

Allocation et placement dynamiques sur ressources hétérogènes pour le cloud serverless

20 novembre 2024

Thèse soutenue par **Vincent Lannurien**^{1, 2}

Sous la direction de **Jalil Boukhobza**^{1, 2}

Co-dirigée par **Laurent D'Orazio**^{1, 3}

Encadrée par **Olivier Barais**^{1, 3}

Stéphane Paquelet¹



¹b<>com Institute of Research and Technology

²ENSTA Bretagne, Lab-STICC, CNRS, UMR 6285

³Univ. Rennes, Inria, CNRS, IRISA



Computing as a Public Utility

In concluding I should like to say a word on management and the computer of the future. At present, computers are bought by individual companies or other institutions and are used only by the owning institution. If computers of the kind I have advocated become the computers of the future, then computation may someday be organized as a public utility, just as the telephone system is a public utility. We can envisage computing service companies whose subscribers are connected to them by telephone lines. Each subscriber needs to pay only for the capacity that he actually uses, but he has access to all programming languages characteristic of a very large system.

1. [27] John McCARTHY. « Management and the Computer of the Future ». 1962

Plan de la présentation

1. Contexte

2. État de l'art

3. HeROfake – Orchestrer des fonctions sur ressources hétérogènes dans le modèle serverless

4. HeROcache – Applications serverless et coûts associés aux systèmes de stockage

5. HeROsim – Simuler pour élaborer et évaluer des politiques d'orchestration serverless

6. Conclusion

The collage consists of several rectangular panels, each containing a snippet of text from a scientific paper. The topics include:

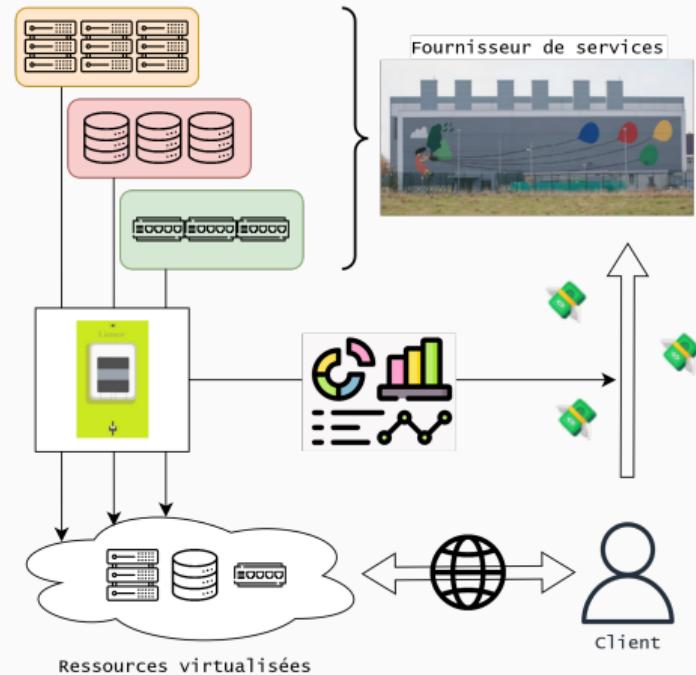
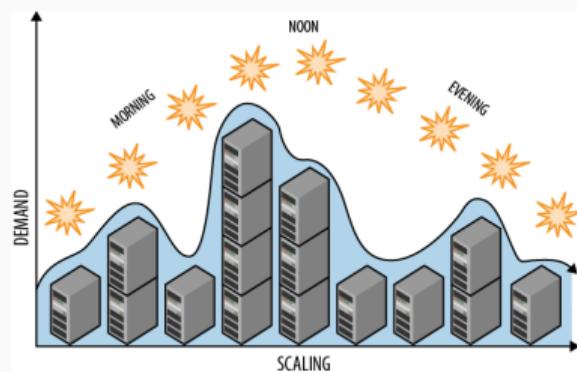
- Serverless Cloud Computing: State of the Art and Challenges** by Vincent Lemesle, I. (Abstract)
- HeROfake: Heterogeneous Resources Orchestration in a Serverless Cloud – An Application to Deepfake Detection** (Abstract)
- HeROcache: Storage-Aware Scheduling in Heterogeneous Serverless Edge – The Case of IDS Detection** (Abstract)
- HeROsim: An Allocation and Scheduling Simulator for Evaluating Serverless Orchestration Policies** (Abstract)

Each panel includes the author's name, affiliation, and a brief summary of the research findings or methodology.

Contexte

Cloud computing : une définition

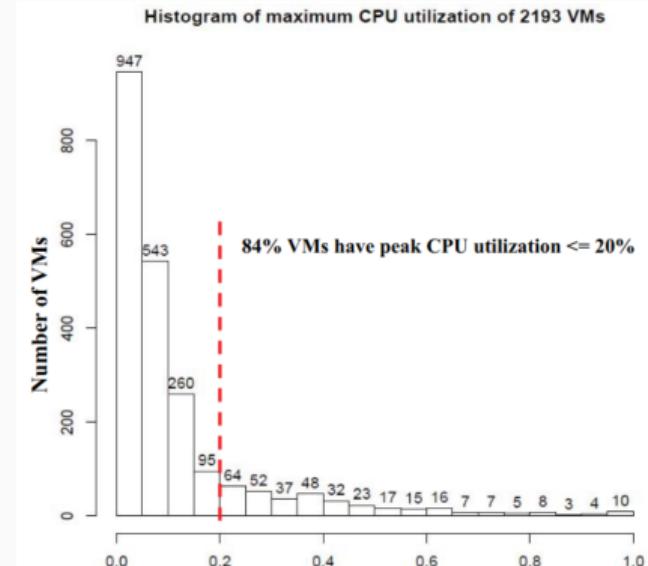
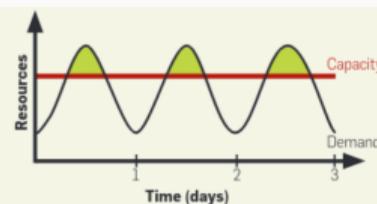
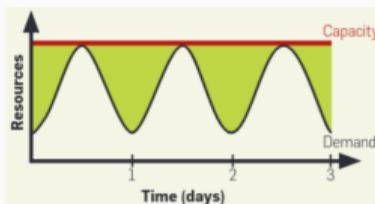
- Définition donnée par le NIST² :
 - Service à la demande;
 - Accessible par le réseau;
 - Ressources partagées;
 - Élasticité rapide;
 - Service mesuré.



2. [28] Peter MELL et Timothy GRANCE. « The NIST Definition of Cloud Computing ». National Institute of Standards and Technology Special Publication 800-145 (2011)
3. [44] Bill WILDER. Cloud Architecture Patterns : Using Microsoft Azure. 2012

Problématique : dimensionnement

- Dimensionnement aux pics
 - Over provisioning
- Exigences de QoS
 - Over committing
- Ressources dormantes
 - Under utilization

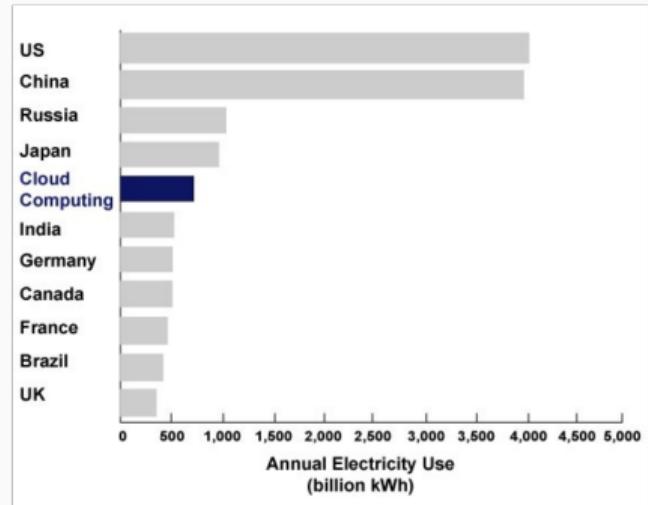


4. [2] Michael ARMBRUST et al. « A View of Cloud Computing ». *Commun. ACM* (2010)

5. [10] Rahul GHOSH et Vijay K. NAIK. « Biting Off Safely More Than You Can Chew : Predictive Analytics for Resource Over-Commit in IaaS Cloud ». *IEEE Cloud '12*. 2012

Problématique : consommation d'énergie

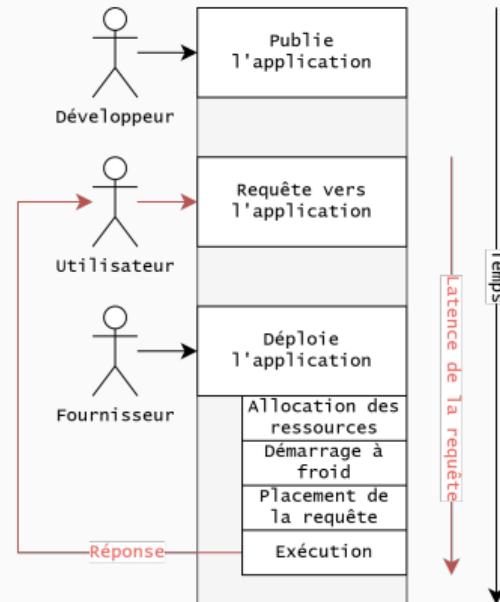
- Énergie dans le cloud :
 - 2010 : ~ 1,5% de la consommation mondiale (273 TWh)⁶;
 - ~ 4% de la demande d'énergie en Europe⁷;
- Utilisation des ressources :
 - Centres de données américains : 10 à 30% des serveurs **inactifs**⁸;
 - Taux d'utilisation des ressources cloud : <15%⁹;
 - Serveurs consomment 80% de leur énergie à 20% d'utilisation réelle¹⁰.
- Secteur en forte croissance : 15 à 20% par an d'ici à 2030¹¹ (tolérance aux pannes, prédiction de la demande future)



6. [25] Eric MASANET et al. « Recalibrating Global Data Center Energy-Use Estimates ». *Science* (2020)
7. [14] INTERNATIONAL ENERGY AGENCY. *Electricity 2024 - Analysis and Forecast to 2026*. Rapp. tech. 2024
8. [35] Arman SHEHABI et al. *United States Data Center Energy Usage Report*. Rapp. tech. 2016
9. [40] Arunchandar VASAN et al. « Worth Their Watts ? - An Empirical Study of Datacenter Servers ». HPCA '16. 2010
10. [32] Massoud PEDRAM. « Energy-Efficient Datacenters ». *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2012)
11. [13] INTERNATIONAL DATA CORPORATION. *Worldwide Spending on Public Cloud Services Is Forecast to Reach \$1.35 Trillion in 2027, According to New IDC Spending Guide*.
12. [4] Frédéric BORDAGE. *Numérique : 40 % du budget GES soutenable d'un européen*. 2021

Serverless computing : une définition

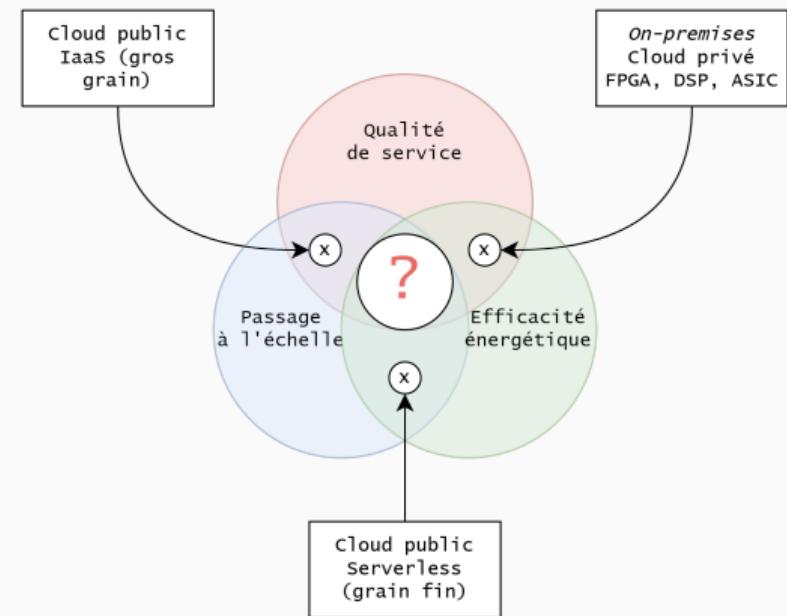
- Une nouvelle abstraction pour le cloud :
 - Unité d'allocation : **temps** plutôt que **ressource**;
 - Granularité fine : déploiement de **fonctions**;
 - Déplacement de la responsabilité : de l'utilisateur vers le fournisseur de services.



