



HAL
open science

Monitoring étendu d'infrastructures avec Kwolect

Simon Delamare, Lucas Nussbaum

► **To cite this version:**

Simon Delamare, Lucas Nussbaum. Monitoring étendu d'infrastructures avec Kwolect. JRES (Journées réseaux de l'enseignement et de la recherche) 2021, Renater, May 2022, Marseille, France. hal-04807316

HAL Id: hal-04807316

<https://hal.science/hal-04807316v1>

Submitted on 27 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Monitoring étendu d'infrastructures avec Kwollect

Simon Delamare

Laboratoire de l'Informatique du Parallélisme (CNRS)

46 allée d'Italie

69 007 Lyon

Lucas Nussbaum

Inria Nancy – Grand Est

615 rue du Jardin Botanique

54 600 Villers-lès-Nancy

Résumé

Kwollect est un logiciel de monitoring pour les utilisateurs des infrastructures informatiques spécialisé dans la collecte de métriques environnementales.

C'est la solution de monitoring d'expérience proposée aux utilisateurs de Grid'5000/SILECS, l'infrastructure de recherche en informatique distribuée : pour alimenter les expérimentations des chercheurs, Kwollect y réalise la collecte et la mise à disposition de métriques issues de plus de 2500 équipements différents, dont des dispositifs de mesure de consommation électrique réalisant 50 mesures par seconde, des sondes de températures, etc.

Kwollect possède plusieurs spécificités par rapport aux solutions de monitoring usuelles :

- il est en mesure d'effectuer la collecte de métriques à haute fréquence et de les conserver, sans perte, durant de longues durées ;*
- il est spécialisé dans la collecte de métriques « externes » aux serveurs : capteurs environnementaux, équipements réseaux, contrôleurs BMC, PDUs... mais supporte également les métriques Prometheus ;*
- il est prévu pour être utilisé par les utilisateurs des plateformes et pas uniquement par ses administrateurs (possibilité d'insérer ses propres métriques, métriques activables à la demande, API pour récupérer les données, intégration avec les « jobs schedulers » des centres de calcul...).*

Kwollect est conçu comme un cadriciel et réutilise au maximum des briques logicielles existantes : il utilise notamment une base de donnée PostgreSQL et son extension TimescaleDB pour stocker les métriques, fournir une API et s'occuper de la logique applicative.

La visualisation s'appuie sur Grafana.

Dans cet article, nous détaillerons le fonctionnement de Kwollect, expliquerons les choix technologiques effectués pour sa conception et présenterons son déploiement sur Grid'5000.

Le site web de Kwollect est : <https://gitlab.inria.fr/grid5000/kwollect>.

Mots-clefs

Monitoring, Infrastructure, Mesures environnementales, Consommation d'énergie, PostgreSQL, TimescaleDB

1 Introduction

La compréhension du comportement d'un système informatique complexe nécessite de collecter des métriques sur le matériel, les services ou l'application étudiée, ainsi que sur son environnement. Différents types de mesures peuvent être considérés, comme :

- les mesures faisant directement l'objet de l'étude, comme la consommation d'énergie du système testé, ou le taux d'utilisation d'un CPU ;
- les mesures qui aident à comprendre le comportement, comme les données de trafic réseau vues depuis les équipements de réseau ;
- les mesures sur l'environnement pouvant influencer le comportement résultant, certains facteurs environnementaux, tels que la température ou l'humidité, pouvant par exemple influencer les performances ; une machine excessivement chaude pouvant limiter la capacité de son CPU.

Les plateformes permettant à leurs utilisateurs de réaliser des expériences, que ce soit en lien avec le HPC, le réseau, l'IoT ou les clouds, se doivent de fournir un service facilitant l'obtention de ces mesures pour leurs utilisateurs. Cependant, comme nous le détaillerons plus tard, les services de monitoring existants sont souvent limités en termes de fonctionnalités et de capacité à s'adapter à un grand nombre de métriques collectées, parfois à haute fréquence.

