

Adrien Faure

Ingénieur informatique, PhD

+33 6 73 18 80 27 ✉ adrien.faure@protonmail.com 🌐 adfaure 📍 Grenoble, France



SYNTHÈSE DES EXPERIENCES

Lead développeur	Université Grenoble Alpes	2021 – 2023
Architecte et développeur fullstack	Aktid	2022 – 2023 (3 mois)
Doctorant – Ingénieur	Atos – Université Grenoble Alpes	2017 – 2020 (3 ans)
Enseignant	Université Grenoble Alpes	2017 – 2023 (~120h)
Stagiaire R&D	Atos	2016 (6 mois)
Stagiaire développement Web	Wizacha	2014 (3 mois)

COMPÉTENCES TECHNIQUES

Langages	Python, Rust, C/C++, SQL, Bash, Perl, R, Nix, Java, Go, Javascript, Typescript, Dart, PHP
Framework	FastApi, React, Angular, Spark, Spring, Flutter
Bibliothèques	pandas, numpy, tidyverse, zmq, asyncio, protobuff, thrift, MPI
Environnement	Linux, AWS, Kubernetes, HPC, Cloud, Big Data, NixOS, debian

COMPÉTENCES FONCTIONNELLES

Outils	Git, Docker, Latex, Nginx, Traefik
SGBD	Postgres, Redis, sqlserver, sqlite, mongodb
Méthodologie	Scrum, Pomodoro, Agile
Documentation	Documentation, Tutoriaux, Talks, Cours, Publications scientifique, CIR, Diagrammes explicatifs
Conceptes	Architecture informatique, Modélisation de données (UML), Systèmes distribués, Scalabilité, Parallélisme, Asynchronisme, Data pipeline, DevOps, SRE, Monitoring, Intégration Continue (CI), Déploiement continue, Tests (unitaire, intégration), Design patterns, Méthode scientifique
Compétences générales	Autonome, Sociale, Curieux, Esprit critique, Ouvert à la critique
Langues	Français, Anglais

FORMATION

Doctorat en informatique	2020	Grenoble
Master informatique	2016	Lyon et Grenoble
Licence informatique	2014	Lyon
BAC S	2009	Bourgoin-Jallieu

i Contexte

Deux contrats CDD au sein du laboratoire d'informatique de Grenoble à l'Université Grenoble Alpes (UGA). J'ai travaillé au pour un projet européen avec des acteurs privés et publiques en Europe. Mon rôle principal a majoritairement été le développement d'un logiciel open source **OAR3**.

OAR3 est un logiciel de gestion de plateforme HPC. C'est un logiciel complexe et distribué. Impliquant notamment:

- de la gestion de ressources à l'échelle (ordonnancement de plusieurs milliers de programmes),
- des manipulations systèmes fines pour l'exécution de programmes parallèles (via les cgroup et la namespace Linux),
- ainsi qu'un interface utilisateur en cli et restapi

Objectif

- **L'objectif principal est de rendre OAR3 production ready**
- Répondre aux besoins du projet européen
 - Contrôle et améliorations des politiques d'ordonnements pour la gestion d'énergie
 - Préparer et présenter des POCs qui fonctionne à l'échelle (rendus du projet européen)
- Participer aux activités du laboratoire et de l'équipe de recherche
 - Encadrement de stagiaires
 - Aide aux expériences de recherches
 - Rédaction de papiers scientifiques

Travail réalisé

- **Développeur principal sur OAR3 en python3**
 - Développement de nouvelles fonctionnalités
 - Amélioration de la qualité du code
 - Refactoring de modules importants (eg. ré-implémentation de l'api REST avec Fastapi)
 - Commentaires sur les parties obscures
 - Écriture de tests
 - Garant de l'intégration continue avec github actions et maintiens des paquets nix et debian
 - Support utilisateurs :
 - Pour les administrateurs lors de la mise en place d'un nouveau cluster
 - Pour les utilisateurs via l'ajout de nouvelles fonctionnalités ou le debugage
 - Collaboration internationale (Barcelone, Munich, Bologne) pour l'intégration d'outils
 - Développement d'outils satellite à OAR3: (monitoring, benchmarking, déploiement etc)
- **Médiation, documentation:**
 - Documentations: user guides, getting started, diagrams and schematics, reference API
 - Écriture de tutoriaux pour des ingénieurs et chercheurs:

