, de manière séquentielle, l'exhaustivité des tirs (18 tirs ont été exécutés au total).

Une synthèse est proposée au § 6 pour les lecteurs souhaitant disposer plus rapidement des éléments de restitution.

1.2 Version

Version	Date	Auteur	Commentaire
1.0	13/03/2019	Jean-François RIVIERE	Version finale

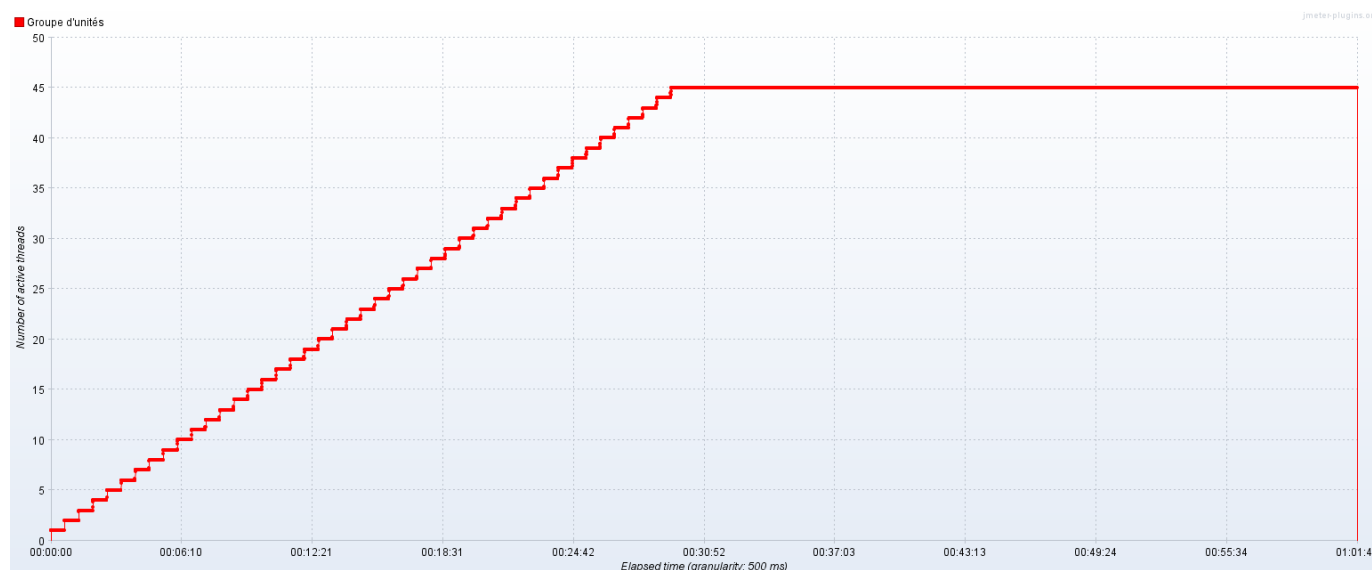
2 Rappel des principes d'injection de la charge

Les scripts JMeter, en version 5.0, (avec RNVP et sans RNVP) sont adaptés depuis les dernières campagnes (quelques soucis d'URL de PREP/PROD) et utilisés.

La version « sans RNVP » est liée à une licence réduite du service « RNVP » sur l'environnement de pré-production ne permettant pas de solliciter le service à un débit supérieur à 2 appels par seconde (soit 120 dem / min).

- La totalité de la charge JMeter (de 0 à 270 dem / min) est injectée en 30 ou 60 minutes selon les cas.
- Une équivalence threads JMeter / Débit est déterminée de manière empirique. (De 0 à 45 threads pour 0 à 270 dem / min).
- Les temps de pause sont réduits au maximum pour stresser le système de manière significative et limiter l'empreinte mémoire coté injecteur, qui peut alors perturber la mesure.

Le profil de charge est généralement le suivant :



Pour simuler les flux retour de l'INSEE en mode « bouchonné », un shell spécifique est développé et déposé sur le petit serveur d'injection de la PEC.

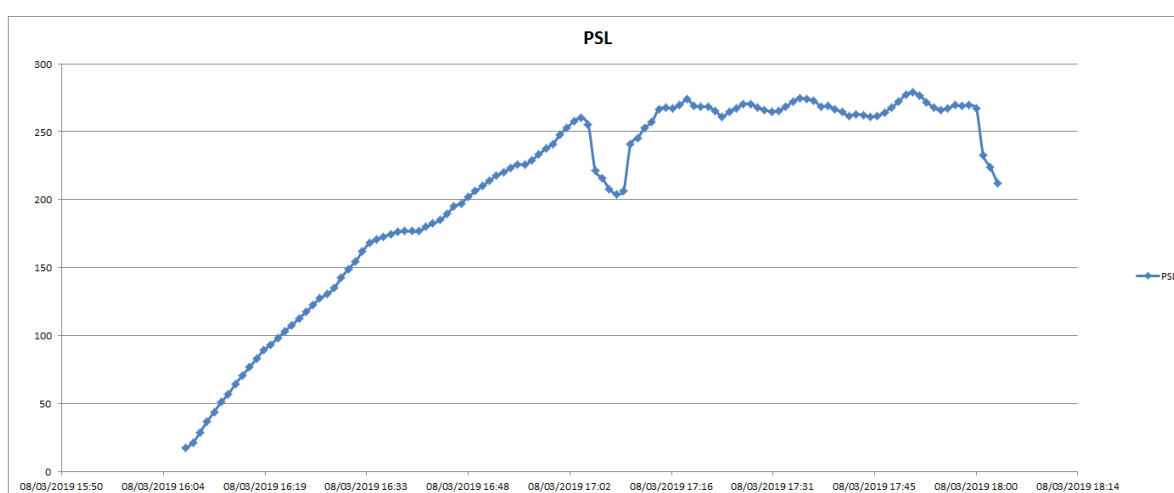
La taille moyenne de dossiers est de 1.5 Mo (unitairement entre 235 ko et 3.5 Mo).

3 Comportement du front SP/PSL

C'est le point d'entrée des démarches DILE et il est important que les performances soient les plus élevées possibles. Tous les tests ont montré un comportement globalement satisfaisant de SP et de PSL avec un débit maximal atteint de **279 démarches par minute** (pour 3 nœuds PART contre 4 en PRODUCTION).

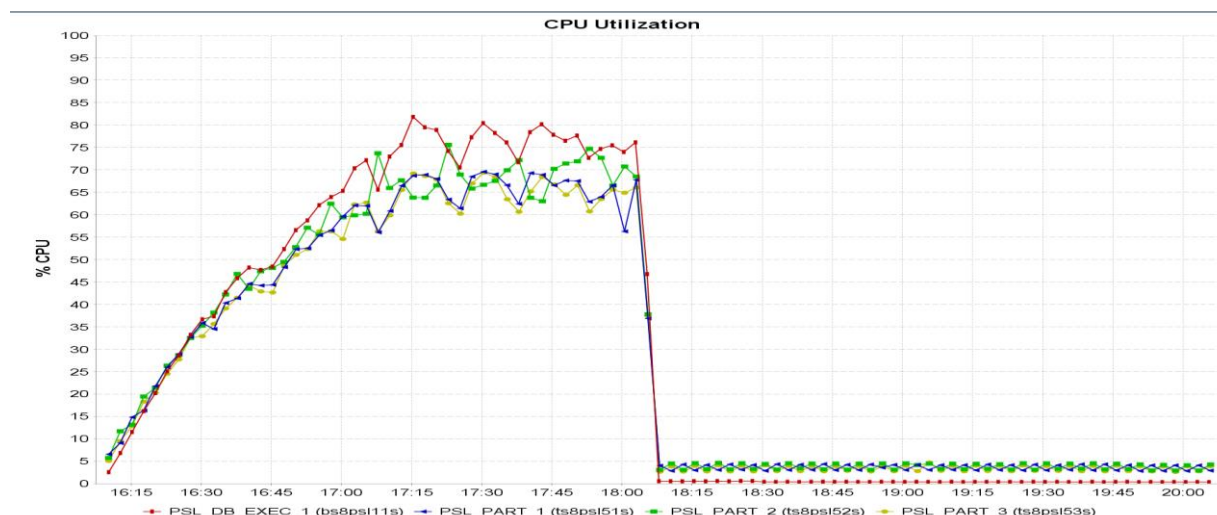
3.1 Débit restitué

Ci-dessus, le débit restitué lors d'un tir de 2h



On constate malgré tout sur ce tir (mais aussi sur d'autres) un petit ralentissement sur quelques minutes, avec une baisse du débit. Il n'y a pas d'explication claire sur cet évènement, le fait est que parfois les pages renvoyées par l'application ne sont pas totalement complètes (assertion réponse en erreur car il manque un libellé recherché) et alors JMeter considère cela comme une erreur et le script ne termine pas le cycle (pas de dépôt de TD). **On peut imaginer une petite contention réseau mais sans certitude.**

Et la consommation CPU sur les serveurs clés de la PSL très satisfaisante sur les 4 serveurs.