Cet article décrit Kwollect, un service de monitoring d'infrastructures informatiques (y compris celles issues de *wattmètres* à haute fréquence, d'où son nom) ayant pour objectif de faciliter la collecte et la mise à disposition de métriques lors de la réalisation d'une expérience. Il a été développé dans le contexte de la plateforme Grid'5000/SILECS [1], mais il est conçu pour être utilisé sur n'importe quelle infrastructure. Kwollect peut collecter des métriques à haute fréquence (inférieure à la seconde), puis les exposer à l'expérimentateur via une API REST ou une interface de visualisation interactive.

2 Contexte

2.1 Grid'5000, plate-forme pour la réalisation d'expériences

Grid'5000/SILECS [1] est une plateforme pour la réalisation d'expériences dans les domaines de la recherche en systèmes distribués, incluant le calcul haute performance (HPC), le *cloud* et le « *edge computing* », le réseau, et l'IA. Grid'5000 est réparti en différents sites, chacun regroupant jusqu'à plusieurs centaines de nœuds (serveurs) pouvant être réservés et reconfigurés par les utilisateurs de la plateforme pour réaliser une expérience.

Dans les domaines de recherche couverts par Grid'5000, la capacité d'expérimenter à grande échelle est cruciale. Ainsi, il n'est pas rare de voir des expériences se dérouler sur des centaines de nœuds, ce qui sollicite d'autant le service de monitoring. De plus, dans ces domaines de recherche (et surtout dans le HPC), les expérimentateurs sont souvent intéressés par des comportements subtils qui ne sont visibles que lorsque les mesures sont collectées à une fréquence élevée (1 Hz ou plus).

Une des motivations centrale justifiant la mise à disposition d'un service de monitoring aux utilisateurs de Grid'5000 est la communauté de recherche liée à l'efficacité énergétique des

systèmes informatiques. Leurs expériences reposent en grande partie sur du matériel spécifique (dispositifs de mesure de consommation énergétique appelés wattmètres) permettant des mesures précises de la puissance électrique des serveurs, afin d'étudier par exemple l'influence du comportement des applications sur la consommation d'énergie.

2.2 Travaux connexes

Les solutions de supervision d'infrastructure, telles que Munin, Nagios, Ganglia[2] ou, plus récemment, Prometheus[3], sont des services généralement utilisés par les opérateurs des infrastructures pour monitorer leur bon fonctionnement. Ces solutions sont éprouvées et remplissent très bien leurs fonctions, mais sont généralement conçues pour fournir des mesures peu précises et à basse fréquence (une mesure toutes les quelques secondes, ou plus). En outre, ces solutions peuvent ne pas être prévues pour mettre les mesures à dispositions des utilisateurs, mais uniquement des opérateurs de l'infrastructure. Enfin, ces solutions sont principalement conçues pour réaliser des mesures sur les systèmes d'exploitation des nœuds. Leur capacité à monitorer les dispositifs externes, tels que les unités d'alimentation (PDU) ou les cartes d'administration des serveurs (Baseboard Management Controller -BMC-, comme l'iDRAC de Dell), est souvent fournie comme une extension moins bien supportée, alors que c'est un objectif principal de Kwolect.

Plusieurs plateformes pour l'expérimentation proposent à leurs utilisateurs un système de monitoring. Sur la plateforme Chameleon Cloud, le service OpenStack Gnocchi[5] est utilisé dans ce but, en s'appuyant sur collectd comme agent fonctionnant sur chaque nœud (avec des données récupérées toutes les 10 s), et en le complétant avec des données de puissance récupérées via IPMI (Intelligent Platform Management Interface). Sur Grid'5000, Kwapi[4] était utilisé avant que Kwolect ne le remplace.

3 Conception de Kwolect

Cette section décrit le fonctionnement de Kwolect et explique les choix qui ont été faits lors de sa conception. La figure 1 donne un aperçu des composants de Kwolect qui seront présentés ici.