- Sur OAR
- Sur Nix: <https://nix-tutorial.gitlabpages.inria.fr/nix-tutorial>

Résultats:

- Le projet OAR3 est passé en production (beta test au moment de mon départ) sur un nouveau cluster
- Démonstration de fonctionnalités via une vidéo enregistrant le management d'une plateforme avec OAR3 (rendu pour un projet européen)
 - Intégration de plusieurs outils participant au projet REGALE: monitoring tool, deep learning application
- Contribution scientifiques

Languages	Python, Perl, Nix, Rust, R, C/C++
Tools	FastApi, Linux, github actions, pandas, numpy, scikit-learn, docker, docker-compose, PostgreSQL, asyncio, MPI
Conceptes	Système distribué, Management d'infrastructure, Ordonnancement, Recherche, Monitoring

Aktid

Architecte et développeur fullstack

Freelance

2022 – 2023 (3 mois)

i Contexte

Aktid est une entreprise qui conçoit et réalise des centres de tri pour la valorisation des déchets. Dans le cadre de travaux de R&D d'intelligence artificielle de maintenance prédictive, Aktid avait besoin d'un architecte et développeur pour créer de zéro une dashboard de maintenance prédictive.

Travail réalisé

- **Architecture complète d'une nouvelle plateforme de maintenance prédictive**
 - Reprise des travaux de recherche pour concevoir une implémentation sur site
- **Conception de la partie *backend*:**
 - Service de prédiction en ligne asynchrone (asyncio)
 - Conception du schéma de données pour le stockage des prédictions
 - RestApi en python pour la partie *frontend*
- **Conception de la partie *frontend*:**
 - Dashboard en ReactJs
 - Affichage des prédictions en ligne avec ReCharts
- **Déploiement en production:**
 - Déploiement en production dans une usine pour une version beta
 - Avec la technologie docker-compose sur un serveur debian
- **Documentation et formation:**

- Écriture des documentations nécessaire pour la reprise par d'autres équipes
- Présentation et formation sur site de l'outil

Résultats: Déploiement de la version 1 du tableau de bords de maintenance prédictive pour les usines de centre de tris d'Aktid. L'objectif était de faire cette réalisation en un moi pour correspondre à mes disponibilités. La suite de développement a été reprise par un développeur sous mes conseils.

Langages	Python, Typescript, SQL
Tools	FastApi, React, Linux, pandas, numpy, scikit-learn, docker, docker-compose, Postgres, asyncio
Conceptes	machine learning, maintenance prédictive

Atos – Université Grenoble Alpes

Doctorant – Ingénieur

Thèse CIFRE

2017 – 2020 (3 ans)

i Contexte

Doctorat en entreprise, entre le laboratoire d'informatique de Grenoble (LIG) et Atos. Mes travaux de recherches se situent dans la branche HPC (High Performance Computing) de Atos (anciennement BULL), plus spécifiquement, sur les ordonnanceurs de tâches pour les clusters de type supercalculateur.

Sujet(s) de Thèse: Le sujet d'étude de mon doctorat sont les ordonnanceurs de plateforme de type HPC. Ce sont des logiciels complexes et distribués, au centre de l'utilisation des plateformes. Leur rôle central, en font à la fois un sujet d'étude important (un ordonnanceur inefficace impacte toute la plateforme – ce qui est onéreux) et difficile car il est quasiment impossible de remplacer ces logiciels pour conduire des études (sans en impacter le fonctionnement).

Mes contributions principales sont:

- Méthodes et outils pour l'évaluation de performance des ordonnanceurs HPC.
- Algorithmes d'ordonnancement pour les ordonnanceurs HPC avec redirection de tâches.
- Utilisation de Nix pour faire des expériences informatiques reproductibles.