3.2 Trafic réseau avec la base de données

Les défauts déjà connus de la PSL sont toujours là, à savoir un nombre de requêtes sur la base de données gigantesque générant un trafic réseau très élevé. Heureusement les requêtes sont très rapides et s'exécutent en moins de 1 ms.

2/21/2019, 3:00 PM to 3:30 PM ▾

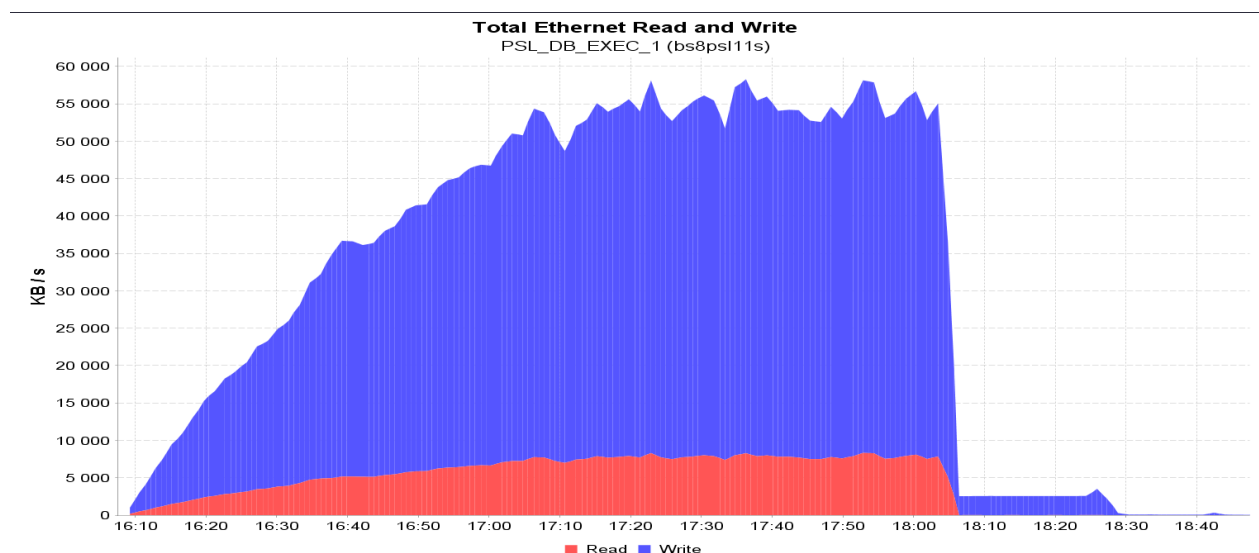
All Web Transactions

Response time Slow traces (81) **Queries** Service calls Thread profile

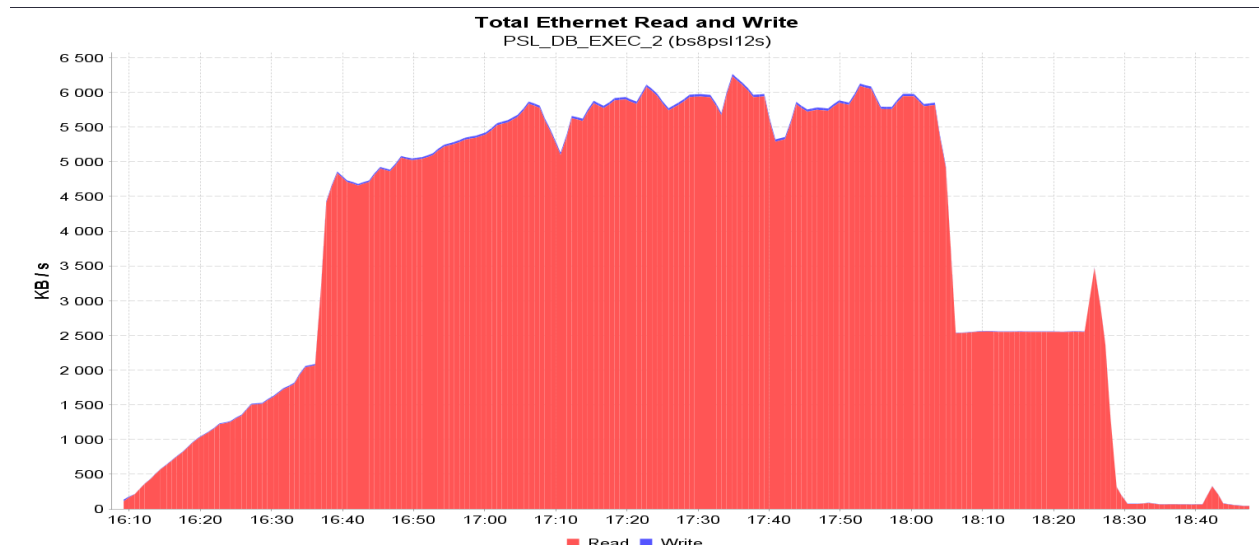
	Total time (ms)	Total count	Avg time (ms)	Avg rows
SELECT ds.simple_valeur_par_defaut, ds.simple_valeur_modifiee FROM exe_demarch...	1228371,7	851,161	1,4	1,0
select versiondep@_.VER_ID_TECH as VER_ID_T1_22_, versiondep@_.DATE_DEPLOIEMEN...	548721,9	462,588	1,2	1,0
select abstractdo@_.DSP_ID_TECH as DSP_ID_T2_17_, abstractdo@_.ACTIF as ACTIF3...	418484,6	392,516	1,1	1,0
select donneesmet@_.DME_ID_TECH as DME_ID_T1_5_0_, donneesmet@_.ACTIVATION_CPL...	280665,8	277,545	1,0	0,2
select demarche@_.DEM_ID_TECH as DEM_ID_T1_1_0_, demarche@_.CATEGORIE_USAGER a...	266454,3	277,545	0.96	1,0
select templatema@_.MAIL_ID_TECH as MAIL_ID_1_13_0_, templatema@_.CORPS_MAIL a...	65064,3	64,141	1,0	1,0
select donneesmet@_.DMG_ID_TECH as DMG_ID_T1_6_0_, donneesmet@_.GS_CODE_SOU MIS...	63568,6	64,141	0.99	1,0
select donneesmet@_.DMM_ID_TECH as DMM_ID_T1_7_0_, donneesmet@_.ACTIVATION_SUI...	60508,3	64,141	0.94	1,0
select userconver@_.USER_CONVERSATION_ID_TECH as USER_CON1_0_, userconver@_.en...	62522,0	29,569	2,1	1,0
select count(userconver@_.USER_CONVERSATION_ID_TECH) as col_0_0_ from EXE_USER...	19682,8	16,286	1,2	1,0
update EXE_USER_CONVERSATION_INFO set codeDemarche=?, conversationId=?, digest...	16062,4	15,870	1,0	1,0
select extensions@_.TypePieceJointe_TPJ_ID_TECH as TypePiec1_15_0_, extensions...	9252,2	9,920	0.93	4,0
select periodeval@_.DONNES_METIER_ID as DONNES_M4_5_0_, periodeval@_.PER_ID_TE...	8842,4	9,253	0.96	1,0
update EXE_USER_CONVERSATION set encryptedCurrentPageData=?, encryptedData=?, ...	17953,2	8,594	2,1	1,0
select count(PER_ID_TECH) from EXE_DM_DONNES_METIER_PERIODE_VALIDITE where DO...	8127,5	8,591	0.95	1,0