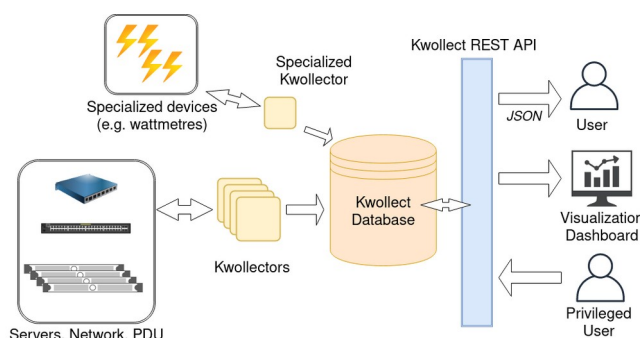


Figure 1 : Architecture de Kwolect avec ses différents composants

Au cœur de Kwolect se trouve le gestionnaire de base de données PostgreSQL et son extension TimescaleDB, qui permet de stocker efficacement les métriques. L'API REST est directement fournie par une autre extension PostgreSQL. Les métriques sont collectées par un ensemble de

Kwollectors, qui sont des programmes autonomes pouvant fonctionner en parallèle. Enfin, une interface de visualisation interactive est également fournie. Kwollect est un logiciel libre. Son code source et les instructions d'installation sont disponibles en ligne¹.

3.1 Stockage des Métriques

Kwollect utilise une base de données PostgreSQL avec son extension TimescaleDB pour le stockage des métriques. L'utilisation d'une base de données relationnelle bien établie telle que PostgreSQL offre de nombreux avantages : cela permet d'associer les métriques à d'autres données disponibles dans PostgreSQL (y compris externes via les *Foreign Data Wrapper* de PostgreSQL) en utilisant des jointures SQL classiques. Cela permet également de profiter de la maturité et de la richesse de l'écosystème de PostgreSQL.

Cependant, par défaut, PostgreSQL n'offre pas de performances élevées pour la gestion de grands ensembles de données temporelles. Une approche naïve consistant à insérer les données dans une seule table ne serait pas possible, car la table ne cesserait de croître, ce qui finirait par poser des problèmes de performance : les index de la table ne pourraient pas être contenus dans la mémoire du système, ce qui entraînerait des opérations d'entrée/sortie intensives à chaque accès aux données.

TimescaleDB résout ce problème en effectuant un partitionnement automatique des tables en fonction de l'horodatage des données insérées. Le découpage des tables est réalisé de telle sorte à ce que la partition la plus récente tienne toujours dans la mémoire du système. Ainsi, en présupposant que dans la plupart des cas, les données insérées ou consultées se rapportent à une période temporelle récente, les performances obtenues restent bonnes et constantes dans le temps, même si de grandes quantités de données sont insérées.

TimescaleDB propose également d'autres fonctionnalités utiles pour la gestion des données temporelles, comme la compression, utilisée par Kwollect pour réduire l'utilisation du disque d'un facteur de 100, l'agrégation continue des données et une politique de rétention personnalisable.

3.2 Format des métriques

Kwollect stocke toutes les métriques qu'il collecte dans une unique table *metrics*, qui utilise le schéma décrit dans le tableau 1.

| Column | Type |
|-----------|--------------------------|
| timestamp | timestamp with time zone |
| device_id | text |
| metric_id | text |
| value | double precision |
| labels | jsonb |

Tableau 1 : Format des métriques Kwollect utilisé dans la base de données.

Ce format est largement inspiré de Prometheus et de son standard dérivé *OpenMetrics*.

Chaque donnée de mesure est associée à ces informations :

- *timestamp* : la date de la mesure, avec une précision en microseconde ;
- *device_id* : l'identifiant du dispositif qui est mesuré ;

¹ <https://gitlab.inria.fr/grid5000/kwollect>

- *metric_id* : l'identifiant de la métrique mesurée. Le nom doit également suivre les directives de Prometheus², par exemple, le suffixe de l'identifiant de la métrique doit indiquer l'unité de la mesure réalisée ;
- *value* : la valeur de la mesure, sous la forme d'un nombre à virgule flottante ;
- *labels* : des valeurs supplémentaires arbitraires, au format JSON (par exemple, un nom alternatif pour le dispositif, des propriétés supplémentaires qui ne peuvent pas être représentées par le champ de valeur, etc.).