13. [33] Johann SCHLEIER-SMITH et al. « What Serverless Computing is and Should Become : The next Phase of Cloud Computing ». *Commun. ACM* (2021)

Cloud computing : une affaire de compromis

On-premises	IaaS	PaaS	Serverless
Fonctions	Fonctions	Fonctions	Fonctions
Application	Application	Application	Application
Environnement	Environnement	Environnement	Environnement
Système d'exploitation	Système d'exploitation	Système d'exploitation	Système d'exploitation
Virtualisation	Virtualisation	Virtualisation	Virtualisation
Plateforme d'exécution	Plateforme d'exécution	Plateforme d'exécution	Plateforme d'exécution
Stockage	Stockage	Stockage	Stockage
Réseau	Réseau	Réseau	Réseau

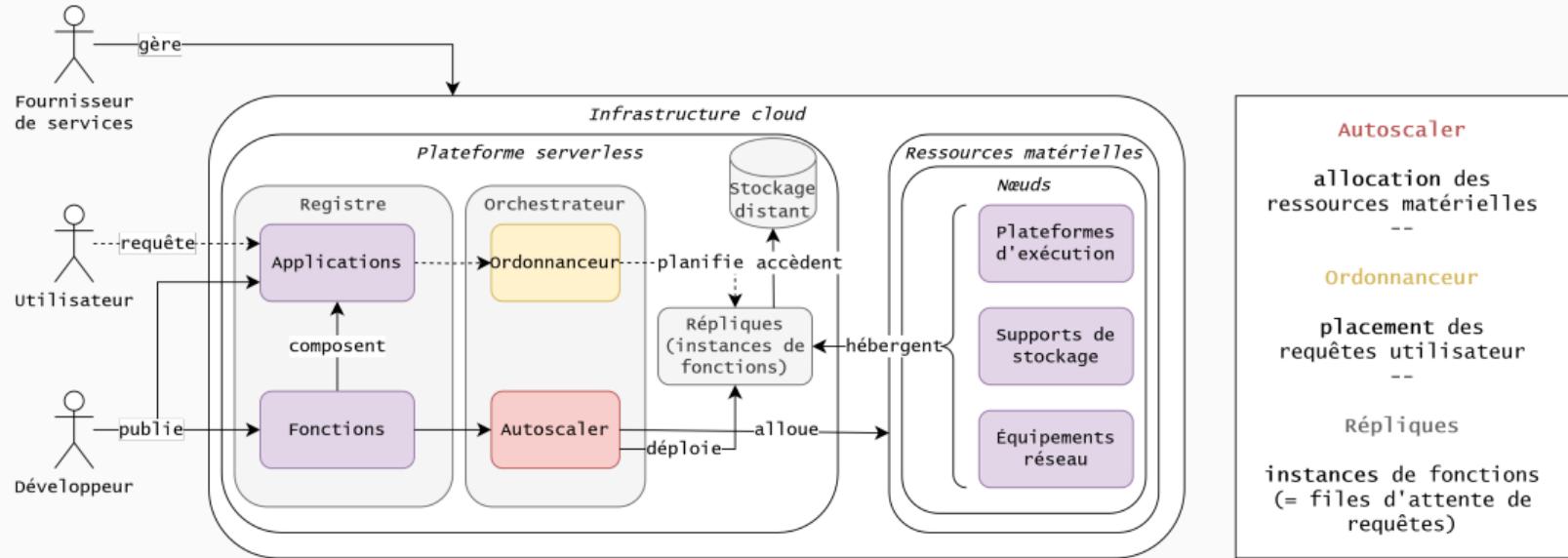


Serverless computing : une définition

Caractéristique	Serverful	Serverless
Architecture logicielle	Pas de contrainte	Découpage en fonctions sans état
Provisionnement	En fonction de l'offre	Géré par le fournisseur
Mise à l'échelle	<i>Capacity planning</i>	Dimensionnement automatique
Disponibilité	Dépend des ressources réservées	Nombreuses instances réparties
Tolérance aux fautes	Dépend de la stratégie de l'utilisateur	Garantie par le fournisseur
Concurrence	Dépend des ressources réservées	Virtuellement infinie
Mesure de l'utilisation	Ressources provisionnées	Ressources utilisées

État de l'art

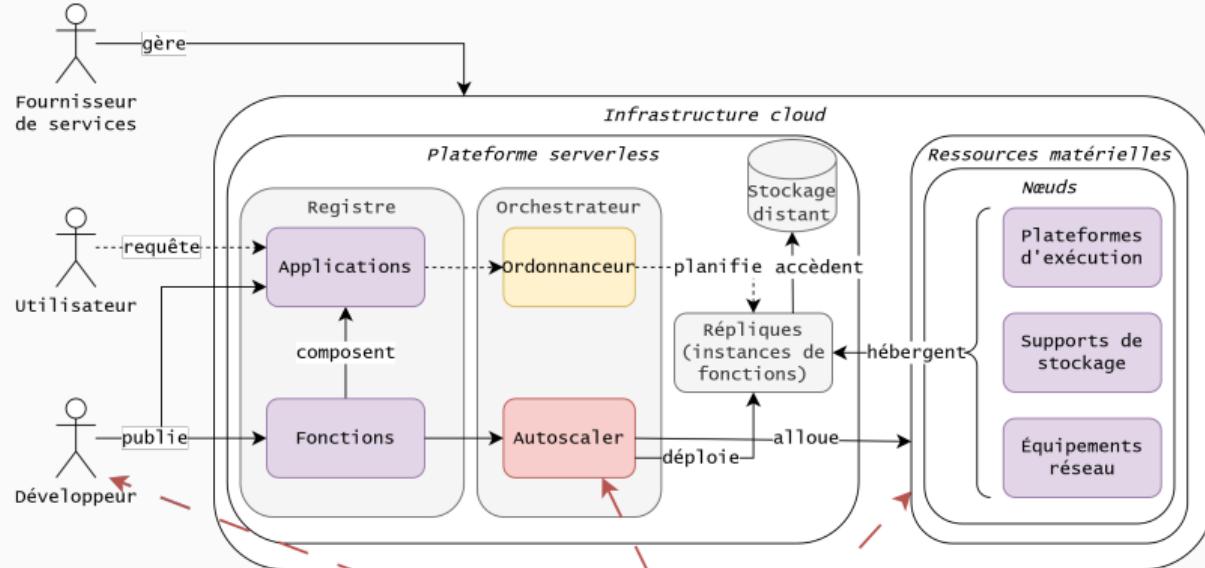
Serverless computing : plateforme considérée



15. Google Knative – <https://knative.dev>

16. Apache OpenWhisk – <https://openwhisk.apache.org>

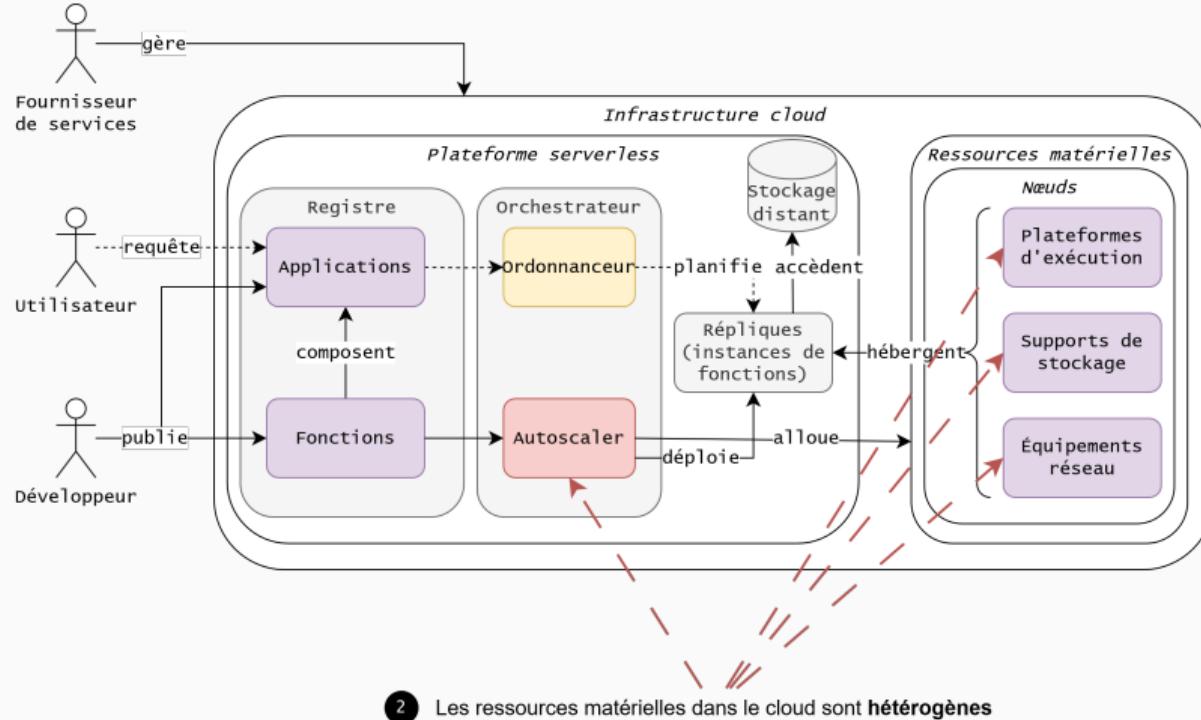
Serverless computing : défis



1 Les ressources matérielles ne sont pas réservées

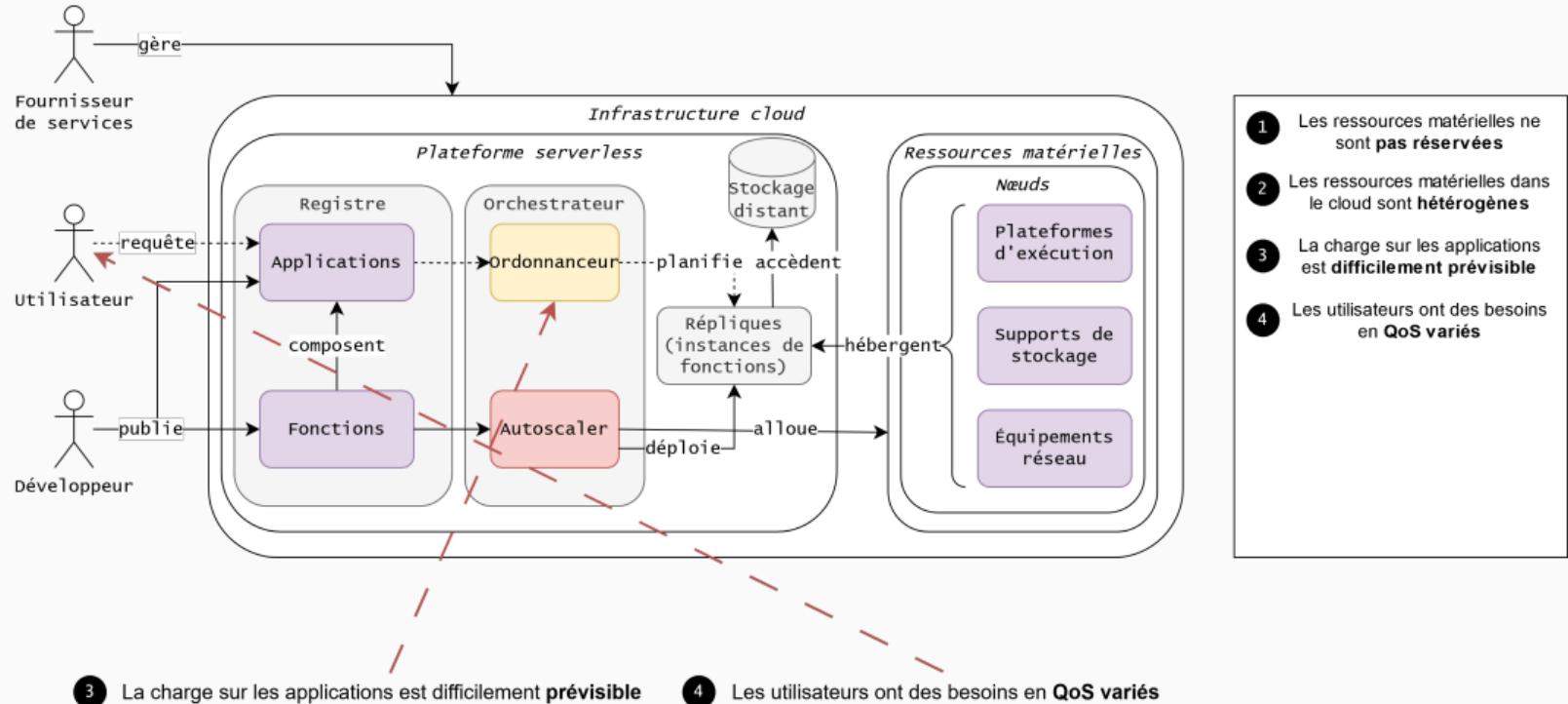
1 Les ressources matérielles ne sont pas réservées