Par exemple, on mesure ici plus de **2.500.000 requêtes SQL** (select uniquement) exécutées en 30 minutes, **soit un débit de moyen de plus de 1.340 requêtes par seconde.....**

On constate également facilement ce souci lorsque l'on compare les consommations réseaux des deux nœuds BDD de la PSL, la base active (PSL_DB_EXEC_1 sur le bs8psl11s) et la base esclave (PSL_DB_EXEC_2 sur le bs8psl12s).



Series Name	Minimum	Average	Maximum	Std Dev
Write	1,200	26 912,408	61 114,400	19 756,487
Read	0,700	4 447,964	9 211,600	3 327,573

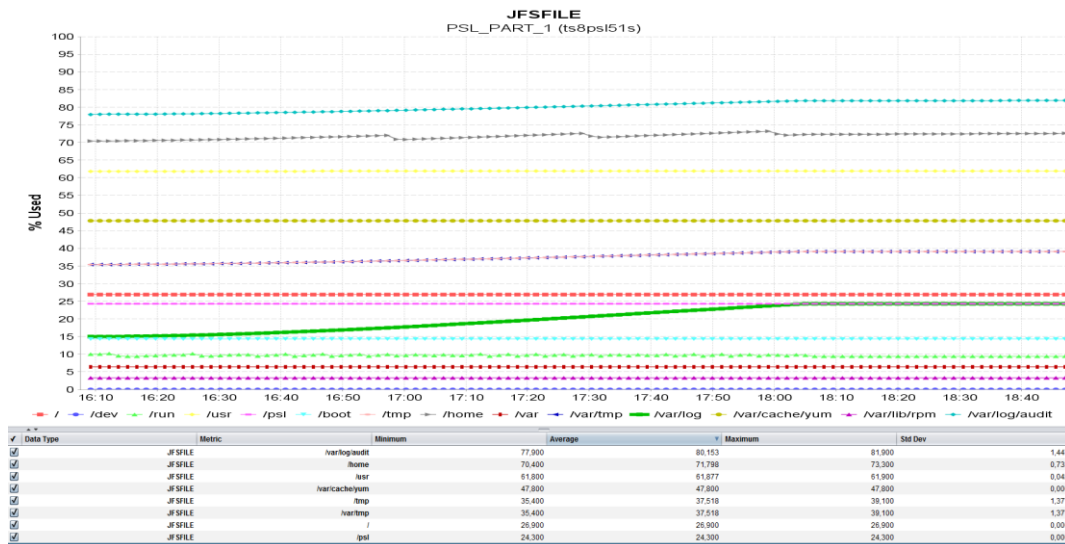


Series Name	Minimum	Average	Maximum	Std Dev
Read	0,600	3 660,981	7 848,500	2 289,095
Write	1,000	21,114	429,800	15,549

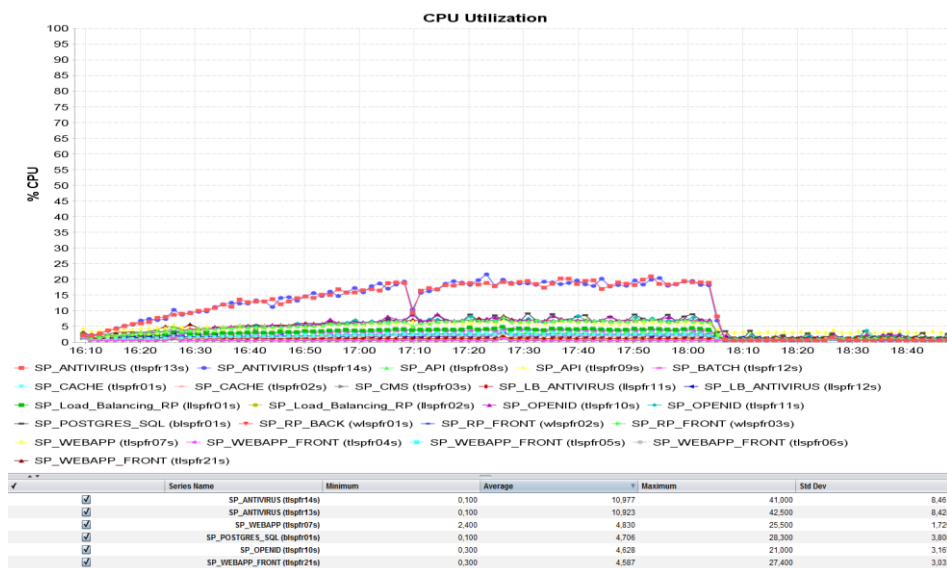
Il y a un facteur x10 entre les modifications sur la base (que l'on synchronise sur le nœud esclave...en rouge sur les graphes) et le trafic réseau lié à tous ces ordres SQL (en bleu).

3.3 Anomalie des PJ PDF

Concernant l'anomalie des « PJ pdf » et la génération d'une miniature sur le système de fichier qui n'était pas libéré, **cette anomalie a été corrigée** au vu du graphe suivant qui montrent les consommations d'espace disque sur un nœud PSL_PART.



Coté SP, rien à dire, à 270 dem / minute, les consommations CPU restent faibles sur l'ensemble de l'application.



3.4 Page de délestage

Des tests avec l'exploitant ont été réalisés afin de valider le fonctionnement de la page de délestage. Plusieurs tests avec des valeurs de 2000, 4000 et 6000 sessions sur une période de 900 s permettent de conclure **qu'une valeur autour de 6000 sessions** est un bon compromis.

Cela autorise donc une moyenne de 400 sessions par minute.

4 Comportement M6G : GS et ORCHESTRA

Les derniers tests de performance de Décembre 2018 ont montré que le point de contention principal se situe sur cette brique et en particulier sur ORCHESTRA.

De nombreux tests avec des configurations différentes ont été réalisés pendant cette campagne de test et force est de constater qu'il a été difficile d'améliorer significativement les choses ici.

Malgré tout, il ressort de cette campagne certaines certitudes. Les points à traiter étaient les suivants :

- Améliorer les performances en terme de dossiers / min de GS & ORCHESTRA.
- Réduire les locks base de données et les contentions lors des flux retours.
- Stabiliser cette brique autant que possible.

4.1 Performances brutes GS

Le suivi des performances de GS et ORCHESTRA est réalisé à l'aide de requêtes dans la base GS et des status 26 et 11.