3.3 Collecte des métriques

Les *Kwcollectors* sont chargés de lire les métriques sur les différents périphériques et de stocker les valeurs résultantes dans la base de données. Les *Kwcollectors* sont des services autonomes : ils n'ont besoin que d'accéder à la base de données et aux périphériques qu'ils monitorent. Il peut donc y avoir plusieurs *Kwcollectors* en cours d'exécution pour monitorer différents dispositifs et leur déploiement peut être réparti sur des machines supplémentaires. Cette conception permet une prise en charge aisée d'un nombre très élevé de dispositifs monitorés pour les opérations de lecture et de stockage des métriques.

L'implémentation actuelle du *Kwcollector* fourni dans *Kwollect*, supporte la lecture de métriques via les protocoles SNMP (Simple Network Management Protocol), IPMI et Prometheus. Il est écrit en Python et utilise la bibliothèque *asyncio* pour exploiter la forte concurrence des opérations d'entrées/sorties réalisées : un seul cœur d'un serveur standard suffit pour faire fonctionner une instance d'un *Kwcollector* réalisant le monitoring de centaines de dispositifs comprenant des centaines de métriques chacun. Les *Kwcollectors* acceptent des fichiers de configuration décrivant les dispositifs et les métriques à récupérer. Un exemple est fourni dans l'extrait 1.

```
- name: bmc_power_watt
  device_id: node-1
  url: snmp://public@node-1-admin.domain.com/1.3.6.1.4.1.674.10892.5.4.600.30.1.6.1.3
  update_every: 5000

- name: prom_node_procs_running
  device_id: node-2
  url: prometheus://node-2.domain.com:9100/node_procs_running
  update_every: 15000
```

Extrait 1 : Exemple de configuration du Kwcollector

Certaines implémentations alternatives de *Kwcollector* ont été créées, afin de réaliser le monitoring de périphériques spécifiques, tels que les wattmètres à haute fréquence disponibles dans Grid'5000.

3.4 API Utilisateur

Kwollect fournit une API REST permettant aux utilisateurs de récupérer les valeurs des métriques. L'API accepte des paramètres tels que la liste des dispositifs à partir desquels récupérer les valeurs, le début et la fin de la période de temps, ou le nom des métriques. Elle renvoie les résultats au format JSON. Un exemple de résultats est fourni dans l'extrait 2.

```
[{"timestamp": "2020-11-26T11:53:50.164813+01:00",
  "device_id": "node-1", "metric_id": "bmc_power_watt", "value": 138, "labels": {}},
```

2 <https://prometheus.io/docs/practices/naming/>

```

{"timestamp": "2020-11-26T11:53:50.164813+01:00",
 "device_id": "node-2", "metric_id": "prom_node_procs_running", "value": 393, "labels": {}},
{"timestamp": "2020-11-26T11:53:55.175123+01:00",
 "device_id": "node-1", "metric_id": "bmc_power_watt", "value": 138, "labels": {}},
{"timestamp": "2020-11-26T11:54:05.150139+01:00",
 "device_id": "node-2", "metric_id": "prom_node_procs_running", "value": 371, "labels": {}}

```

Extrait 2 : Exemple de résultats retournés par une requête à l'API de Kwollect

L'API est implémentée par PostgREST, qui fournit une API REST au-dessus d'une base de données PostgreSQL. La disponibilité de PostgREST met en évidence les avantages à utiliser une solution de stockage répandue telle que PostgreSQL : cela permet à Kwollect de disposer d'une API sans avoir à la développer en interne.