Serverless computing : défis

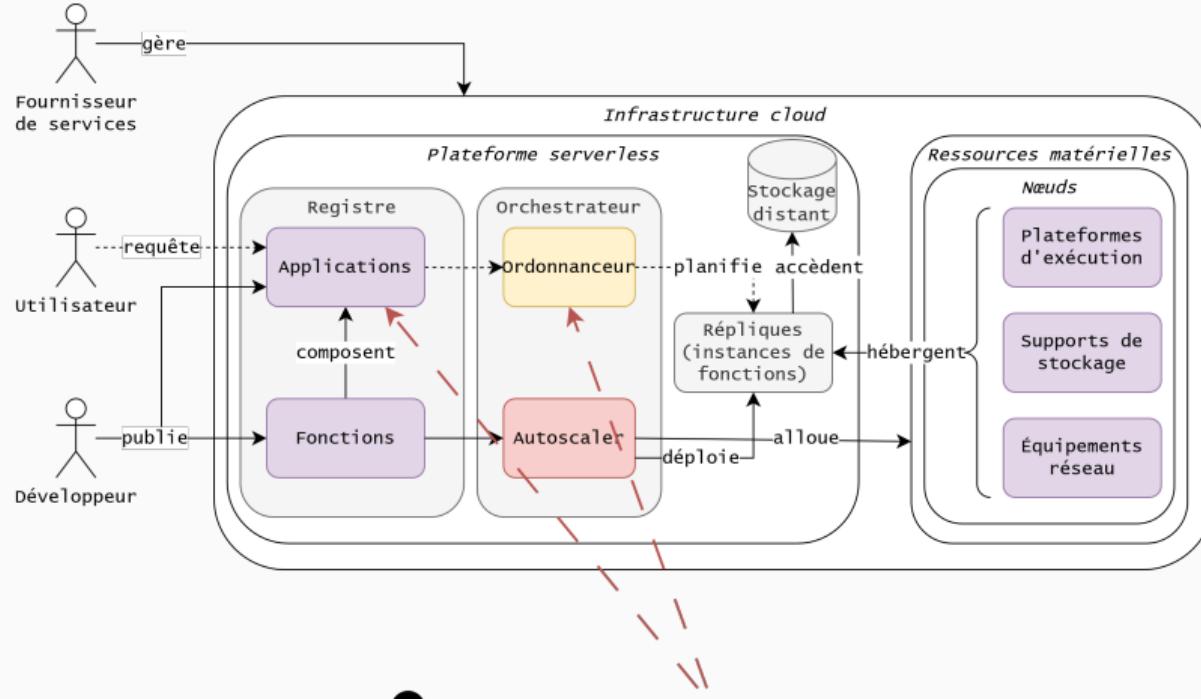


- 1 Les ressources matérielles ne sont pas réservées
- 2 Les ressources matérielles dans le cloud sont hétérogènes

Serverless computing : défis

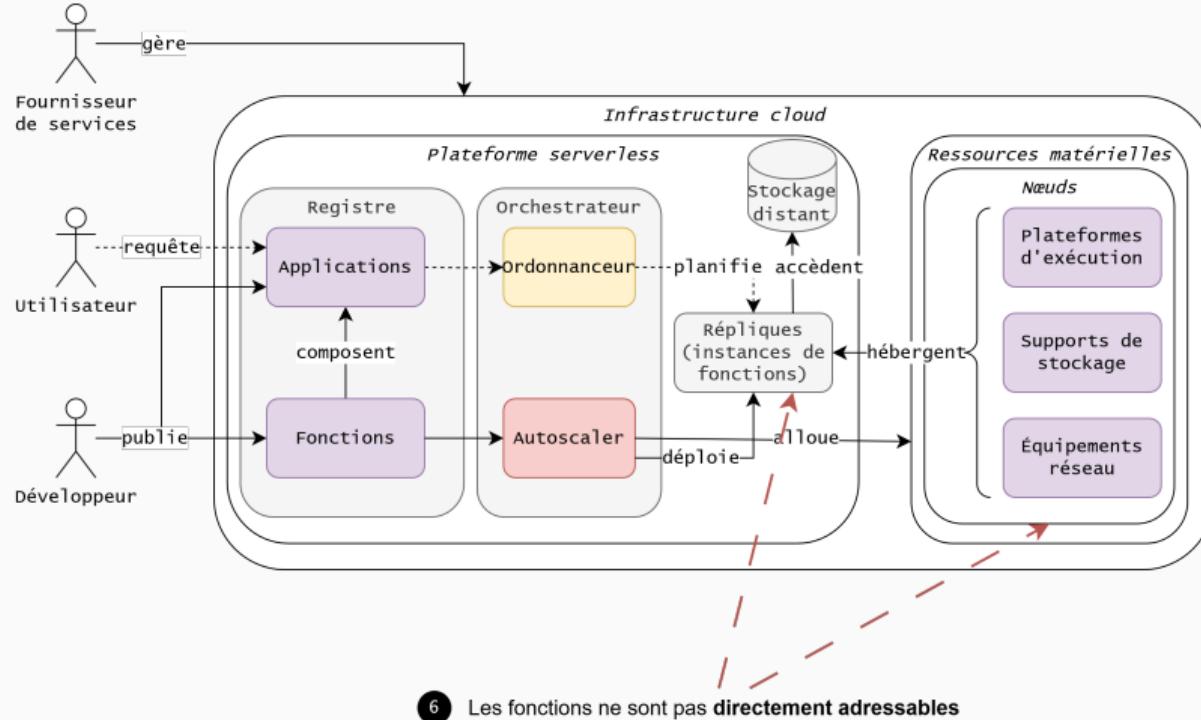


Serverless computing : défis



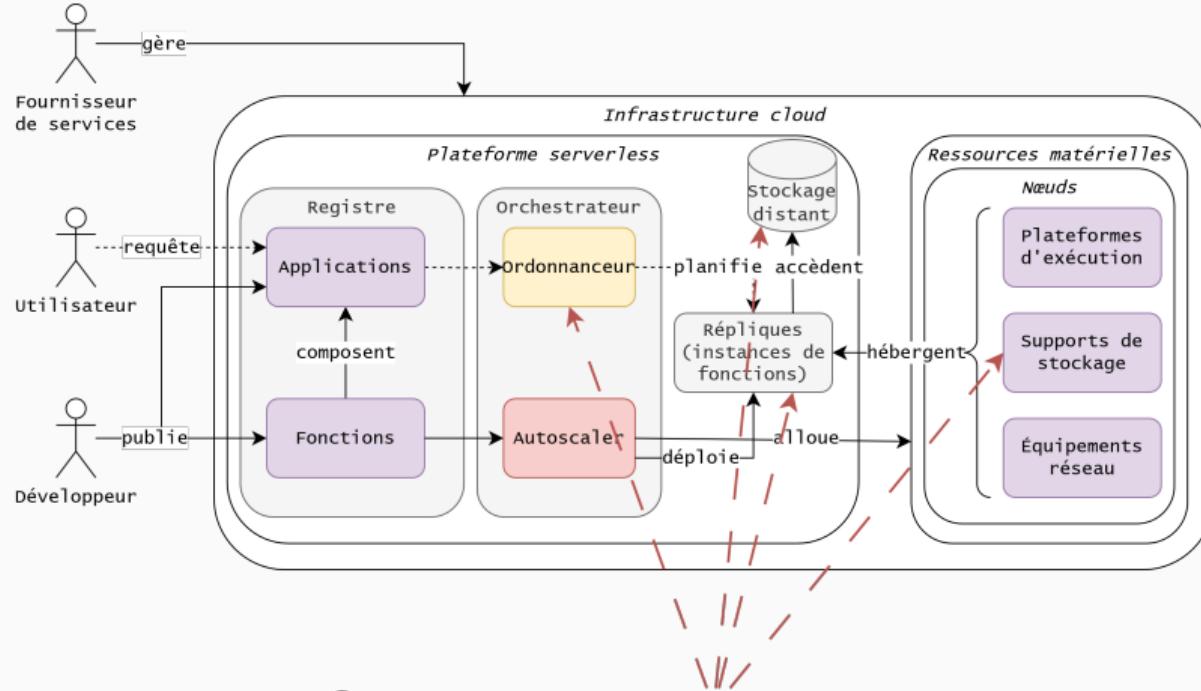
- 1 Les ressources matérielles ne sont **pas réservées**
- 2 Les ressources matérielles dans le cloud sont **hétérogènes**
- 3 La charge sur les applications est **difficilement prévisible**
- 4 Les utilisateurs ont des besoins en **QoS variés**
- 5 Les applications reposent sur la **composition** de fonctions

Serverless computing : défis



- 1 Les ressources matérielles ne sont **pas réservées**
- 2 Les ressources matérielles dans le cloud sont **hétérogènes**
- 3 La charge sur les applications est **difficilement prévisible**
- 4 Les utilisateurs ont des besoins en **QoS variés**
- 5 Les applications reposent sur la **composition** de fonctions
- 6 Les fonctions ne sont **pas directement adressables**

Serverless computing : défis



7. Les résultats intermédiaires sont communiqués au travers du stockage

Serverless computing : opportunités et menaces

Propriété	Défis	Opportunité	Menace	Contribution
Dimensionnement dynamique des ressources en fonction du besoin réel	<ul style="list-style-type: none">1 Les ressources matérielles ne sont pas réservées2 Les ressources matérielles dans le cloud sont hétérogènes3 La charge sur les applications est difficilement prévisible4 Les utilisateurs ont des besoins en QoS variés	Optimisation de l' utilisation des ressources et de l'énergie	Pics de latence (démarrages à froid)	HeROfake
Placement dynamique des requêtes utilisateur sur les répliques	<ul style="list-style-type: none">5 Les applications reposent sur la composition de fonctions6 Les fonctions ne sont pas directement adressables7 Les résultats intermédiaires sont communiqués au travers du stockage	Respect de la Qualité de Service au plus proche des besoins	Baisse de débit (communications inter-fonctions)	HeROcache

HeROfake – Orchestrer des fonctions sur ressources hétérogènes dans le modèle serverless

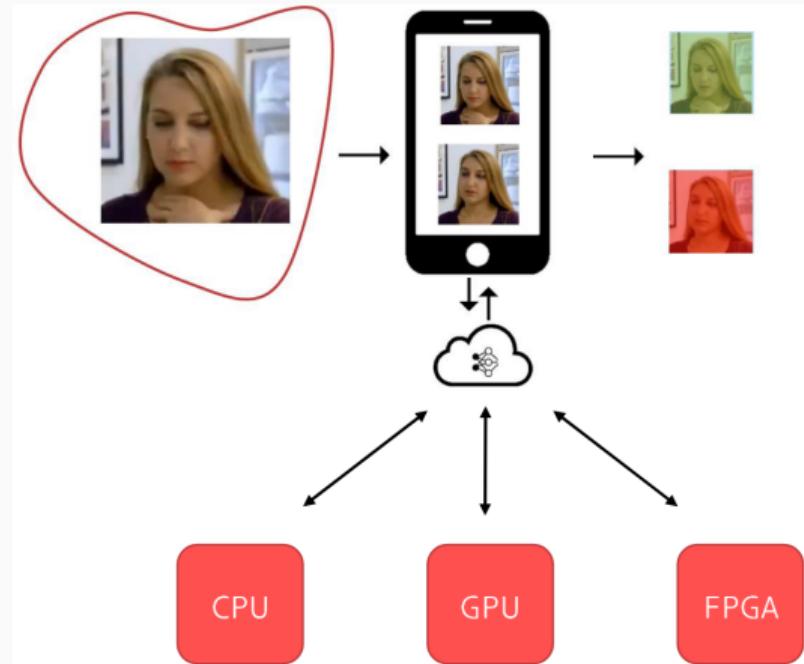
HeROfake – Cas d'usage

Cas d'usage : détection de deepfake

- Détection à la demande, application dirigée par les événements;
- Traitements sans état (images) ou avec état (vidéos).

Détection de deepfake "*as a service*"

- Différents niveaux de Qualité de Service :
 - Utilisation par des particuliers (réseaux sociaux, etc.);
 - Usages critiques (autorités, médias);
- Possibilité d'accélération à la demande (GPU, FPGA);
- Ressources partagées et contraintes.



Question de recherche 1 (QR1)

Comment dimensionner les allocations de ressources hétérogènes pour une application simple, constituée de fonctions de courte durée, et comment ordonner efficacement les requêtes des utilisateurs, lorsque ces derniers ont des besoins variés en matière de qualité de service ?

HeROfake – Travaux connexes

Table 1 : État de l'art des solutions de déploiement avec mise à l'échelle automatique pour des tâches de courte durée.