Concernant GS, il est possible (avant application du patch ORCHESTRA) de cadencer la vitesse d'initialisation de GS (statut 26) en modifiant le paramétrage suivant

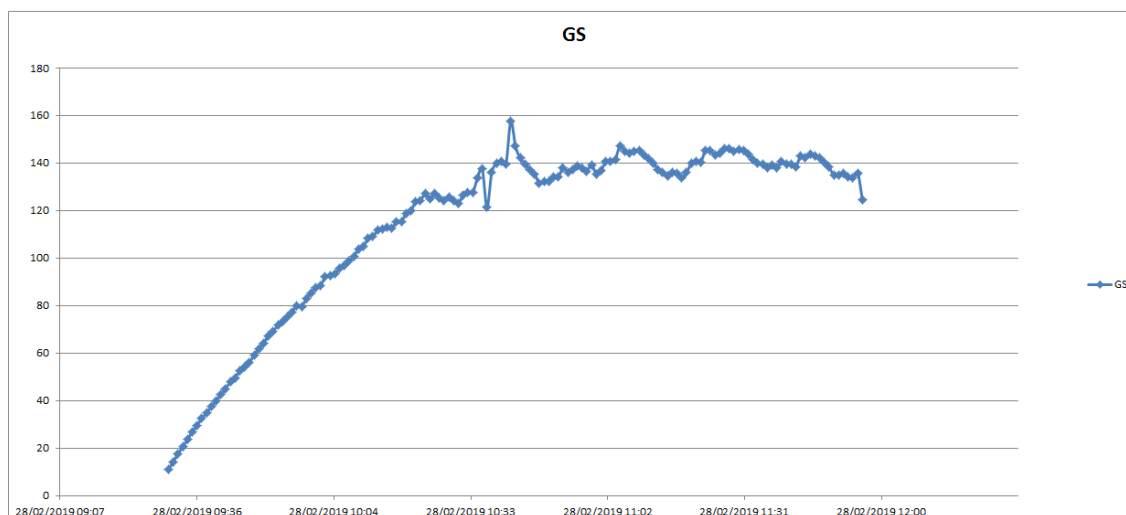
Batch d'initialisation

Activation du batch : Oui

Limite du nombre 100
d'enveloppes traitées :

Délai avant initialisation : 15

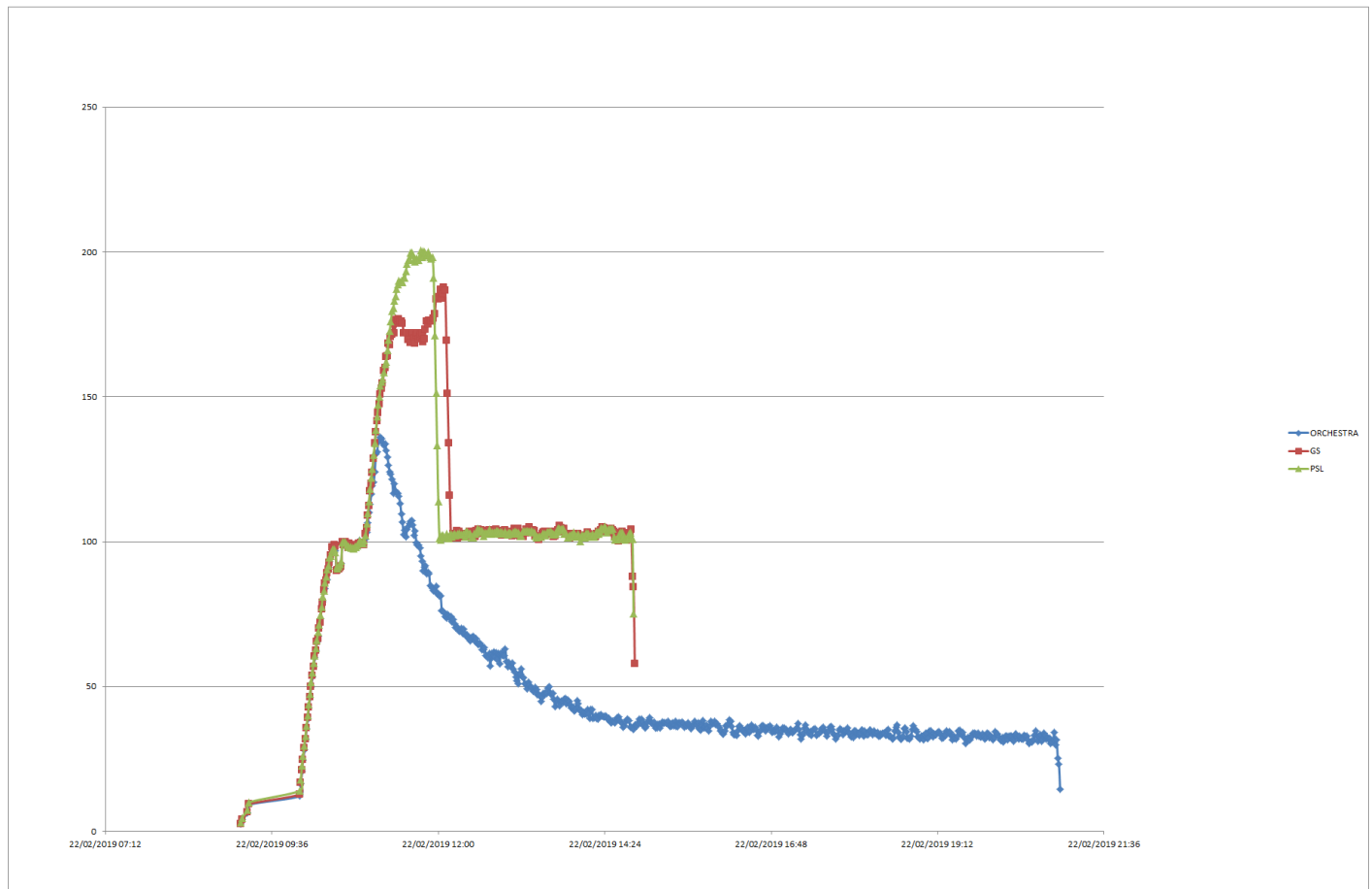
En jouant sur ces valeurs, il est possible d'obtenir un débit GS proche de 150 dem / min comme lors du tir avec le branchement sur l'INSEE.



Cependant, la contention principale se situe sur ORCHESTRA et il y a ici beaucoup de chose à dire.

4.2 Stabilité d'ORCHESTRA

Les tests ont clairement mis en évidence une anomalie sur ORCHESTRA lorsque le débit dépasse une valeur critique autour de 130 dem / min. La courbe suivant présente le sujet :



On réalise une injection en front PSL à 100 dem / min puis on réalise une « bosse » à 200 dem / min pendant une heure. On constate que GS suit globalement le débit, avec quelques ralentissement autour de 150 -180.

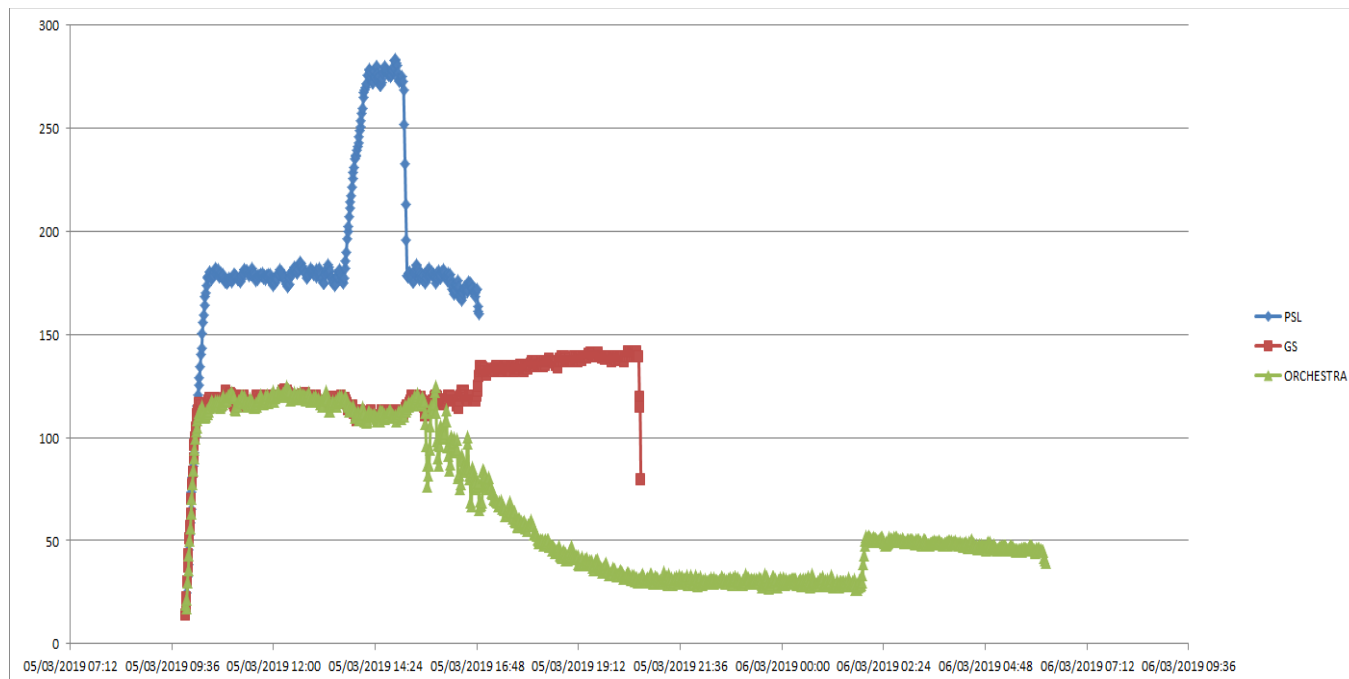
Mais on observe un point de rupture sur ORCHESTRA autour de 135 dem / minute. Lorsque ce point est atteint, les performances d'ORCHESTRA chutent irrémédiablement sans que l'on puisse y faire quelque chose (arrêt/relance inefficace par exemple). Le débit peut baisser jusqu'à 30 dem / minute.