L'enregistrement de valeurs dans la base de données est également possible pour les utilisateurs authentifiés (le contrôle d'accès est basé sur les privilèges des utilisateurs définis dans la base de données).

3.5 Interface graphique

L'utilisation de PostgreSQL permet d'utiliser directement les solutions de visualisation existantes, telles que Grafana. Kwollect fournit un tableau de bord Grafana préconfiguré pour afficher les valeurs des métriques. Un exemple de ce tableau de bord est donné à la figure 2.

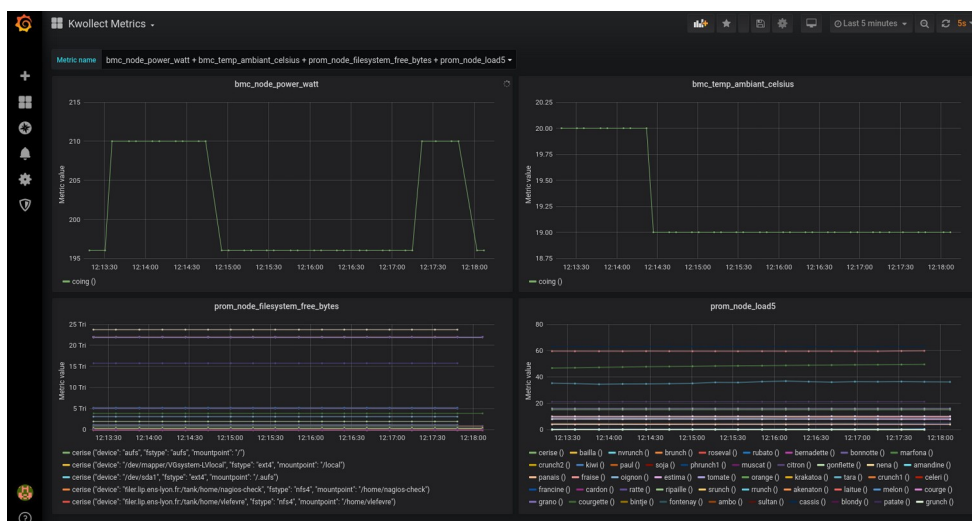


Figure 2 : Capture d'écran illustrative du tableau de bord de visualisation de Kwollect basé sur Grafana.

3.6 Fonctionnalités spécifiques

La polyvalence de PostgreSQL permet d'implémenter les fonctionnalités spécifiques à une solution de monitoring orientée pour les usages expérimentaux directement dans la base de données, sans nécessiter aucun autre développement supplémentaire, comme illustré par les exemples suivants.

3.6.1 Intégration au *job scheduler*

Kwollect fournit à ses utilisateurs un moyen facile de récupérer les métriques d'une expérience. Dans Grid'5000, les expériences sont orchestrées par OAR, un système de *job scheduler* (une expérience étant représentée par un job).

L'intégration des informations sur les métriques avec le *job scheduler* est effectuée par PostgreSQL en utilisant une opération JOIN entre la table des métriques et celle représentant les jobs OAR. Les mesures peuvent ainsi être sélectionnées par un numéro de job via une VIEW PostgreSQL, qui est à son tour automatiquement rendue disponible pour les utilisateurs via l'API REST.

3.6.2 Métriques à la demande

Pour éviter de consommer inutilement de l'espace disque et des ressources, Kwolect permet que certaines métriques ne soient pas collectées par défaut mais uniquement à la demande. Le monitoring de ces métriques doit être activé en remplissant une table spécifique de la base de données, qui liste les périphériques et les métriques à la demande qui sont à activer. La table est exposée aux Kwolectors pour les informer des métriques à la demande à collecter.

Dans Grid'5000, toutes les métriques possibles ne sont pas activées par défaut, mais les utilisateurs de Kwolect ont la possibilité de les activer en spécifiant certaines options lors de la soumission de leur job. En interne, puisque les informations sur les jobs d'OAR sont accessibles par PostgreSQL utilisé par Kwolect, ce dernier peut renseigner la table des métriques à la demande à activer avec les métriques demandées par les utilisateurs pour leurs jobs.