	Serverless	Déploiement cible	QoS	Hétérogénéité matérielle	Utilisation des ressources	Énergie	Conscient du coût
Swayam [11]	✗	Public (Azure)	✓	✗	✓	✗	✗
Pigeon [20]	✓	Privé	✗	✓	✓	✗	✗
MArk [46]	✗	Public (AWS)	✓	✓	✓	✗	✓
ENSURE [38]	✓	Privé	✗	✗	✓	✗	✓
Mampage et al. [24]	✓	Privé	✓	✗	✓	✗	✓
Atoll [36]	✓	Privé	✓	✗	✗	✗	✗
INFless [45]	✓	Privé	✓	✗	✓	✗	✓
SMIF [9]	✓	Privé	✓	✓	✓	✗	✗
Solution cible	✓	Privé	✓	✓	✓	✓	✓

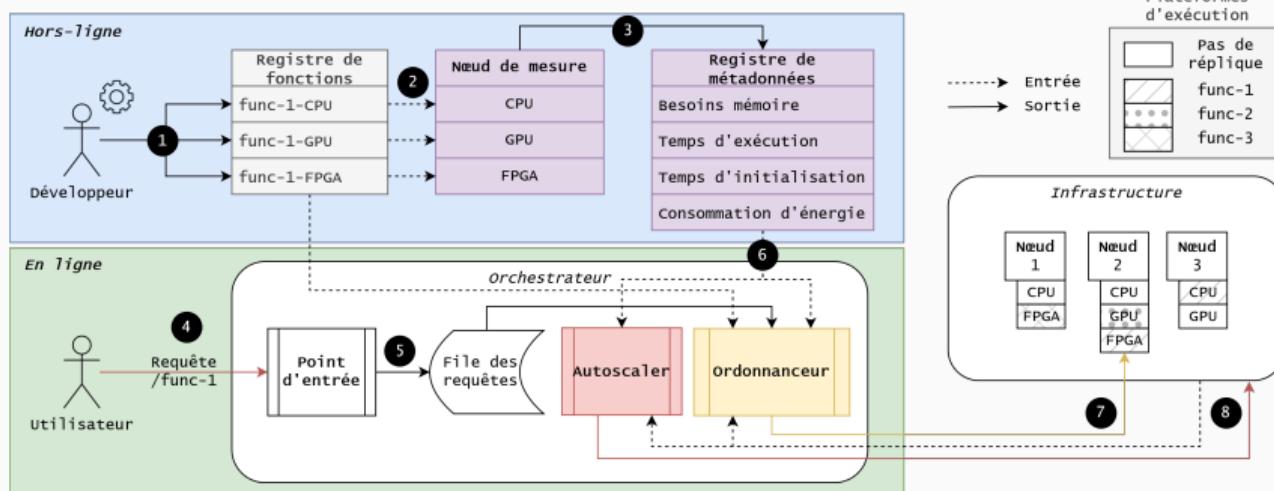
HeROfake – Système considéré

Caractérisation des fonctions (hors-ligne)

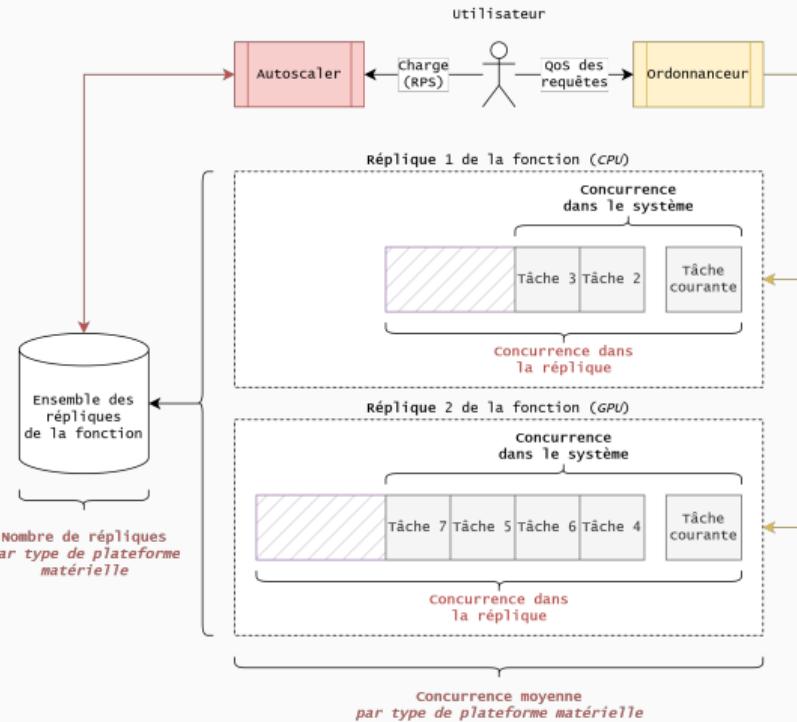
- Générer des métadonnées pour les fonctions
- Utiles pour guider les décisions d'allocation et d'ordonnancement

Orchestration des fonctions (en ligne)

- Allocation des ressources au plus près des besoins
- Éviter les violations de QoS
- Minimiser la consommation d'énergie



HeROfake – Contribution : Mise à l'échelle dynamique



Knative²³, répliques homogènes :

$$\text{replicaCount}_f = \frac{\text{inSystemConcurrency}_f}{\text{concurrencyTarget}_f}$$

Hétérogénéité matérielle :

$$\text{replicaCount}_{f_h} = \frac{\text{inSystemConcurrency}_{f_h}}{\text{concurrencyTarget}_{f_h}}$$

Ratio composite pour seuil de concurrence :

$$cT_{f,h} = cT_{f,c} \cdot (k_{ET} \cdot \frac{ET_{f_c}}{ET_{f_h}} + k_{EC} \cdot \frac{EC_{f_c}}{EC_{f_h}} + k_{HP} \cdot \frac{HP_{f_c}}{HP_{f_h}})$$

HeROfake – Contribution : Stratégie de minimisation des coûts

• Autoscaling

- Minimiser le coût des démarrages à froid, la consommation d'énergie et le coût de possession total :

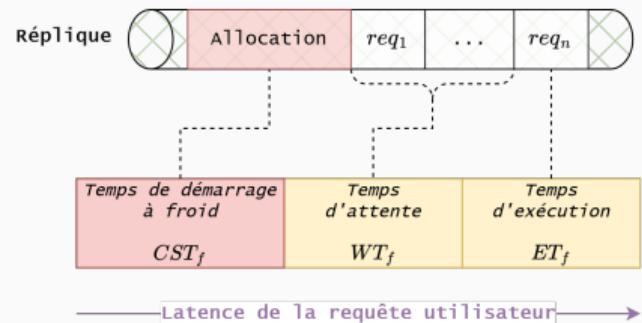
$$\begin{aligned}\forall N, \forall P \in N, \text{scaleCost}_{f_{N,P}} = & k_{TT} \cdot TT_{f_{N,P}} \\ & + k_{EC} \cdot EC_{f_{N,P}} \\ & + k_{HP} \cdot HP_{f_{N,P}}\end{aligned}$$

• Ordonnancement

- Minimiser les pénalités sur QoS, la consommation d'énergie et la dispersion des tâches :

$$\begin{aligned}\forall (N, P) \in R_f, \text{schedCost}_{f_{N,P}} = & k_{QP} \cdot QP_{f_{N,P}} \\ & + k_{EC} \cdot EC_{f_{N,P}} \\ & + k_{TC} \cdot TC_{f_{N,P}}\end{aligned}$$

Mobiliser les métadonnées issues de la phase de caractérisation :



$$TT_{f_{N,P}} = CST_{f_{N,P}} + WT_{f_{N,P}} + ET_{f_{N,P}}$$

HeROfake – Évaluation : Méthodologie

Évaluation en deux phases :

- Comparaison aux politiques de référence
- Impact des différents composants

Infrastructure retenue :

- 10 nœuds (10 CPUs, 6 GPUs, 2 FPGAs);

Scénario d'évaluation :

- 50 000 requêtes;
- 50 à 100 requêtes par seconde;
- Distribution uniforme des niveaux de QoS requis.

Autoscalers :

- HeROfake (**HRO**) – Conscience de l'hétérogénéité matérielle pour les performances et la consommation d'énergie;
- Knative (**KN**) – Sélection du nœud le plus disponible, *i.e.* équilibrage de charge.

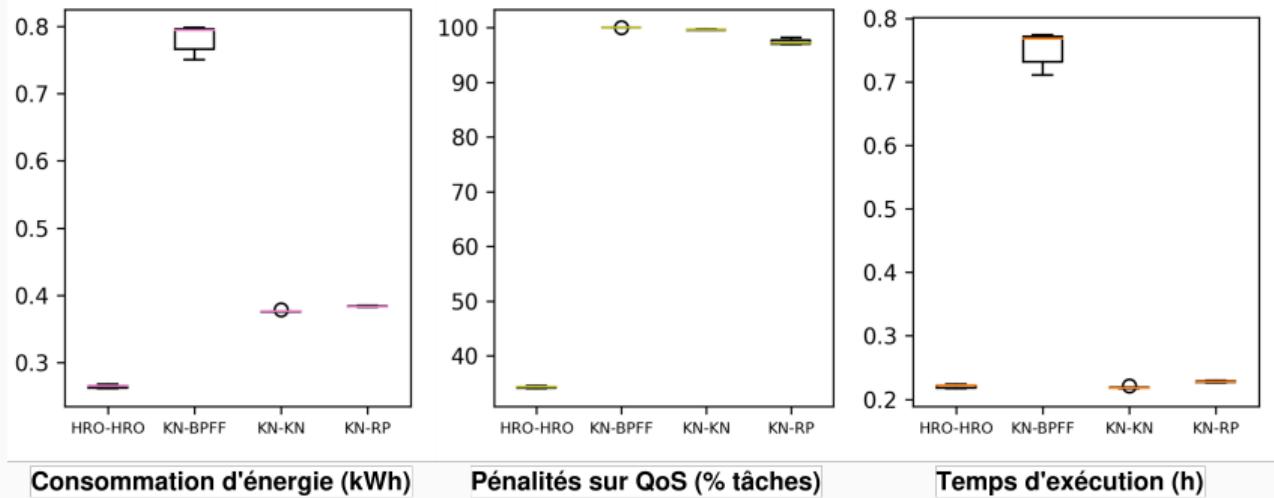
Ordonnanceurs :

- HeROfake (**HRO**) – Conscience de l'hétérogénéité matérielle et des échéances pour le respect de la QoS;
- Knative (**KN**) – Répliques homogènes, sélection par file d'attente la plus courte²⁴;
- Bin-Packing First Fit (**BPFF**) – Ordonnanceur dans AWS Lambda, sélection par file d'attente la plus chargée²⁵;
- Random Placement (**RP**) – Sélection d'une réplique aléatoire.

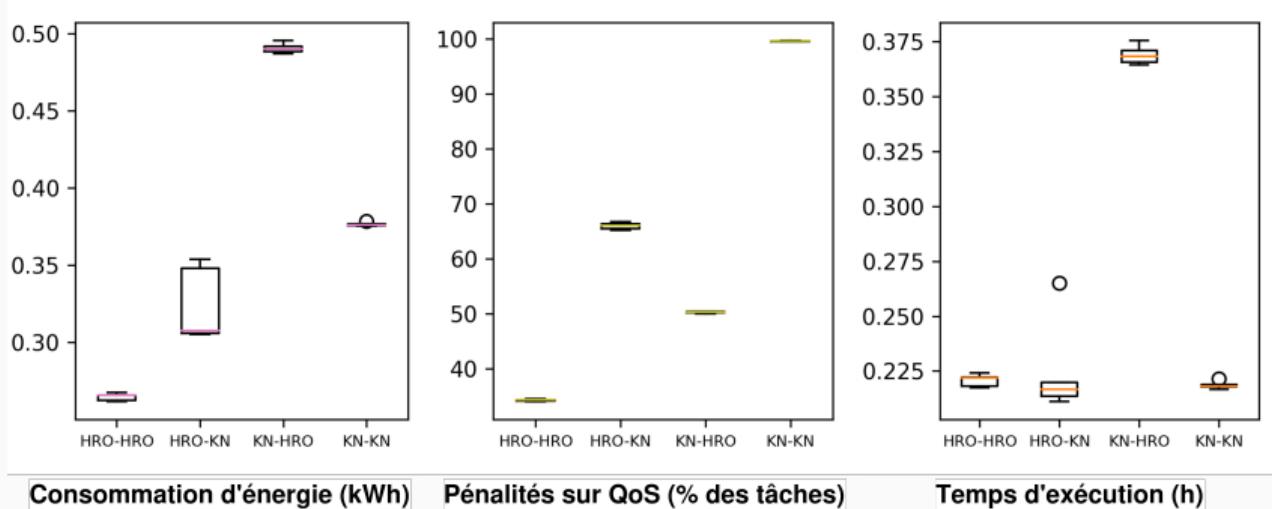
24. [38] Amoghavarsha SURESH et al. « ENSURE : Efficient Scheduling and Autonomous Resource Management in Serverless Environments ». ACSOS '20. 2020

25. [41] Liang WANG et al. « Peeking Behind the Curtains of Serverless Platforms ». USENIX ATC'18. 2018

HeROfake – Évaluation : Analyse des résultats i

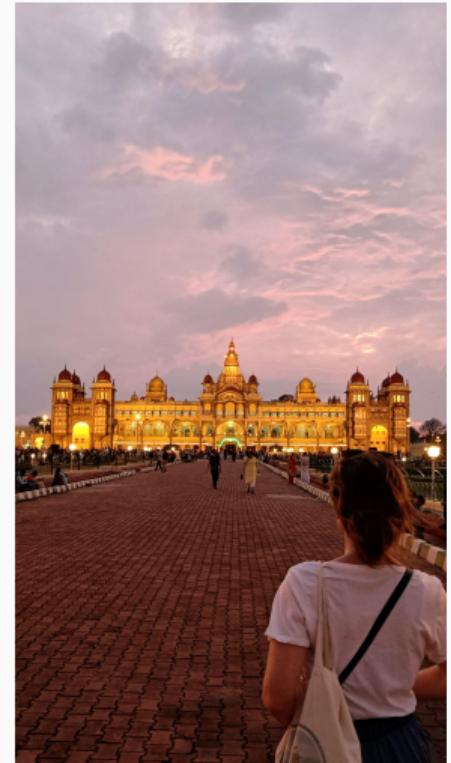


HeROfake – Évaluation : Analyse des résultats ii



HeROfake – Conclusion

- L'orquestrateur HeROfake donne un *makespan* comparable à Knative, mais :
 - Réduit les **pénalités sur QoS** de 65%;
 - **Consolide** les tâches sur 30% des nœuds de l'infrastructure;
 - Réduit la consommation d'**énergie dynamique** de plus de 35%.
- L'approche de l'orquestration par les coûts offre une alternative aux stratégies de référence :
 - Notre ordonnanceur seul permet de **diminuer de moitié** les violations de QoS en allouant uniquement des CPU.
- Publication à la conférence *CCGrid'23* :
 - [17] Vincent LANNURIEN et al. « HeROfake : Heterogeneous Resources Orchestration in a Serverless Cloud – An Application to Deepfake Detection ». 2023



HeROcache – Applications serverless et coûts associés aux systèmes de stockage

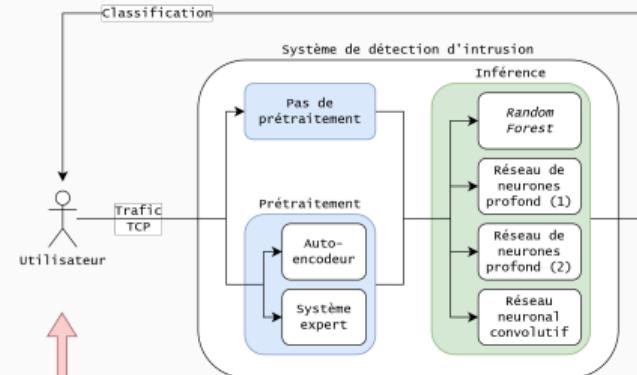
HeROcache – Cas d'usage

Projet AID DISPEED : Systèmes de détection d'intrusion

- Utilisation intermittente de ressources contraintes :
 - IDS déployés à l'edge, dans le cadre de missions pour des drones de surface;
- Les IDS reposent sur l'apprentissage automatique :
 - Random Forests, réseaux de neurones;
 - Accélération matérielle.