La seule solution est alors d'attendre la purge des queues de traitement, c'est-à-dire la fin des démarches en amont.

4.3 Performances d'ORCHESTRA et flux « retour »

Si on ne dépasse pas ce seuil critique (et sans injection de flux retour) il est possible de maintenir les performances d'ORCHESTRA autour de 120 dem / min sur une durée assez longue.

Lors de ce test, GS est bridé autour de 130 dem / min (fluctuation possible en fonction de la charge frontal PSL).

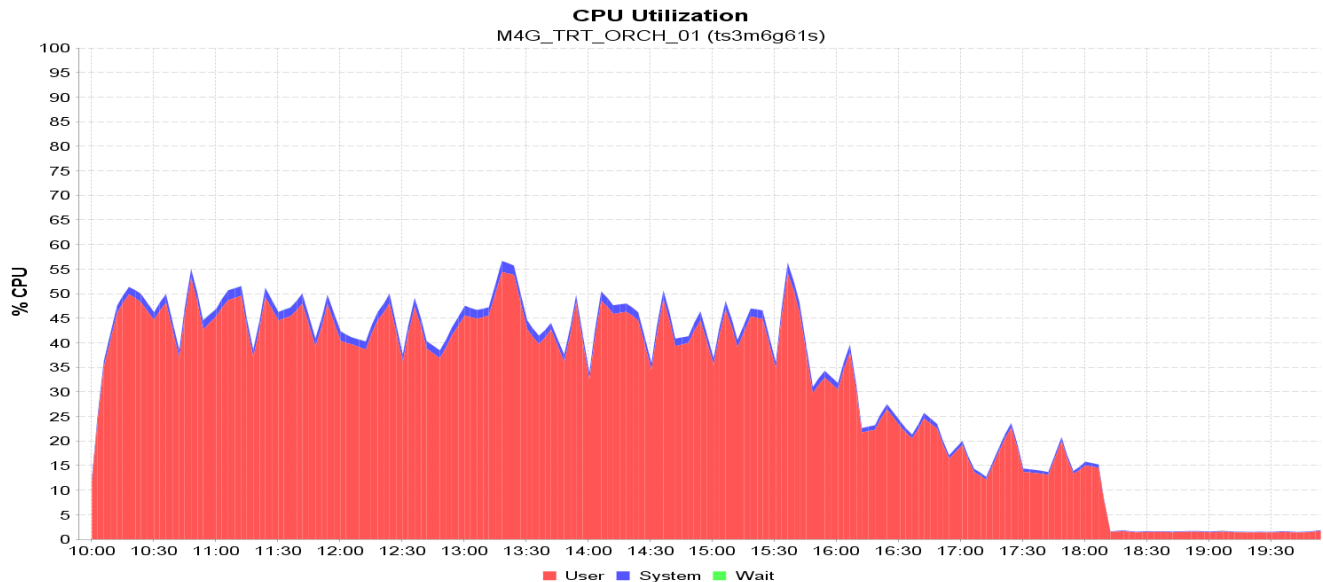


On constate que le débit en sortie d'ORCHESTRA reste stable et aligné sur celui de GS, c'est-à-dire autour de 120. Même lors d'une bosse à 270, **le bridage de GS sécurise totalement le flux de transfert des dossiers à l'INSEE.**

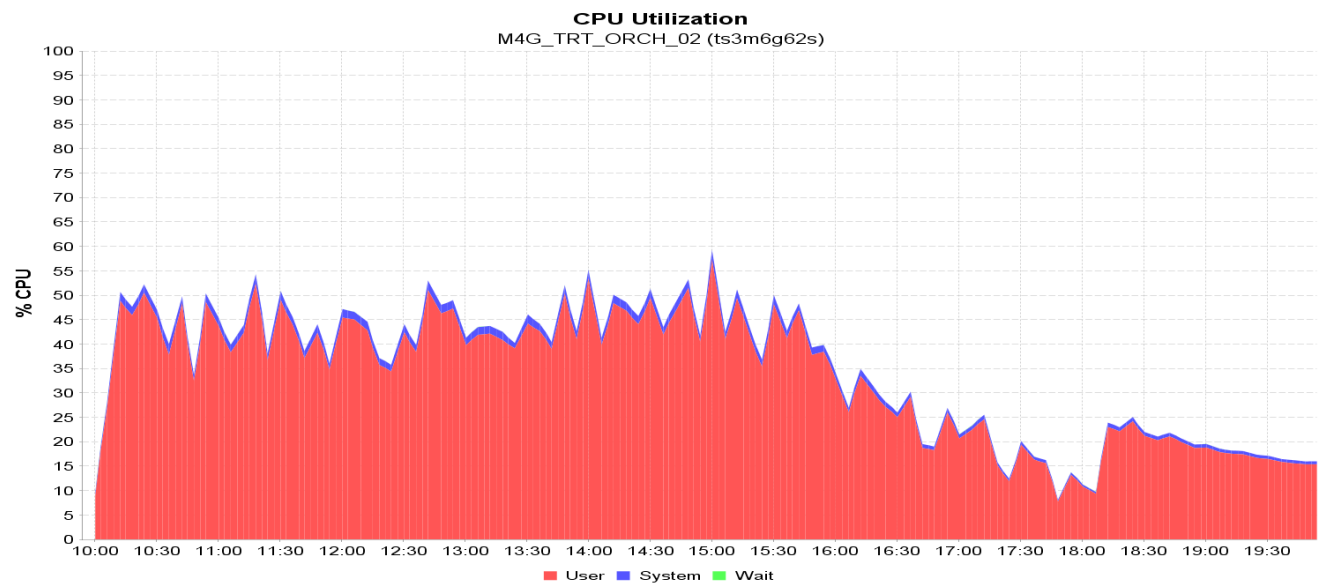
Mais à la fin de la bosse, des flux retour sont injectés faiblement (lot de 500 status 12 et 14) ce qui perturbe totalement ORCHESTRA (qui est plus sollicité et atteint son seuil de rupture probablement). A partir de ce point, les performances ne cessent de chuter et le retard pris est de plusieurs heures.