4 Utilisation dans Grid'5000 et scénario d'exemple

Dans cette section, nous présentons l'utilisation de Kwolect dans Grid'5000 et donnons un exemple de scénario illustrant son utilité pour les utilisateurs.

Grid'5000 est réparti sur 9 sites. Sur chacun d'entre eux, une instance de Kwolect indépendante est déployée sur une machine virtuelle hébergeant la base de données et un unique Kwolector réalisant le monitoring de l'ensemble des nœuds et des périphériques du site.

Les métriques collectées par défaut sont les suivantes :

- la consommation d'énergie mesurée à la prise, toutes les secondes, pour les nœuds connectés à un PDU pilotable via le protocole SNMP ou pour ceux disposant d'un wattmètre dédié ;
- les mesures les plus importantes provenant des capteurs présents dans les BMC des nœuds, toutes les 5 secondes, à l'aide des protocoles SNMP ou IPMI (selon ce que le dispositif prend en charge) ;
- le trafic des équipements réseau, toutes les une à 10 secondes (selon ce que le dispositif prend en charge), en utilisant le protocole SNMP ;
- les métriques les plus importantes provenant des exporteurs Prometheus (y compris certaines métriques GPU), toutes les 15 secondes ;
- des métriques arbitraires exposées par les utilisateurs depuis les nœuds via un fichier dédié analysé périodiquement par Kwolectors.

De plus, beaucoup d'autres métriques sont disponibles à la demande, lorsque les utilisateurs les activent explicitement. Par exemple, pour collecter 50 mesures chaque seconde sur les wattmètres ou pour activer les métriques BMC ou Prometheus qui ne le sont pas par défaut.

Au total, 88 métriques distinctes sur 2500 dispositifs différents sont disponibles pour le monitoring via Kwolect dans Grid'5000 et entre 5000 et 10 000 valeurs sont collectées chaque seconde.

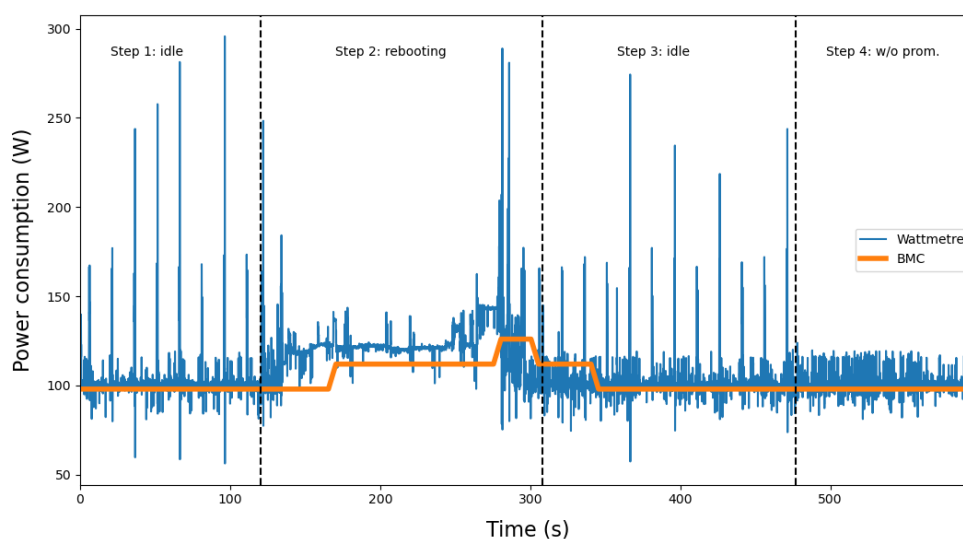


Figure 3 : Evolution de la puissance électrique consommée par un nœud au cours du temps lors du scénario d'exemple, mesurée par sa BMC et par un dispositif de wattmètre.