Défis à relever

- Ordonnancer des chaînes de fonctions;
- Communication de résultats intermédiaires;
- Images de fonctions lourdes (plusieurs Go);
- Temps d'exécution très courts (millisecondes).



. DISPEED : Détection d’Intrusion et compromis Sécurité/Performance/Énergie, Étude pour les meutes de Drones

Question de recherche 2 (QR2)

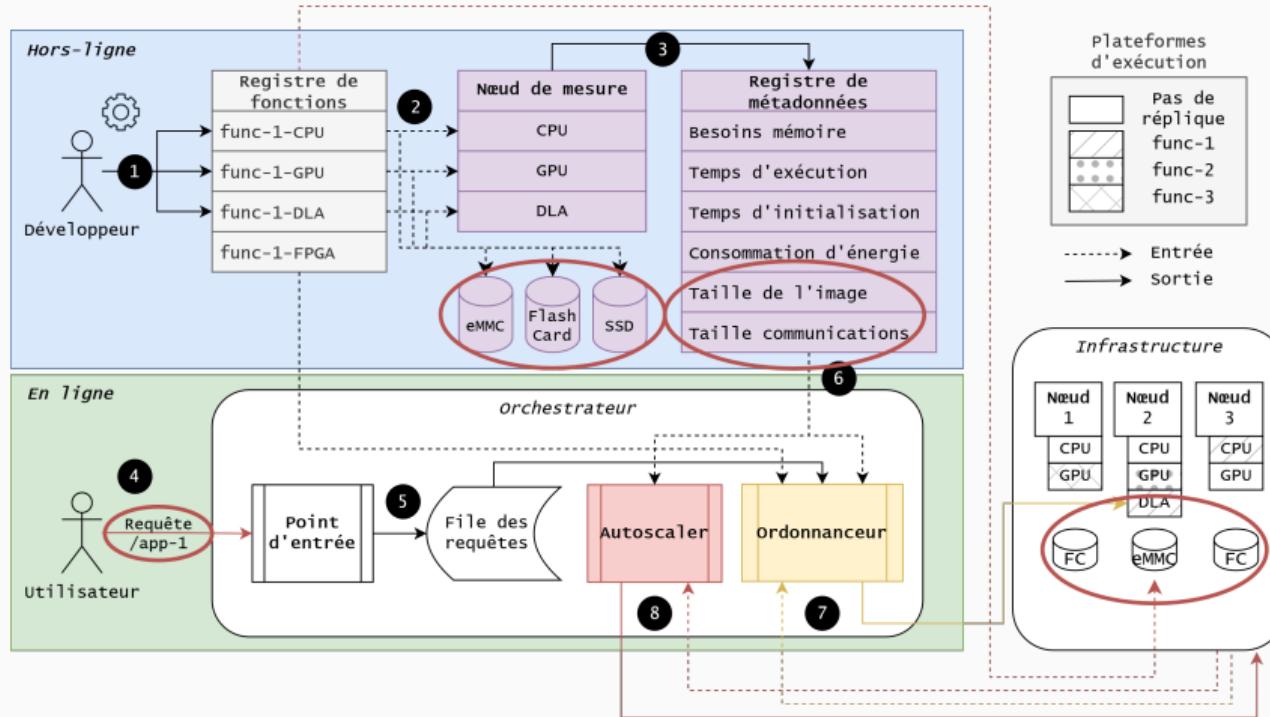
Comment déployer des **applications complexes**, composées de chaînes de fonctions de courte durée, et comment tirer parti de l'**hétérogénéité** des nœuds disponibles à l'edge, pour respecter la **qualité de service** requise par les utilisateurs tout en contenant la **consommation d'énergie** de l'infrastructure ?

HeROcache – Travaux connexes

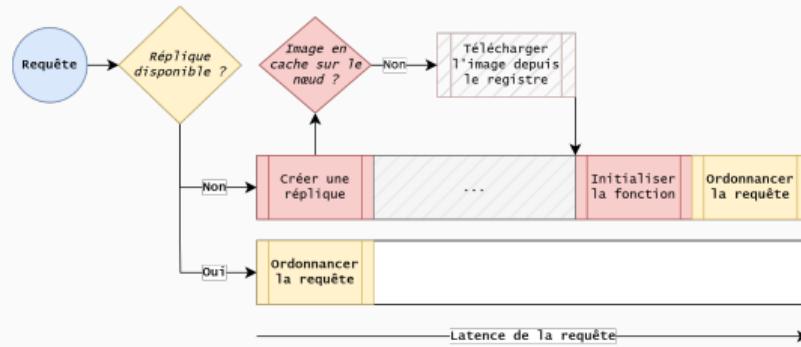
Table 2 : État de l'art des plateformes d'orchestration prenant en compte les données.

	Chaînes de fonctions	QoS par requête	Hétérogénéité matérielle	Contrainte de programmation	Énergie	Cache de fonctions	Communications
Cypress [3]	✓	✓	✗	✓	✓	✗	✓
FaDO [37]	✗	✗	✗	✓	✗	✗	✓
FaaSFlow [19]	✓	✗	✗	✗	✗	✗	✗
FIRST [47]	✗	✗	✗	✓	✓	✗	✗
HeROfake [17]	✗	✓	✓	✓	✓	✗	✗
Netherite [5]	✓	✗	✗	✓	✗	✗	✓
Palette [1]	✓	✗	✗	✗	✗	✓	✓
Solution cible	✓	✓	✓	✓	✓	✓	✓

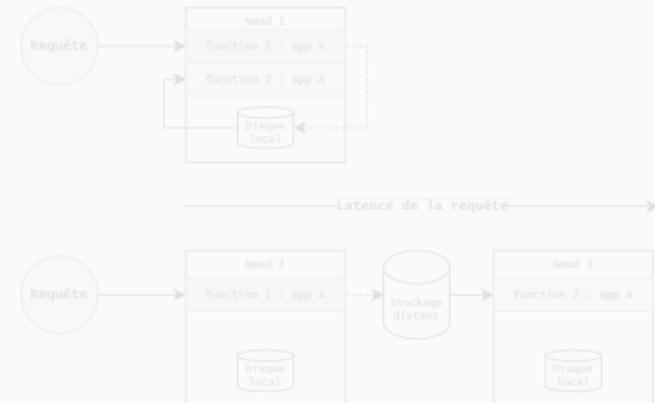
HeROcache – Système considéré



HeROcache – Contribution : Cache et communications



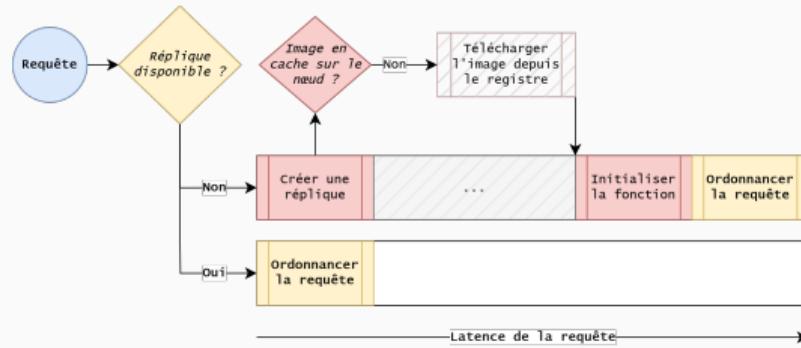
Mise en cache des images de fonctions sur les nœuds pour accélérer les démaragements à froid



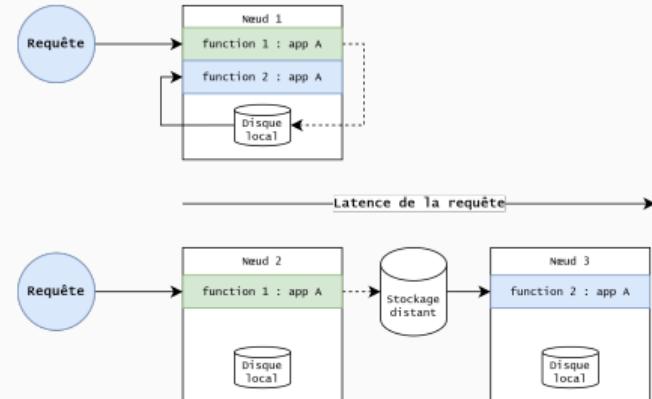
Consolidation des fonctions d'une même application pour favoriser les communications locales

Éviter la contention sur le stockage local aux nœuds

HeROcache – Contribution : Cache et communications



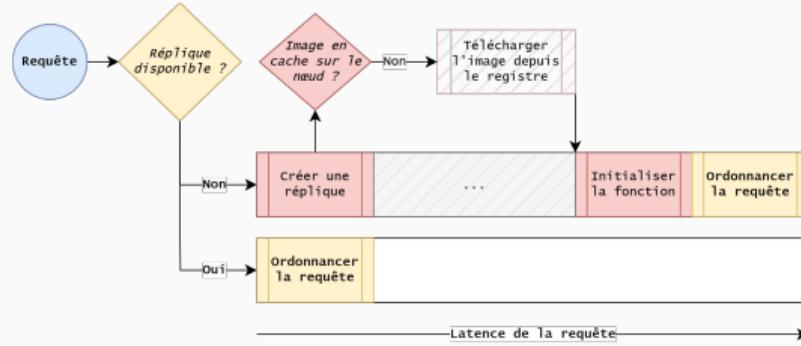
Mise en cache des images de fonctions sur les nœuds pour accélérer les démarrages à froid



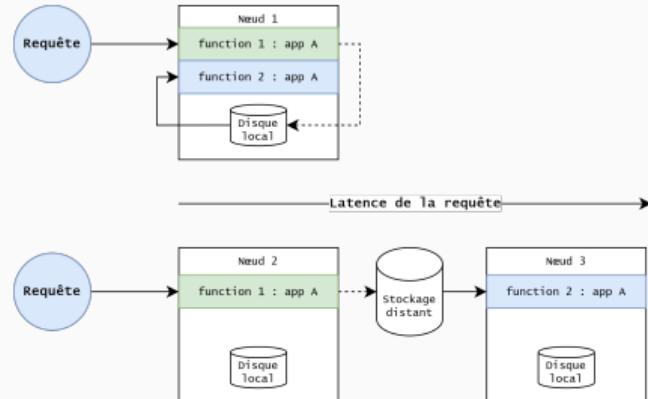
Consolidation des fonctions d'une même application pour favoriser les communications locales

Éviter la contention sur le stockage local aux nœuds

HeROcache – Contribution : Cache et communications



Mise en cache des images de fonctions sur les nœuds pour accélérer les démarrages à froid



Consolidation des fonctions d'une même application pour favoriser les communications locales

Éviter la contention sur le stockage local aux nœuds

HeROcache – Contribution : Stratégie de minimisation des coûts

- Autoscaling

- Minimiser les défauts de cache, la consommation d'énergie et le coût de possession total :

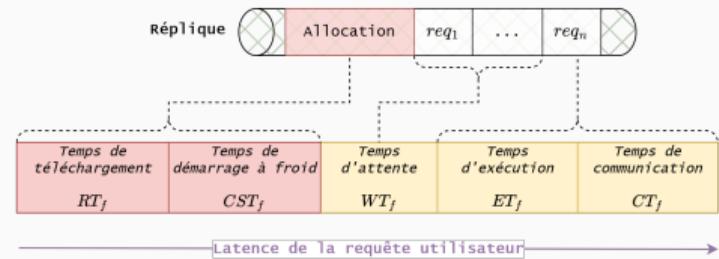
$$\forall N, \forall P \in N, scaleCost_a^{f_{N,P}} = k_{CP} \cdot CP_{aN} + k_{TT} \cdot TT_{f_{N,P}} + k_{EC} \cdot EC_{f_{N,P}} + k_{HP} \cdot HP_{f_{N,P}}$$

- Ordonnancement

- Minimiser les pénalités sur QoS, la consommation d'énergie et la dispersion des tâches :

$$\forall (N, P) \in R_f, schedCost_{f_{N,P}} = k_{QP} \cdot QP_{f_{N,P}} + k_{EC} \cdot EC_{f_{N,P}} + k_{TC} \cdot TC_{f_{N,P}}$$

Mobiliser les métadonnées issues de la phase de caractérisation :



$$TT_{f_{N,P}} = RT_{f_{N,P}} + WT_{f_{N,P}} + CST_{f_{N,P}} + ET_{f_{N,P}} + CT_{f_{N,P}}$$

HeROcache – Évaluation : Méthodologie

Politiques évaluées pour la **mise à l'échelle** :

- HeROcache (**HRC**) – Autoscaler conscient des coûts de stockage (images de fonctions, communications entre fonctions);
- HeROfake (**HRO**) – Autoscaler sans conscience des coûts du stockage;
- Knative (**KN**) – Équilibrage de charge.