Si on regarde la charge sur les nœuds de traitement ORCHESTRA, on constate des choses surprenantes

Sur le nœud 1



Sur le nœud 2



L'utilisation CPU d'ORCHESTRA baisse à partir de 15H30 (injection des 1ers flux retours) et le nœud 1 cesse de travailler vers 18H10. Nous avons donc :

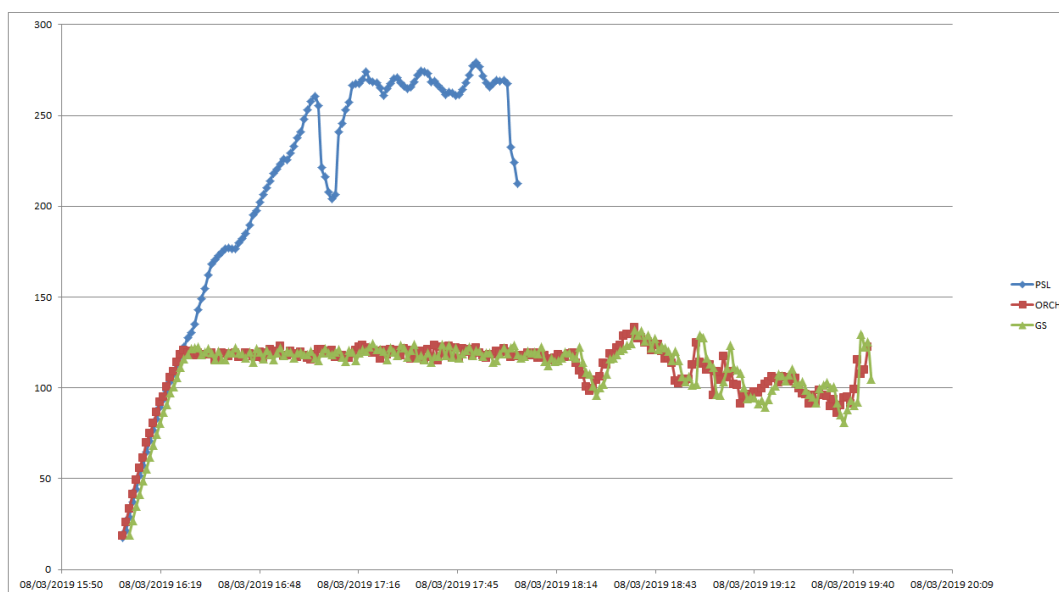
- Une baisse de performance sans explication apparente....probablement un problème d'accès concurrent quelque part (rien dans les logs), dans les traitements ORCHESTRA.
- Une anomalie de répartition de la charge sur les nœuds ORCHESTRA et l'arrêt complet du traitement d'un des nœuds.

Plusieurs paramétrages ont été réalisé par la TMA sur la configuration d'ORCHESTRA mais sans donner de résultats convaincants.

4.4 Patch ORCHESTRA

La TMA a proposé l'installation d'un patch de performance (permettant de gérer une taille de lot plus grande) datant de 2013 qui aurait été oublié lors de la migration de DC. **Un seul test a été réalisé avec ce patch ce qui est insuffisant pour valider son efficacité.** L'objectif était de débrider totalement GS et de voir le comportement d'ORCHESTRA.

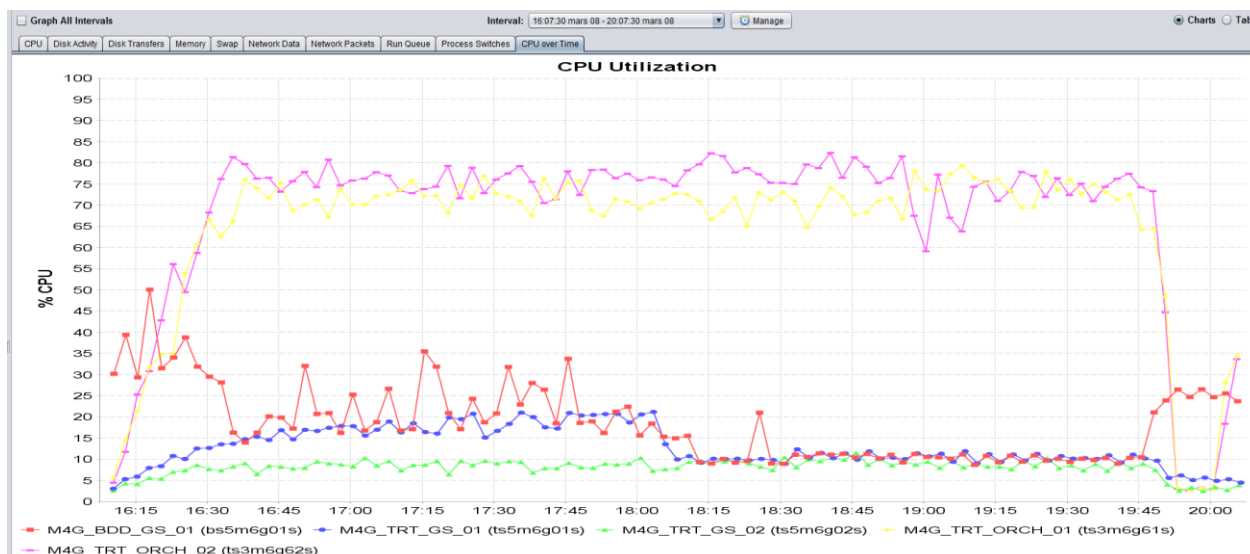
Voici les résultats obtenus :



Malgré un débridage théorique important de GS, les performances de GS ne dépassent pas les 120 dem / minute, rythme sur lequel se cale ORCHESTRA de manière plutôt stable.

Clairement, le patch ORCHESTRA a un effet « négatif » sur les performances de GS et bride l'ensemble à un maximum de 120 dem / minute Quoi que l'on fasse.

Le comportement global d'ORCHESTRA semble malgré tout plus stable, l'utilisation des ressources étant bien équilibrées.



En définitive, l'efficacité du patch reste au final à valider sur des tests complémentaires.

4.5 Bases de données MySQL GS & ORCHESTRA

Certaines requêtes lentes ont été identifiées sur les bases de données GS et ORCHESTRA.

Base GS

```
# User@Host: suiv[ui] @ ts5m6g22s.sys.meshcore.net [10.42.184.112]
# Query_time: 1.554429 Lock_time: 0.000060 Rows_sent: 1 Rows_examined: 4810729
SET timestamp=1550841604;
select count(*) as y0_ from teledossier this_ left outer join TELEDossier_COMPLEMENT this_1_ on this_.tel_id=this_1_.TEL_CPLT_ID where not exists (select 1 from demarche_en_cours
where this_.tel_id=dec_tel_id) and this_.tel_date_initiation_orchestration is null and this_.tel_orchestration_init=0 and this_.tel_orchestration_init is not null and this_.tel_date_soumission<'2019-
02-22 14:19:48';
# Time: 190222 14:20:06

# User@Host: suiv[ui] @ ts5m6g22s.sys.meshcore.net [10.42.184.112]
# Query_time: 1.658667 Lock_time: 0.000087 Rows_sent: 6 Rows_examined: 4810735
SET timestamp=1550841606;
select this_.tel_id as y0_ from teledossier this_ left outer join TELEDossier_COMPLEMENT this_1_ on this_.tel_id=this_1_.TEL_CPLT_ID where not exists (select 1 from demarche_en_cours
where this_.tel_id=dec_tel_id) and this_.tel_date_initiation_orchestration is null and this_.tel_orchestration_init=0 and this_.tel_orchestration_init is not null and this_.tel_date_soumission<'2019-
02-22 14:19:49' order by this_.tel_date_soumission asc limit 20;
```