Nous considérons maintenant un scénario illustrant l'utilisation de Kwolect consistant en une recherche des raisons de la consommation énergétique d'un serveur. Avec Kwolect, un utilisateur est capable de visualiser l'énergie consommée par un nœud de Grid'5000 au cours du temps, en utilisant les mesures réalisées par les wattmètres et le BMC des nœuds (Figure 3).

Après une période d'inactivité, le nœud est redémarré. La consommation d'énergie augmente lorsque le nœud est en charge et que les mécanismes d'économie d'énergie du système d'exploitation sont désactivés. Une fois le démarrage terminé à l'étape 3, le nœud retourne à un état d'inactivité.

Les mesures effectuées par les deux appareils semblent valides : elles suivent la même évolution globale, bien que les valeurs de la BMC soient décalées de quelques secondes dans le temps et mesurent quelques Watts de moins.

Une observation remarquable est celle des pics de consommation rapportés par les wattmètres qui sont absents des valeurs rapportées par la BMC. On peut se demander s'ils correspondent à du bruit ou à une consommation réelle, invisible dans les mesures effectuées par la BMC moins précises. Il s'avère que l'essentiel de ces pics sont causés par l'exporteur Prometheus installé sur le nœud, qui est interrogé toutes les 15 secondes. Après l'avoir arrêté (étape 4), les pics disparaissent.

Cet exemple montre comment Kwolect peut être utilisé pour analyser la consommation électrique, notamment en comparant des données provenant de plusieurs sources et de différentes résolutions. En réalisant la moyenne des valeurs fournies par Kwolect pendant les étapes d'inactivité avec ou sans Prometheus, nous pouvons estimer le coût énergétique du fonctionnement de ce service à 0,35 W.

5 Évaluation des performances

Nous présentons maintenant quelques résultats de performance issus du déploiement de Kwollect dans Grid'5000, à partir de deux de ses sites, Nancy, qui est le plus important (par défaut et pour les résultats présentés dans cette section, 1350 dispositifs distincts sont monitorés, pour une moyenne de 4000 mesures par seconde) et Lyon, qui est de taille moyenne, mais possède de nombreux wattmètres réalisant des mesures à haute fréquence (par défaut, 250 périphériques sont monitorés pour un total de 850 mesures par seconde, et en plus pour les résultats présentés ici, 20 wattmètres à haute fréquence réalisant 50 mesures par seconde chacun, ont été activés à la demande). Les machines virtuelles hébergeant Kwollect disposent de 4 CPUs virtuels, 16 Go de mémoire, 500GB d'espace de stockage (disques mécaniques à Lyon, SSD à Nancy). La charge CPU de la machine virtuelle Kwollect de Nancy, dans des conditions normales de service et avec ou sans requêtes utilisateurs, est présentée dans la figure 4.

Les résultats montrent qu'en l'absence de demandes, moins d'une unité de CPU est nécessaire pour faire fonctionner Kwollect, la charge étant due au Kwollector pour obtenir les données des dispositifs et les insérer dans la base : Kwollect ne nécessite ainsi pas plus d'un seul CPU pour monitorer des centaines de dispositifs. Nous constatons également qu'approximativement, une requête nécessite un CPU pour être traitée, indiquant que des CPUs supplémentaires permettraient de traiter davantage de demandes en parallèle (même si les performances de stockage deviendraient un autre goulot d'étranglement).

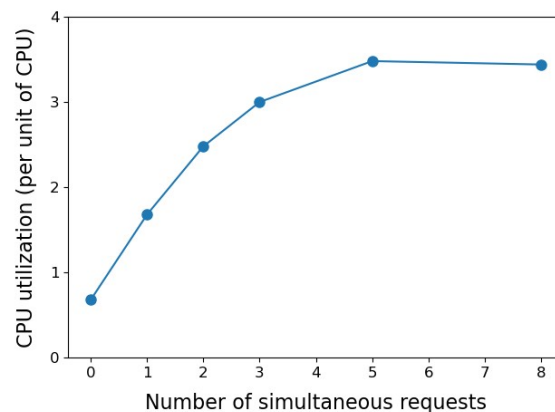


Figure 4 : Taux d'occupation CPU de la machine virtuelle (CPU indiqué actif par l'outil vmstat), en fonction du nombre de requêtes utilisateurs simultanées à traiter (taux moyen sur 60 secondes pour chaque résultat).