Infrastructure retenue :

- 10 nœuds edge : 8x Raspberry Pi 4B, 1x Nvidia Jetson Xavier AGX, 1x PYNQ-Z2 (*System on Chip* avec FPGA).

Scénario d'évaluation :

- Processus de Poisson, $\lambda = 83$ (12,5 Mo/s, 4G LTE);
- 10 minutes de requêtes utilisateur.

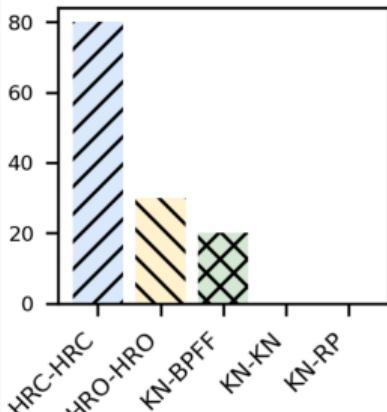
Politiques évaluées pour l'**ordonnancement** :

- HeROcache (**HRC**) – Ordonnanceur conscient des dépendances entre les tâches;
- HeROfake (**HRO**) – Ordonnanceur sans conscience des dépendances;
- Knative (**KN**) – Répliques homogènes, équilibrage de charge²⁶;
- Bin-Packing First Fit (**BPFF**) – Ordonnanceur AWS Lambda, consolidation maximale²⁷;
- Random Placement (**RP**) – Sélection d'une réplique aléatoire.

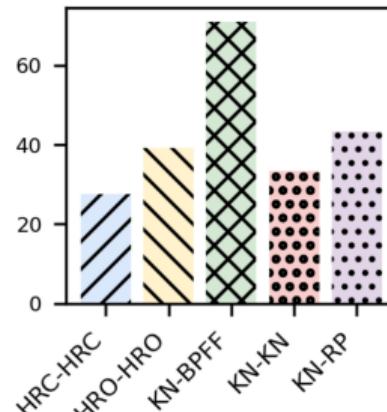
26. [38] Amoghavarsha SURESH et al. « ENSURE : Efficient Scheduling and Autonomous Resource Management in Serverless Environments ». ACSOS '20. 2020

27. [41] Liang WANG et al. « Peeking Behind the Curtains of Serverless Platforms ». USENIX ATC'18. 2018

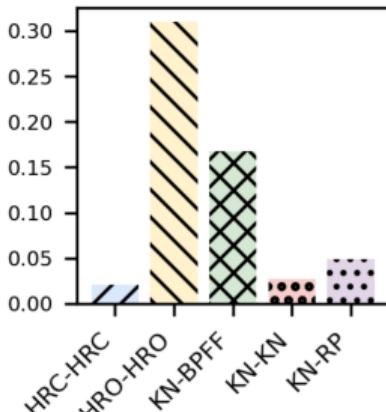
HeROcache – Évaluation : Analyse des résultats



Nœuds inutilisés

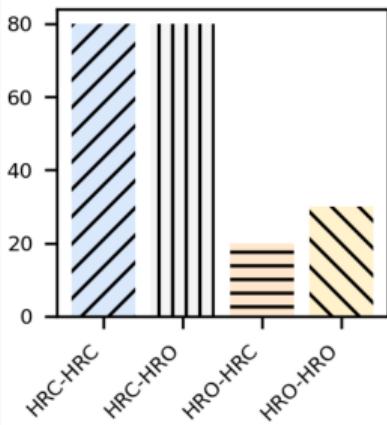


Pénalités sur QoS (% des tâches)

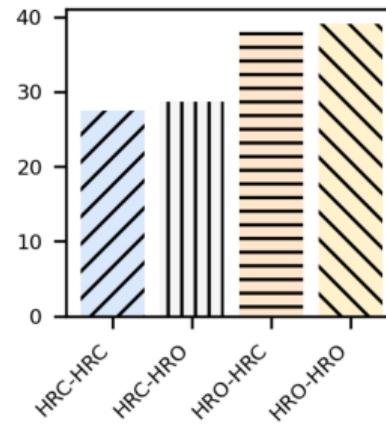


Consommation d'énergie (kWh)

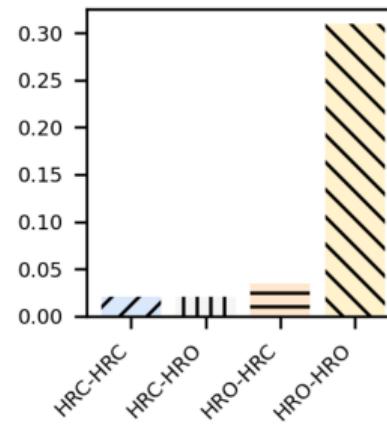
HeROcache – Évaluation : Analyse des résultats



Nœuds inutilisés



Pénalités sur QoS (% des tâches)



Consommation d'énergie (kWh)

HeROcache – Conclusion

- HeROcache consolide les fonctions des mêmes applications :
 - réduction des délais d'**initialisation** de **17,6%** en moyenne;
 - réduction des délais de **communication** de **88,4%**;
- HeROcache améliore la Qualité de Service :
 - maintien des **violations de QoS** sous les **28%** des requêtes;
 - potentiel de réduction de la consommation d'**énergie statique** de **80%**.
- Publication à la conférence *CCGrid'24* :
 - [16] Vincent LANNURIEN et al. « HeROcache : Storage-Aware Scheduling in Heterogeneous Serverless Edge – The Case of IDS ». 2024



HeROsim – Simuler pour élaborer et évaluer des politiques d'orchestration serverless

Question de recherche 3 (QR3)

Du point de vue d'un fournisseur de services pour le cloud, comment évaluer et comparer l'impact sur la qualité de service de différentes politiques d'allocation de ressources et d'ordonnancement de tâches dans le modèle serverless ?

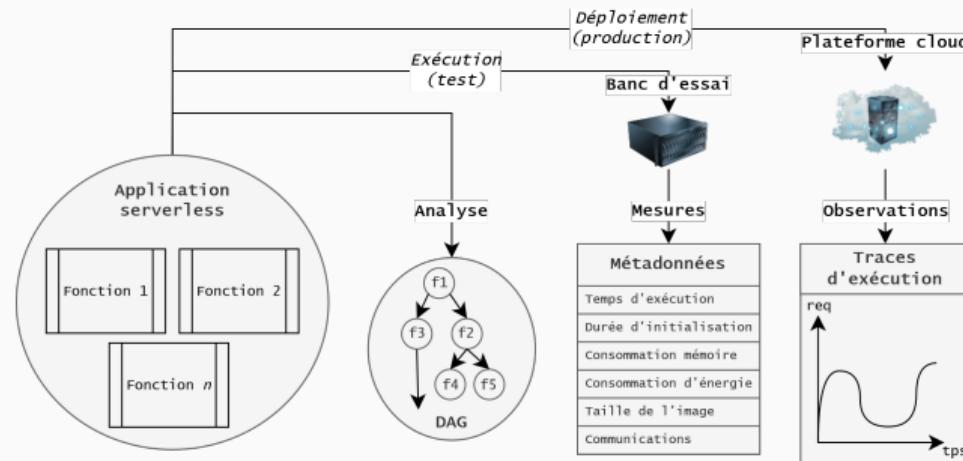
HeROsim – Travaux connexes

Table 3 : État de l'art des outils de simulation pour l'orchestration des ressources et des charges de travail dans le cloud.

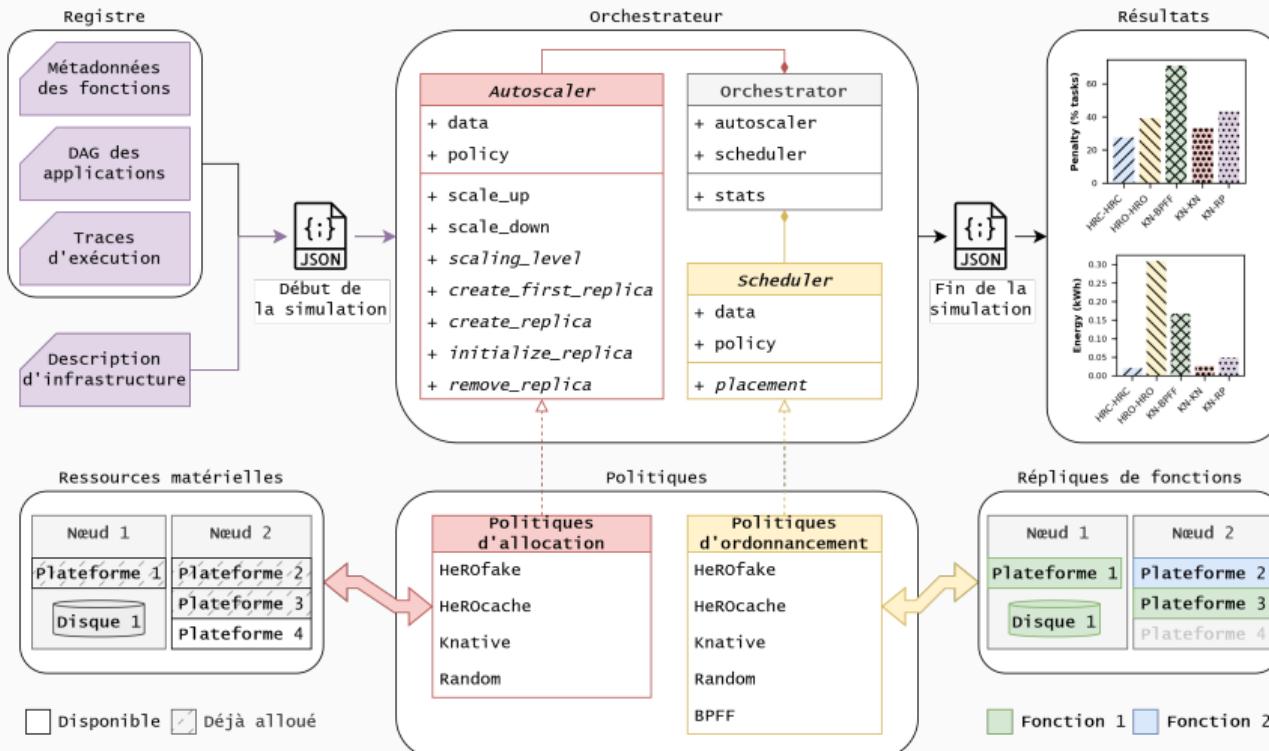
	Serverless	Déploi-ments	Chaînes de fonctions	Hétérogé-néité matérielle	QoS par requête	Énergie	Visualisation
CloudSim [8]	✗	Public, privé, hybride	✗	✓	✗	✓	✗
CloudSimSC [23]	✓	Public, privé, hybride	✗	✓	✗	✓	✗
CloudAnalyst [43]	✗	Public, privé, hybride	✗	✓	✗	✓	✓
DFaaSCloud [15]	✓	Hybride multi-strates	✗	✗	✓	✗	✓
ElasticSim [7]	✗	Public	✓	✗	✗	✗	✓
GridSim [6]	✗	Grille	✗	✓	✓	✗	✓
iCanCloud [31]	✗	Public	✗	✗	✗	✗	✓
iFogSim2 [22]	✗	Edge, Fog	✗	✓	✗	✓	✗
OpenDC 2.0 [26]	✓	Public, privé, hybride	✓	✓	✗	✓	✓
SimFaas [21]	✓	Public	✗	✗	✗	✓	✓
Solution cible	✓	Privé	✓	✓	✓	✓	✓

HeROsim – Contribution

- **Granularité** : tracer les événements d'orchestration au niveau d'une requête utilisateur;
- **Charges de travail** : modéliser les dépendances temporelles et de données;
- **Hétérogénéité** : des ressources matérielles, des besoins en Qualité de Service;
- **Reproductibilité** : rejouer des traces d'exécution;
- **Explicabilité** : comparer des politiques au regard de métriques de performance.