Base ORCHESTRA

```
# User@Host: orchestra[orchestra] @ ts3m6g62s.sys.meshcore.net [10.42.180.80]
# Query_time: 2.755906 Lock_time: 0.000056 Rows_sent: 1 Rows_examined: 43422
SET timestamp=1550856528;
select jobimpl0_.DBID_ as DBID1_7_, jobimpl0_.DBVERSION_ as DBVERSION3_7_, jobimpl0_.DUEDATE_ as DUEDATE4_7_, jobimpl0_.ISSUSPENDED_ as ISSUSPEN5_7_,
jobimpl0_.ISEXCLUSIVE_ as ISEXCLUS6_7_, jobimpl0_.LOCKOWNER_ as LOCKOWNER7_7_, jobimpl0_.LOCKEXPTIME_ as LOCKEXPT8_7_, jobimpl0_.PRIORITY_ as PRIORITY9_7_,
jobimpl0_.RETRIES_ as RETRIES10_7_, jobimpl0_.DELAY_ as DELAY11_7_, jobimpl0_.EXCEPTION_ as EXCEPTION12_7_, jobimpl0_.PROCESSINSTANCE_ as PROCESS13_7_,
jobimpl0_.EXECUTION_ as EXECUTION14_7_, jobimpl0_.SIGNAL_ as SIGNAL15_7_, jobimpl0_.NODE_ as NODE16_7_, jobimpl0_.EVENT_ as EVENT17_7_, jobimpl0_.REPEAT_ as
REPEAT18_7_, jobimpl0_.INST_UUID_ as INST19_7_, jobimpl0_.INVOKE_DATA_ as INVOKE20_7_, jobimpl0_.SIG_INVOKE_DATA_ as SIG21_7_, jobimpl0_.CLASS_ as CLASS2_7_ from
NOVA_JOB jobimpl0_ where (jobimpl0_.LOCKEXPTIME_ is null or jobimpl0_.LOCKEXPTIME_ <='2019-02-22 18:28:45') and jobimpl0_.DUEDATE_ <='2019-02-22 18:28:45' and
jobimpl0_.RETRIES_>0 and jobimpl0_.ISSUSPENDED_=0 order by jobimpl0_.PRIORITY_ asc, jobimpl0_.DUEDATE_ asc limit 1;
# User@Host: orchestra[orchestra] @ ts3m6g62s.sys.meshcore.net [10.42.180.80]
# Query_time: 3.465710 Lock_time: 0.000035 Rows_sent: 0 Rows_examined: 0
SET timestamp=1550856528;
insert into NOVA_EXECUTION (DBVERSION_, ID_, PRIORITY_, NEXTLOGIDX_, STARTED_DATE_, INSTANCE_UUID_, ACTIVITY_UUID_, CLASS_, DBID_) values (0, 'f2654943-df5f-4eb4-
9fa6-2ad4c75b22b5', 0, 0, '2019-02-22 18:28:44', '43$76199879', '43$76199879$5cc9e64-f22e-40ea-bf40-60298d7908a5', 'org.ow2.orchestra.runtime.BpelExecution', 76852075);
# User@Host: orchestra[orchestra] @ ts3m6g62s.sys.meshcore.net [10.42.180.80]
# Query_time: 3.298685 Lock_time: 0.000040 Rows_sent: 0 Rows_examined: 0
SET timestamp=1550856528;
insert into NOVA_EXECUTION (DBVERSION_, ID_, PRIORITY_, NEXTLOGIDX_, STARTED_DATE_, INSTANCE_UUID_, ACTIVITY_UUID_, CLASS_, DBID_) values (0, '191a86a4-12a0-
4f81-be32-639b7edc1a57', 0, 0, '2019-02-22 18:28:44', '43$76199883', '43$76199883$0922f1a7-7b92-494f-84b7-bc087df1354a', 'org.ow2.orchestra.runtime.BpelExecution', 76852083);
# User@Host: orchestra[orchestra] @ ts3m6g62s.sys.meshcore.net [10.42.180.80]
# Query_time: 3.389463 Lock_time: 0.000032 Rows_sent: 0 Rows_examined: 0
SET timestamp=1550856528;
```

Des optimisations ou des purges doivent être menées pour soulager ces traitements.

Concernant la base ORCHESTRA, une consommation CPU non négligeable est observée. L'ajout de 2 vcores sur l'instance de pré-production a été réalisé et doit être également fait sur la PRODUCTION.

4.6 Purge du NAS GS

Des difficultés à purger correctement le NAS GS de pré-production (et de production) ont été observées durant cette campagne.

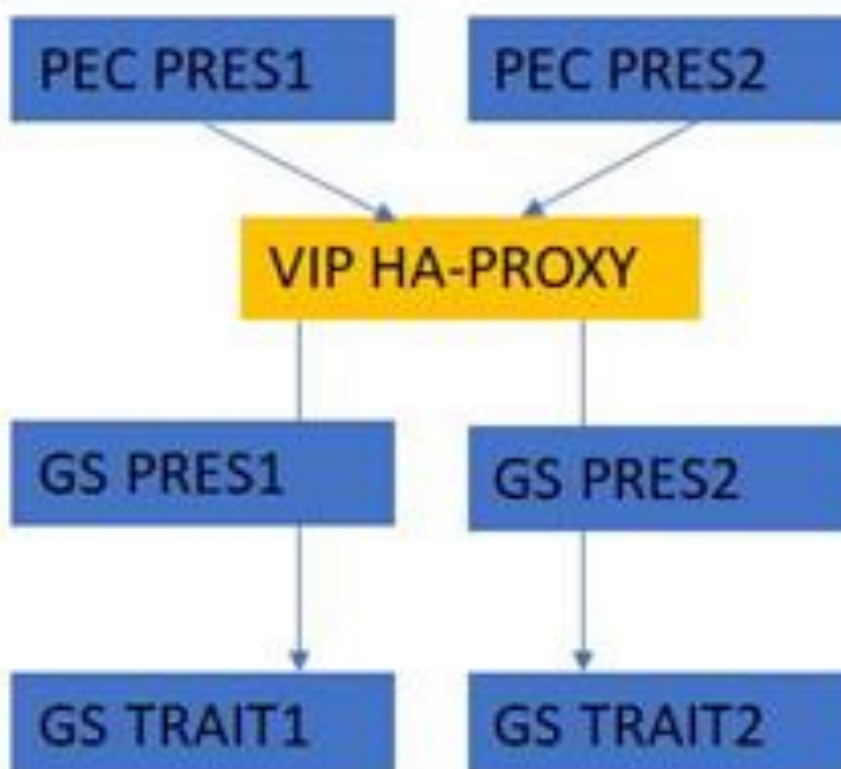
Il faut trouver une solution rapidement sur cette purge du NAS GS.

4.7 Mise en place des VIP HAPROXY

Le début de la campagne de performance a été perturbé par des erreurs sur le lien PEC -> GS.