Travail réalisé:

- Développement d'outils pour l'émulation d'application distribué
 - Moteur d'exécution distribué et parallèle: orchestration d'expérience, parallélisation avec spark
 - Conception et développement Algorithme d'ordonnancement en rust
 - Programmation d'application parallèle en MPI
- Analyse de données en python et R, avec pandas, jupyter, tidyverse et rmarkdown.
- Rédaction de papiers scientifique en anglais
- Encadrement de stagiaires
- Cours en vacation

Résultats:

- Le titre de docteur de l'Université Grenoble Alpes

- La thèse est consultable en ligne – <https://theses.fr/2020GRALM056>.

Langages	Rust, Python, R, C/C++, Scala, Go, Bash, Nix
Tools	Linux, Spark, Nixos, Tidyverse
Conceptes	Programmation parallèle, Système distribué, Ordonnancement, Éprit critique, Rédaction, Esprit de synthèse, Méthode Scrum

Université Grenoble Alpes

Enseignant

Vacataire

2017 – 2023 (~120h)

i Contexte

Le travail en laboratoire est une place de choix pour trouver des missions d'enseignement à la faculté. De 2016 à 2023, pendant mon séjour à l'UGA j'ai effectué plusieurs vacations.

Liste des postes occupés:

- Responsable de modules IUT2 – Failles logicielles (1 semestre)
- Responsable de modules IUT2 – Qualité logicielle (1 semestre)
- Chargé de TP master 1 – Devops (1 semestre)
- Chargé de TD L3 miage – Statistique et visualisation (1 semestre)
- Chargé de TD L3 info – Algorithmes avancés

Langages et outils	Java, Python, R, Docker, Docker-compose, Git
Conceptes	Pédagogie, Autonomie, DevOps

Atos

Stagiaire R&D

Stage de Master 2

2016 (6 mois)

i Contexte

Stage de fin d'étude à Atos. Stage au sein du pôle R&D, au coeur de l'activité High Performance Computing de Atos Echrirolles.

Objectif: Construire un émulateur de système distribué en remplaçant, de façon transparente, la libc par une bibliothèque de simulation. L'émulateur a été nommé simunix.

Travail réalisé:

Développement d'un simulateur de la librairie C : Le simulateur exécute une application distribuée (type gestionnaire de ressources) dans une boîte isolante. Le logiciel distribué s'exécute, sans s'en rendre compte, sur une seule machine au lieu d'être déployé à l'échelle. Le projet marche via un ré-implé-

mentation de la libc, et des mécanisme de réécriture de fichiers binaires. Voici une liste des fonctions que j'ai dû réécrire :

- Les fonctions système bloquantes : Mutex, CondVar.
- Les fonctions de gestion de temps : sleep, gettimeofday.
- Les fonctions de gestion de processus : fork, execv etc.
- Les fonctions de gestion du réseau : API socket (send/recv, connect/accept).

Résultat:

- Émulation d'un système distribué sur plusieurs noeuds (30 à l'issue du stage) sur une seule machine.
- Une présentation de simunix au Slurm user days (https://slurm.schedmd.com/SLUG16/slugin16_simunix.pdf).

Langages	C++, C, Python
Tools	Linux, libc , Thrift, Protobuf, ZMQ, Simgrid, Slurm
Conceptes	Programmation système linux (socket, threads, mutex etc), Émulation, Ordonancement, Méthode Scrum

Wizacha

Stagiaire développement Web

Stage Licence 3

2014 (3 mois)

i Contexte

Stage de fin de Licence au sein de la start-up Wizacha. En 2014, wizacha est une startup dans le domaine des market-places. Le concept est de permettre à des vendeurs de bénéficier de la plateforme pour vendre leur produits.

Objectif: Prendre part aux activités de développement au sein d'une équipe utilisant la méthode Scrum.

Travail réalisé:

- Développement en Javascript du panier utilisateur
 - Partie back-end en PHP et FrontEnd en javascript
 - Écriture des tests unitaires associés
- Participation aux sprints via la méthodologie Agile
- Utilisation de mercurial pour la gestion des versions

Résultat:

- Panier utilisateur en production
- Apprentissage des fondamentaux de la méthode scrum

Langages	Javascript, PHP
Tools	Linux, Mercurial, JQuery

Conceptes

Programmation Web, Méthode Scrum