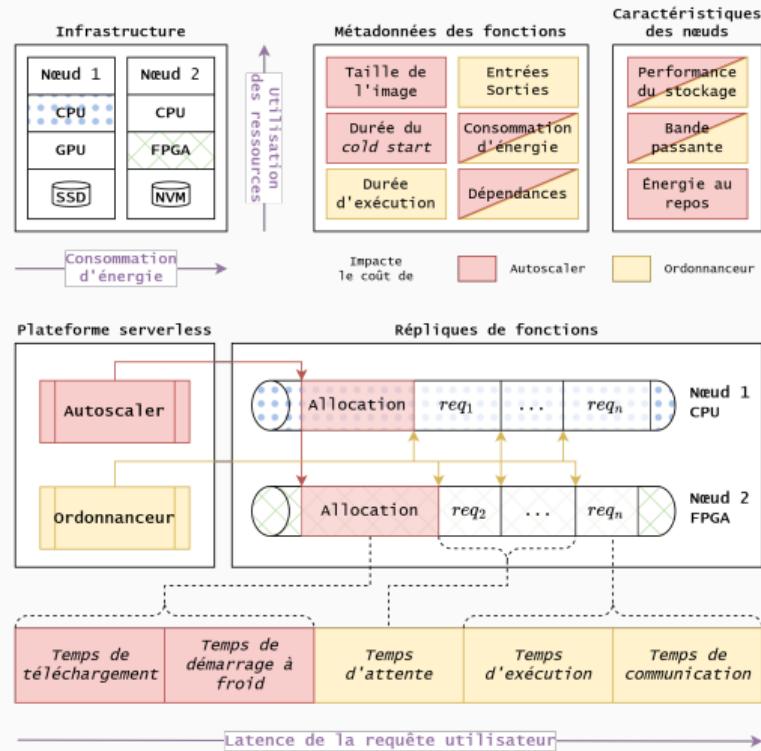


HeROsim – Architecture logicielle



HeROsim – Modèle de coût et validation

- HeROsim est un simulateur à événements discrets;
- L'**horloge** de la simulation n'avance que sur la base de données préalablement mesurées;
- La précision des estimations dépend de la **qualité des données d'entrée**.



HeROsim – Conclusion

- HeROsim est disponible sous licence libre²⁸ ;
- Artefacts pour HeROcache :
 - mis en œuvre dans HeROsim;
 - soumis à CCGrid'24 : trois badges de reproductibilité IEEE²⁹ :
 - *Open Research Objects* (ORO);
 - *Reusable/Research Objects Reviewed* (ROR);
 - ***Results Reproduced*** (ROR-R);
- Soumission au journal *IEEE Internet Computing (Special Issue on Serverless Computing)*.

28. <https://github.com/b-com/HeROsim>

29. <https://www.niso.org/standards-committees/reproducibility-badging>

Conclusion

Conclusion – Résumé des contributions

- Le cloud est un environnement hautement hétérogène :
 - Opportunité : optimisation;
 - Menaces : incertitude, coûts d'opportunité;
- Approche par **prédition des performances** des applications pour l'orchestration serverless :
 - Réduction des démarrages à froid pour des fonctions dont les durées d'initialisation dominent les temps d'exécution;
 - Accélération des phases d'initialisation par la mise en cache des images de fonctions;
 - Minimisation des temps de communication inter-fonctions par la consolidation des applications.
- Estimer pour améliorer :
 - Proposition d'une méthodologie de caractérisation pour les applications serverless;
 - Mise en œuvre d'un simulateur pour évaluer les politiques d'orchestration.

Conclusion – Perspectives

Court terme

- S'affranchir de la phase de caractérisation hors-ligne;
- Formuler le problème de l'ordonnancement pour la dualité prédition / réaction :
 - Prédiction de charge sur séries temporelles;
 - Ajustement des seuils de mise à l'échelle par apprentissage renforcé.

Moyen terme

- Mise à jour du modèle de coût : travaux en cours sur la modélisation de l'impact carbone.

Long terme

- Orchestration dans le *continuum cloud-edge* :
 - Modèle de coût : interférences, pannes, migrations de tâches;
- Stratégies de ralentissement et d'extinction pour un cloud frugal ? Comment éviter l'effet rebond ?

Merci de votre attention.

Bibliographie

Références i

- [1] Mania ABDI et al. « Palette Load Balancing : Locality Hints for Serverless Functions ». *Proceedings of the Eighteenth European Conference on Computer Systems*. EuroSys '23. Rome, Italy : Association for Computing Machinery, mai 2023, p. 365-380. ISBN : 978-1-4503-9487-1. DOI : [10.1145/3552326.3567496](https://doi.org/10.1145/3552326.3567496).
- [2] Michael ARMBRUST et al. « A View of Cloud Computing ». *Commun. ACM* 53.4 (avr. 2010), p. 50-58. DOI : [10.1145/1721654.1721672](https://doi.org/10.1145/1721654.1721672).
- [3] Vivek M. BHASI et al. « Cypress : Input Size-Sensitive Container Provisioning and Request Scheduling for Serverless Platforms ». *Proceedings of the 13th Symposium on Cloud Computing*. SoCC '22. San Francisco, California : Association for Computing Machinery, nov. 2022, p. 257-272. ISBN : 978-1-4503-9414-7. DOI : [10.1145/3542929.3563464](https://doi.org/10.1145/3542929.3563464).
- [4] Frédéric BORDAGE. *Numérique : 40 % du budget GES soutenable d'un européen*. Déc. 2021. URL : <https://www.greenit.fr/2021/12/08/numerique-40-du-budget-ges-soutenable-dun-europeen/>.
- [5] Sebastian BURCKHARDT et al. « Netherite : Efficient Execution of Serverless Workflows ». *Proc. VLDB Endow.* 15.8 (avr. 2022), p. 1591-1604. DOI : [10.14778/3529337.3529344](https://doi.org/10.14778/3529337.3529344).
- [6] Rajkumar BUYYA et Manzur MURSHED. « GridSim : A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing ». *Concurrency and Computation : Practice and Experience* 14.13-15 (nov. 2002), p. 1175-1220. DOI : [10.1002/cpe.710](https://doi.org/10.1002/cpe.710).
- [7] Zhicheng CAI, Qianmu LI et Xiaoping LI. « ElasticSim : A Toolkit for Simulating Workflows with Cloud Resource Runtime Auto-Scaling and Stochastic Task Execution Times ». *Journal of Grid Computing* 15.2 (juin 2017), p. 257-272. DOI : [10.1007/s10723-016-9390-y](https://doi.org/10.1007/s10723-016-9390-y).
- [8] Rodrigo N. CALHEIROS et al. « CloudSim : A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms ». *Software : Practice and Experience* 41.1 (jan. 2011), p. 23-50. DOI : [10.1002/spe.995](https://doi.org/10.1002/spe.995).

Références ii

- [9] Junguk CHO et al. « SLA-Driven ML Inference Framework for Clouds With Heterogeneous Accelerators ». *Proceedings of Machine Learning and Systems*. T. 4. 2022, p. 20-32. URL : https:////proceedings.mlsys.org/paper_files/paper/2022/hash/bcf9bef61a534d0ce4a3c55f09dfcc29-Abstract.html.
- [10] Rahul GHOSH et Vijay K. NAIK. « Biting Off Safely More Than You Can Chew : Predictive Analytics for Resource Over-Commit in IaaS Cloud ». *2012 IEEE Fifth International Conference on Cloud Computing*. IEEE Cloud '12. Honolulu, HI, USA : IEEE, juin 2012, p. 25-32. DOI : [10.1109/CLOUD.2012.131](https://doi.org/10.1109/CLOUD.2012.131).
- [11] Arpan GUJARATI et al. « Swayam : Distributed Autoscaling to Meet SLAs of Machine Learning Inference Services with Resource Efficiency ». *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference*. Las Vegas Nevada : ACM, déc. 2017, p. 109-120. ISBN : 978-1-4503-4720-4. DOI : [10.1145/3135974.3135993](https://doi.org/10.1145/3135974.3135993).
- [12] Edson HORTA et al. « Xar-Trek : Run-Time Execution Migration among FPGAs and Heterogeneous-ISA CPUs ». *Proceedings of the 22nd International Middleware Conference*. Middleware '21. Québec city Canada : ACM, déc. 2021, p. 104-118. DOI : [10.1145/3464298.3493388](https://doi.org/10.1145/3464298.3493388).
- [13] INTERNATIONAL DATA CORPORATION. *Worldwide Spending on Public Cloud Services Is Forecast to Reach \$1.35 Trillion in 2027, According to New IDC Spending Guide*. URL : <https://www.idc.com/getdoc.jsp?containerId=prUS51179523>.
- [14] INTERNATIONAL ENERGY AGENCY. *Electricity 2024 - Analysis and Forecast to 2026*. Rapp. tech. 2024. URL : <https://www.iea.org/reports/electricity-2024>.
- [15] Hongseok JEON et al. « A CloudSim-Extension for Simulating Distributed Functions-as-a-Service ». *2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*. Gold Coast, Australia : IEEE, déc. 2019, p. 386-391. ISBN : 978-1-72812-616-6. DOI : [10.1109/PDCAT46702.2019.00076](https://doi.org/10.1109/PDCAT46702.2019.00076).

Références iii

- [16] Vincent LANNURIEN et al. « HeROcache : Storage-Aware Scheduling in Heterogeneous Serverless Edge – The Case of IDS ». *2024 IEEE/ACM 24th International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. 2024, p. 587-597. DOI : [10.1109/CCGrid59990.2024.00071](https://doi.org/10.1109/CCGrid59990.2024.00071).
- [17] Vincent LANNURIEN et al. « HeROfake : Heterogeneous Resources Orchestration in a Serverless Cloud – An Application to Deepfake Detection ». *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. 2023, p. 154-165. DOI : [10.1109/CCGrid57682.2023.00024](https://doi.org/10.1109/CCGrid57682.2023.00024).
- [18] Vincent LANNURIEN et al. « Serverless Cloud Computing : State of the Art and Challenges ». *Serverless Computing : Principles and Paradigms*. Sous la dir. de Rajalakshmi KRISHNAMURTHI et al. Cham : Springer International Publishing, 2023, p. 275-316. ISBN : 978-3-031-26633-1. DOI : [10.1007/978-3-031-26633-1_11](https://doi.org/10.1007/978-3-031-26633-1_11).
- [19] Zijun Li et al. « FaaSFlow : Enable Efficient Workflow Execution for Function-as-a-Service ». *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. ASPLOS '22*. Lausanne, Switzerland : Association for Computing Machinery, 2022, p. 782-796. ISBN : 9781450392051. DOI : [10.1145/3503222.3507717](https://doi.org/10.1145/3503222.3507717).
- [20] Wei LING et al. « Pigeon : A Dynamic and Efficient Serverless and FaaS Framework for Private Cloud ». *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. Las Vegas, NV, USA : IEEE, déc. 2019, p. 1416-1421. ISBN : 978-1-72815-584-5. DOI : [10.1109/CSCI49370.2019.00265](https://doi.org/10.1109/CSCI49370.2019.00265).
- [21] Nima MAHMOUDI et Hamzeh KHAZAEI. « SimFaaS : A Performance Simulator for Serverless Computing Platforms : » *Proceedings of the 11th International Conference on Cloud Computing and Services Science. SCITEPRESS - Science and Technology Publications*, 2021, p. 23-33. ISBN : 978-989-758-510-4. DOI : [10.5220/0010376500230033](https://doi.org/10.5220/0010376500230033).

Références iv

- [22] Redowan MAHMUD et al. « iFogSim2 : An Extended iFogSim Simulator for Mobility, Clustering, and Microservice Management in Edge and Fog Computing Environments ». *Journal of Systems and Software* 190 (août 2022), p. 111351. DOI : [10.1016/j.jss.2022.111351](https://doi.org/10.1016/j.jss.2022.111351).
- [23] Anupama MAMPAGE et Rajkumar BUYYA. « CloudSimSC : A Toolkit for Modeling and Simulation of Serverless Computing Environments ». *2023 IEEE International Conference on High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. Melbourne, Australia : IEEE, déc. 2023, p. 550-557. ISBN : 9798350330014. DOI : [10.1109/HPCC-DSS-SmartCity-DependSys60770.2023.00081](https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys60770.2023.00081).
- [24] Anupama MAMPAGE, Shanika KARUNASEKERA et Rajkumar BUYYA. « Deadline-Aware Dynamic Resource Management in Serverless Computing Environments ». *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. Melbourne, Australia : IEEE, mai 2021, p. 483-492. ISBN : 978-1-72819-586-5. DOI : [10.1109/CCGrid51090.2021.00058](https://doi.org/10.1109/CCGrid51090.2021.00058).
- [25] Eric MASANET et al. « Recalibrating Global Data Center Energy-Use Estimates ». *Science* 367.6481 (fév. 2020), p. 984-986. DOI : [10.1126/science.aba3758](https://doi.org/10.1126/science.aba3758).
- [26] Fabian MASTENBROEK et al. « OpenDC 2.0 : Convenient Modeling and Simulation of Emerging Technologies in Cloud Datacenters ». *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. Melbourne, Australia : IEEE, mai 2021, p. 455-464. ISBN : 978-1-72819-586-5. DOI : [10.1109/CCGrid51090.2021.00055](https://doi.org/10.1109/CCGrid51090.2021.00055).
- [27] John McCARTHY. « Management and the Computer of the Future ». Sous la dir. de Martin GREENBERGER. M.I.T. Press et Wiley, New York, mars 1962, p. 220-236. ISBN : 9780262070041.
- [28] Peter MELL et Timothy GRANCE. « The NIST Definition of Cloud Computing ». *National Institute of Standards and Technology Special Publication 800-145* (sept. 2011). DOI : [10.6028/NIST.SP.800-145](https://doi.org/10.6028/NIST.SP.800-145).