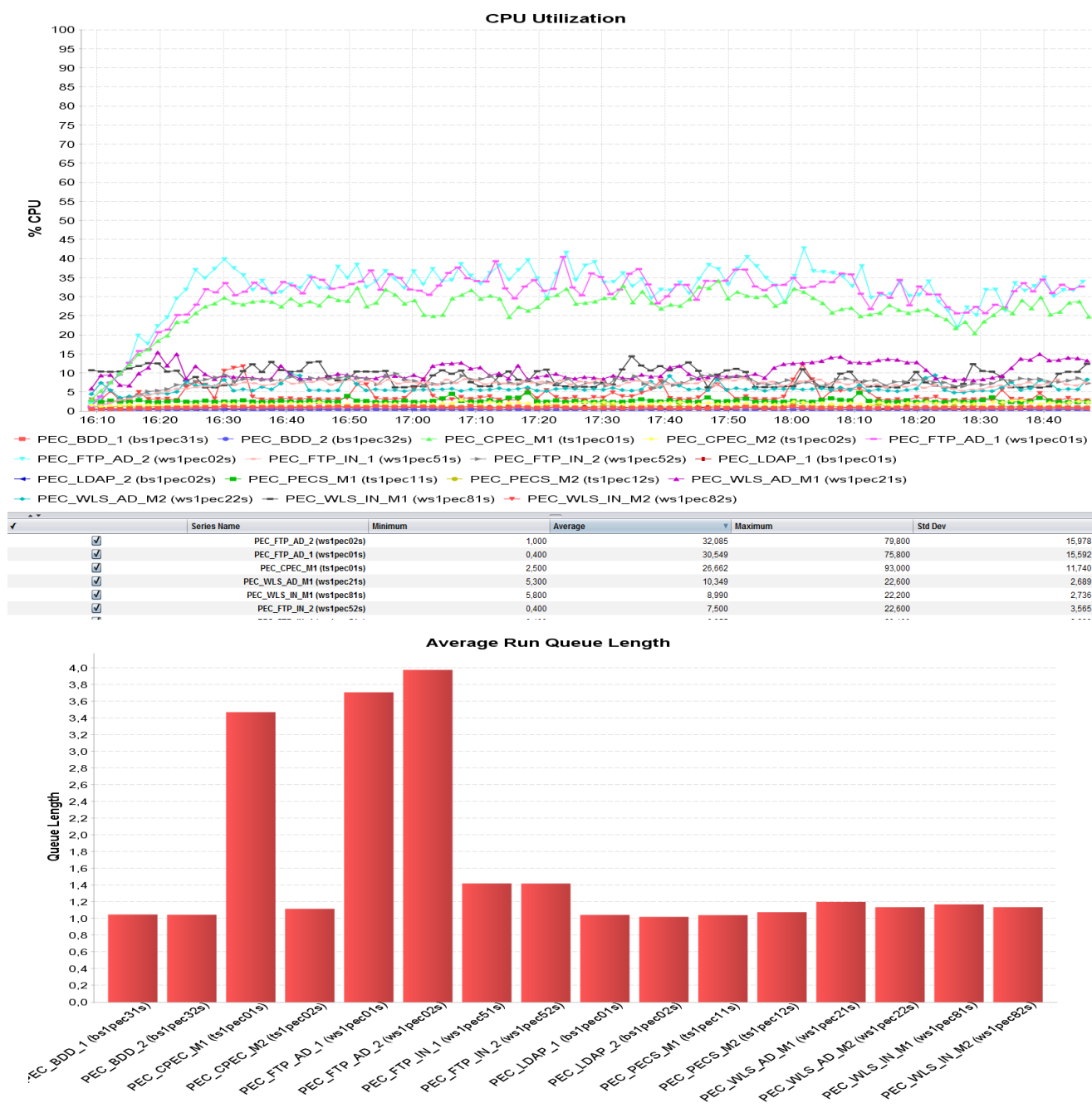
Les soucis furent corrigés rapidement et la solution sur **le HA PROXY est robuste et n'amène aucune inquiétude.**

Configuration HA (pas de check FTP)



5 Comportement PEC

Le comportement de la PEC est satisfaisant et, excepté un souci de répartition de la charge sur les nœuds CPEC_M1 et M2, il n'y a pas d'alerte particulière.



Ce sont les serveurs FTP qui travaillent logiquement le plus puisqu'il réalise le traitement antivirus également. Le reste des machines est peu sollicité.

Aucune erreur rencontrée de type STUCK_THREAD.

Comportement rassurant de la PEC

6 Synthèse

- 1) Les services frontaux PSL et SP fonctionnent bien et sont capables d'encaisser un débit proche de 280 démarches par minute sur une durée de plusieurs heures.
- 2) Quelques comportements erratiques, des pages incomplètes de la PSL arrivent coté client lors de l'injection mais aucune erreur coté serveur pendant quelques secondes parfois....à investiguer coté réseau ?
- 3) Le trafic réseau entre la base de données PSL et les serveurs d'applications PART reste très important même si les temps de réponse sont bons (< 1 ms). A terme, ce sujet pourrait redevenir problématique et limiter la montée en charge.
- 4) L'anomalie des PJ PDF a été corrigée (espace disque non libéré) sur les serveurs PART.
- 5) La page de délestage a été testée, Une valeur de 6000 sessions sur 900 secondes est proposée.
- 6) **ORCHESTRA** reste le maillon faible de la chaîne. En effet :
 - a. En terme de stabilité, un seuil de rupture grave (non réparable par un arrêt/relance) de 130 dem / min a été identifié. Les performances ne cessant de se dégrader par la suite. Il ne faut donc jamais dépasser ce seuil.
 - b. La répartition de la charge est erratique, parfois l'un des nœuds de traitement ne fonctionne plus sans aucune raison.
 - c. Les performances ne dépassent pas les 120 dem / min, patch ou pas.
 - d. Le patch semble apporter quelques améliorations en termes de stabilité (meilleures conso CPU, meilleure répartition de la charge)....mais
 - e.bride de facto les performances de GS....sans explication....on ne dépasse plus 120 sur GS !
- 7) La base **ORCHESTRA** nécessite l'ajout de 2 vcpus pour bénéficier de plus de puissance.
- 8) Les flux retours sont très problématiques dès que le débit atteint 40 dem / min et perturbent fortement les flux aller. Il faut adopter une stratégie pour ces flux retours et ne pas les traiter lors du pic de charge.
- 9) Les bases de données GS et **ORCHESTRA** doivent faire l'objet d'une purge et d'optimisation SQL sur les requêtes lentes identifiées.
- 10) Le NAS GS doit être purgé correctement et de manière fiable pour éviter tout risque de saturation.
- 11) La mise en place des VIP HAPROXY est robuste et satisfaisante.
- 12) Le comportement de la PEC est satisfaisant.

Préconisation : Il ne faut jamais dépasser les 120 dem / min sur le back office GS/ORCHESTRA car le risque de perdre ORCHESTRA (performances alors limitées à 30 dem / min) est important. Et dans ce cas, les flux seront transmis très lentement à l'INSEE avec un retard énorme. De plus, les flux retour ne doivent pas être traités en parallèle avec les flux aller lorsque la charge commence à monter (40 dem / min).

Il pourrait être judicieux de suivre le fonctionnement global des différentes briques (PSL, GS et ORCHESTRA, statuts 10,26 et 11) en exécutant régulièrement les requêtes suivantes dans la base GS de production pendant la période de pic.

```
Select STR_TO_DATE(eta_date, '%Y-%m-%d %H:%i'),';10;',count(*),';' from teledossier inner
join enveloppe on tel_id=env_tel_id inner join etat on ETA_ENV_ID=env_id where
eta_etat_id = 10 and env_si_num_guichet='REU' and eta_date > STR_TO_DATE('${DATE_DEBUT}',
'%Y-%m-%d %H:%i:%s') and eta_date <= STR_TO_DATE('${DATE_FIN}', '%Y-%m-%d %H:%i:%s')
group by STR_TO_DATE(eta_date, '%Y-%m-%d %H:%i')
```

```
Select STR_TO_DATE(eta_date, '%Y-%m-%d %H:%i'),';26;',count(*),';' from teledossier inner
join enveloppe on tel_id=env_tel_id inner join etat on ETA_ENV_ID=env_id where
eta_etat_id = 26 and env_si_num_guichet='REU' and eta_date > STR_TO_DATE('${DATE_DEBUT}',
'%Y-%m-%d %H:%i:%s') and eta_date <= STR_TO_DATE('${DATE_FIN}', '%Y-%m-%d %H:%i:%s')
group by STR_TO_DATE(eta_date, '%Y-%m-%d %H:%i')
```

```
Select STR_TO_DATE(eta_date, '%Y-%m-%d %H:%i'),';11;',count(*),';' from teledossier inner
join enveloppe on tel_id=env_tel_id inner join etat on ETA_ENV_ID=env_id where
eta_etat_id =11 and env_si_num_guichet='REU' and eta_date > STR_TO_DATE('${DATE_DEBUT}',
'%Y-%m-%d %H:%i:%s') and eta_date <= STR_TO_DATE('${DATE_FIN}', '%Y-%m-%d %H:%i:%s')
group by STR_TO_DATE(eta_date, '%Y-%m-%d %H:%i')
```