Le temps de réponse des requêtes est présenté dans la figure 5, où les utilisateurs demandent à Kwollect de leur retourner les mesures associées à un job ayant duré 5 minutes, à un job ayant duré deux heures ou à un job ayant duré 2 heures, mais 24 heures après sa fin (la requête est réalisée dès que le job se termine dans les autres cas). Différentes configurations de jobs sont considérées : ils peuvent comprendre 2 ou 20 nœuds (donc inclure 10 fois plus de métriques) et être situés à Nancy ou Lyon (donc inclure 50 mesures de wattmètres par nœuds et par seconde de job). Le nombre approximatif de métriques qui correspond à chaque requête est également indiqué sur la figure.

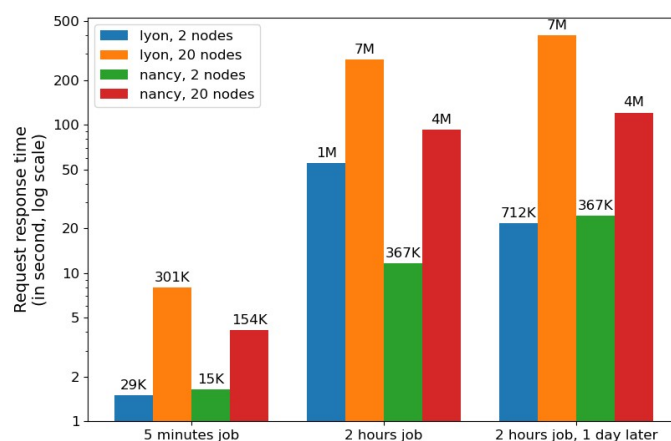


Figure : Temps de réponse de requêtes utilisateurs pour l'obtention des valeurs des métriques associées à un job. Le nombre approximatif de mesures retourné par chaque requête est indiqué en haut des barres (résultats moyens de 3 exécutions différentes, sauf pour la configuration « Lyon, 20 nœuds - travail de 2 heures », où une seule exécution a été réalisée).

Les résultats montrent que, même sur le matériel modeste utilisé dans le déploiement de Grid'5000, Kwollect peut retourner des dizaines de milliers de valeurs en quelques secondes et des millions de valeurs en quelques minutes. Ces durées sont largement compatibles avec les besoins des utilisateurs de Grid'5000, qui n'ont généralement besoin que d'une seule requête par expérimentation.

6 Conclusion

Dans cet article, nous avons présenté la conception, l'évaluation des performances et un scénario d'utilisation de Kwollect, un service de monitoring pour la collecte et la mise à disposition de métriques d'infrastructure informatiques, dans un contexte de réalisation d'expérience. Kwollect a été déployé avec succès sur Grid'5000. Nous allons continuer à étendre Kwollect pour monitorer davantage de métriques et améliorer son intégration avec les outils et les bibliothèques utilisés dans Grid'5000 pour la réalisation d'expériences. Nous espérons également pouvoir déployer Kwollect dans d'autres infrastructures, telles que des plateformes orientées réseau sans fil ou IoT.

Bibliographie

- [1] D. Balouek et al., "Adding virtualization capabilities to the Grid'5000 testbed," in Cloud Computing and Services Science, 2013.
- [2] M. L. Massie et al., "The ganglia distributed monitoring system: Design, implementation, and experience," Parallel Computing, 2004.
- [3] Prometheus monitoring system, <https://prometheus.io/>
- [4] F. Clouet et al., "A unified monitoring framework for energy consumption and network traffic," in TRIDENTCOM, 2015.
- [5] Gnocchi, <https://gnocchi.xyz/>