Références v

- [29] Ingo MÜLLER, Renato MARROQUÍN et Gustavo ALONSO. « Lambada : Interactive Data Analytics on Cold Data Using Serverless Cloud Infrastructure ». *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. SIGMOD '20. Portland OR USA : ACM, juin 2020, p. 115-130. ISBN : 978-1-4503-6735-6. DOI : [10.1145/3318464.3389758](https://doi.org/10.1145/3318464.3389758).
- [30] Seyed Hossein NIKOUNIA et Siamak MOHAMMADI. « Hypervisor and Neighbors' Noise : Performance Degradation in Virtualized Environments ». *IEEE Transactions on Services Computing* (2015), p. 1-1. ISSN : 1939-1374. DOI : [10.1109/TSC.2015.2464811](https://doi.org/10.1109/TSC.2015.2464811). (Visité le 15/11/2024).
- [31] Alberto NÚÑEZ et al. « iCanCloud : A Flexible and Scalable Cloud Infrastructure Simulator ». *Journal of Grid Computing* 10.1 (mars 2012), p. 185-209. DOI : [10.1007/s10723-012-9208-5](https://doi.org/10.1007/s10723-012-9208-5).
- [32] Massoud PEDRAM. « Energy-Efficient Datacenters ». *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 31.10 (oct. 2012), p. 1465-1484. DOI : [10.1109/TCAD.2012.2212898](https://doi.org/10.1109/TCAD.2012.2212898).
- [33] Johann SCHLEIER-SMITH et al. « What Serverless Computing is and Should Become : The next Phase of Cloud Computing ». *Commun. ACM* 64.5 (avr. 2021), p. 76-84. DOI : [10.1145/3406011](https://doi.org/10.1145/3406011).
- [34] Mohammad SHAHRAD et al. « Serverless in the Wild : Characterizing and Optimizing the Serverless Workload at a Large Cloud Provider ». *Proceedings of the 2020 USENIX Conference on Usenix Annual Technical Conference*. USENIX ATC'20. USA : USENIX Association, 2020, p. 14.
- [35] Arman SHEHABI et al. *United States Data Center Energy Usage Report*. Rapp. tech. LBNL-1005775, 1372902. Juin 2016, LBNL-1005775, 1372902. DOI : [10.2172/1372902](https://doi.org/10.2172/1372902).
- [36] Arjun SINGHVI et al. « Atoll : A Scalable Low-Latency Serverless Platform ». *Proceedings of the ACM Symposium on Cloud Computing*. Seattle WA USA : ACM, nov. 2021, p. 138-152. ISBN : 978-1-4503-8638-8. DOI : [10.1145/3472883.3486981](https://doi.org/10.1145/3472883.3486981).

Références vi

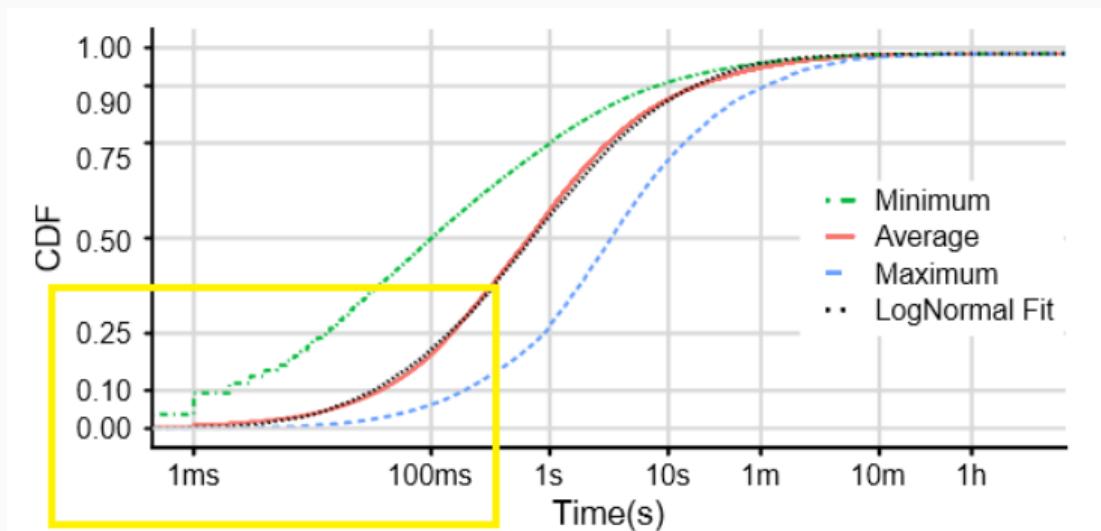
- [37] Christopher Peter SMITH et al. « FaDO : FaaS Functions and Data Orchestrator for Multiple Serverless Edge-Cloud Clusters ». *2022 IEEE 6th International Conference on Fog and Edge Computing (ICFEC)*. Messina, Italy : IEEE, mai 2022, p. 17-25. ISBN : 978-1-66549-524-0. DOI : [10.1109/ICFEC54809.2022.00010](https://doi.org/10.1109/ICFEC54809.2022.00010).
- [38] Amoghavarsha SURESH et al. « ENSURE : Efficient Scheduling and Autonomous Resource Management in Serverless Environments ». *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*. ACSOS '20. Washington, DC, USA : IEEE, août 2020, p. 1-10. ISBN : 978-1-72817-277-4. DOI : [10.1109/ACSOS49614.2020.00020](https://doi.org/10.1109/ACSOS49614.2020.00020).
- [39] THE KNATIVE AUTHORS. *Knative Autoscaling*. 2022. URL : <https://github.com/knative/serving/tree/main/docs/scaling>.
- [40] Arunchandar VASAN et al. « Worth Their Watts ? - An Empirical Study of Datacenter Servers ». *HPCA - 16 2010 The Sixteenth International Symposium on High-Performance Computer Architecture*. HPCA '16. Bangalore : IEEE, jan. 2010, p. 1-10. ISBN : 978-1-4244-5658-1. DOI : [10.1109/HPCA.2010.5463056](https://doi.org/10.1109/HPCA.2010.5463056).
- [41] Liang WANG et al. « Peeking Behind the Curtains of Serverless Platforms ». *2018 USENIX Annual Technical Conference (USENIX ATC 18)*. USENIX ATC'18. Boston, MA : USENIX Association, juill. 2018, p. 133-146. ISBN : ISBN 978-1-939133-01-4. URL : <https://www.usenix.org/conference/atc18/presentation/wang-liang> (visité le 18/09/2024).
- [42] Michael WAWRZONIAK et al. « Boxer : Data Analytics on Network-enabled Serverless Platforms ». *CIDR 2021*. CIDR '21. www.cidrdb.org, jan. 2021. DOI : [10.3929/ETHZ-B-000456492](https://doi.org/10.3929/ETHZ-B-000456492).
- [43] Bhathiya WICKREMASINGHE, Rodrigo N. CALHEIROS et Rajkumar BUYYA. « CloudAnalyst : A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications ». *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. Perth, Australia : IEEE, 2010, p. 446-452. ISBN : 978-1-4244-6695-5. DOI : [10.1109/AINA.2010.32](https://doi.org/10.1109/AINA.2010.32).

Références vii

- [44] Bill WILDER. *Cloud Architecture Patterns : Using Microsoft Azure*. O'Reilly Media, Inc., 2012.
- [45] Yanan YANG et al. « INFless : A Native Serverless System for Low-Latency, High-Throughput Inference ». *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. Lausanne Switzerland : ACM, fév. 2022, p. 768-781. ISBN : 978-1-4503-9205-1. DOI : [10.1145/3503222.3507709](https://doi.org/10.1145/3503222.3507709).
- [46] Chengliang ZHANG et al. « MArk : Exploiting Cloud Services for Cost-Effective, SLO-Aware Machine Learning Inference Serving ». *2019 USENIX Annual Technical Conference (USENIX ATC 19)*. USENIX ATC'19. Renton, WA : USENIX Association, juill. 2019, p. 1049-1062. ISBN : 978-1-939133-03-8. URL :
<https://www.usenix.org/conference/atc19/presentation/zhang-chengliang>.
- [47] Lu ZHANG et al. « FIRST : Exploiting the Multi-Dimensional Attributes of Functions for Power-Aware Serverless Computing ». *2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. St. Petersburg, FL, USA : IEEE, mai 2023, p. 864-874. ISBN : 9798350337662. DOI : [10.1109/IPDPS54959.2023.00091](https://doi.org/10.1109/IPDPS54959.2023.00091).

Annexes

Contributions – Cas d'usage et généricité applicative

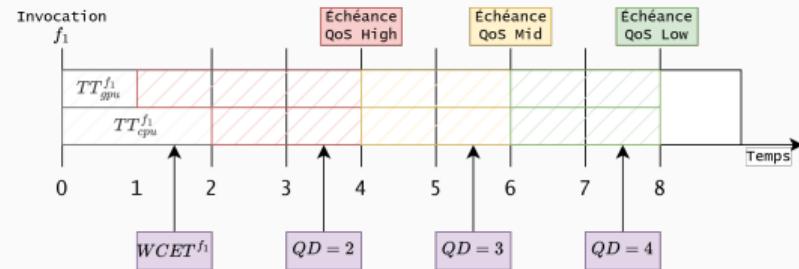


29. [34] Mohammad SHAHRAD et al. « Serverless in the Wild : Characterizing and Optimizing the Serverless Workload at a Large Cloud Provider ». USENIX ATC'20. 2020

HeROfake – Niveaux de QoS

Facteurs de dégradation du temps de réponse (jusqu'à 16x) dans le cloud³⁰ :

- Traversée des couches réseau;
- Partage des ressources (*noisy neighbour*);
- Surcharge de la virtualisation.



$$QP_{f_{N,P}} = \begin{cases} 1 & \text{si } TT_{f_{N,P}} \cdot QD_{f_{N,P}} > WCET_f \\ 0 & \text{si } TT_{f_{N,P}} \cdot QD_{f_{N,P}} \leq WCET_f \end{cases}$$

30. [30] Seyed Hossein NIKOUNIA et Siamak MOHAMMADI. « Hypervisor and Neighbors' Noise : Performance Degradation in Virtualized Environments ». *IEEE Transactions on Services Computing* (2015). ISSN : 1939-1374

HeROcache – Dépendances entre fonctions

$$\forall (N, P) \in R_f, \text{schedCost}_{f_{i_{N,P}}} =$$

$$k_{QP} \cdot QP_{f_{N,P}}$$

$$+ k_{EC} \cdot EC_{f_{N,P}}$$

$$+ k_{TC} \cdot TC_{f_{N,P}}$$

Minimisation des pénalités dans les chaînes de fonctions :

$$QP_a = \sum_{i=0}^{F_a} TT_{f_{i_{N,P}}} > \sum_{i=0}^{F_a} WCET_{f_i} \cdot QD_a$$

Calcul de la consolidation pour maximiser l'utilisation des plateformes :

$$PU_{f_{N,P}} = \frac{Q_{N,P}}{\text{threshold}_{f,P}}$$

Minimisation :

$$TC_{f_{N,P}} = \exp(PU_{f_{N,P}})$$

HeROcache – Cache de fonctions

Pour une application a , calcul de la proportion d'images de ses fonctions déjà en cache sur le nœud :

$$\forall f \in a, CF_a^{f_{i_N, P}} = \frac{\sum_{i=0}^{F_a} isCached(f_i, N, P)}{F_a}$$

Pré-chargement des fonctions du DAG de l'application pour améliorer la proportion de fonctions en cache :

$$\forall N, \forall P \in N, CP_{a_N} = \frac{A}{\sum_{i=0}^{F_a} CF_a^{f_{i_N, P}}}$$

HeROsim – Modéliser des applications

```
"hello-world": {
    "name": "hello-world",
    "dag": {
        "hello": [],
        "world": ["hello"]
    }
}

"hello": {
    "name": "hello",
    "platforms": [
        "exampleCpu"
    ],
    "memoryRequirements": {
        "exampleCpu": 0.0015
    },
    "coldStartDuration": {
        "exampleCpu": 0.002
    },
    "executionTime": {
        "exampleCpu": 8.5e-3
    },
    "energy": {
        "exampleCpu": 5.5e-9
    },
    "imageSize": {
        "exampleCpu": 0.003
    },
    "stateSize": {
        "hello-world": {
            "input": 0,
            "output": 6
        }
    }
}

"world": {
    "name": "world",
    "platforms": [
        "exampleCpu"
    ],
    "memoryRequirements": {
        "exampleCpu": 0.0030
    },
    "coldStartDuration": {
        "exampleCpu": 0.004
    },
    "executionTime": {
        "exampleCpu": 8.5e-3
    },
    "energy": {
        "exampleCpu": 5.5e-9
    },
    "imageSize": {
        "exampleCpu": 0.003
    },
    "stateSize": {
        "hello-world": {
            "input": 6,
            "output": 12
        }
    }
}
```