



Hack-A-Thon VO France

Ecoconception logicielle ?

Laurent Bourgès - OSUG

Page ANF EcoInfo 2019:

<https://ecoinfo.cnrs.fr/2019/04/11/formation-impact-environnemental-du-numerique-comprendre-et-agir/>

Nouvelle ANF à venir en 2020 !

Présentations:

Les enjeux environnementaux globaux — Pierre-Yves Longaretti

Focus sur les consommations en phase d'usage — Laurent Lefèvre

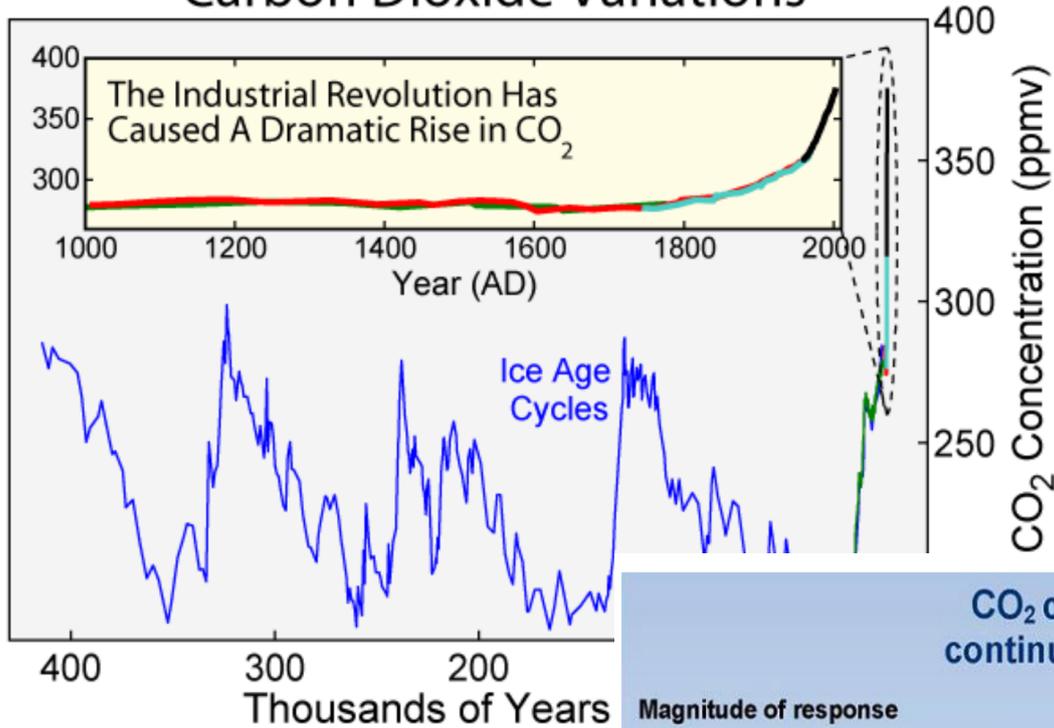
Pourquoi n'agit-on pas (suffisamment) ? — Peter Sturm

Atelier gestion éco responsable des données

Dépot: https://gricad-gitlab.univ-grenoble-alpes.fr/bonamyc/ecoinfo_anf



Carbon Dioxide Variations



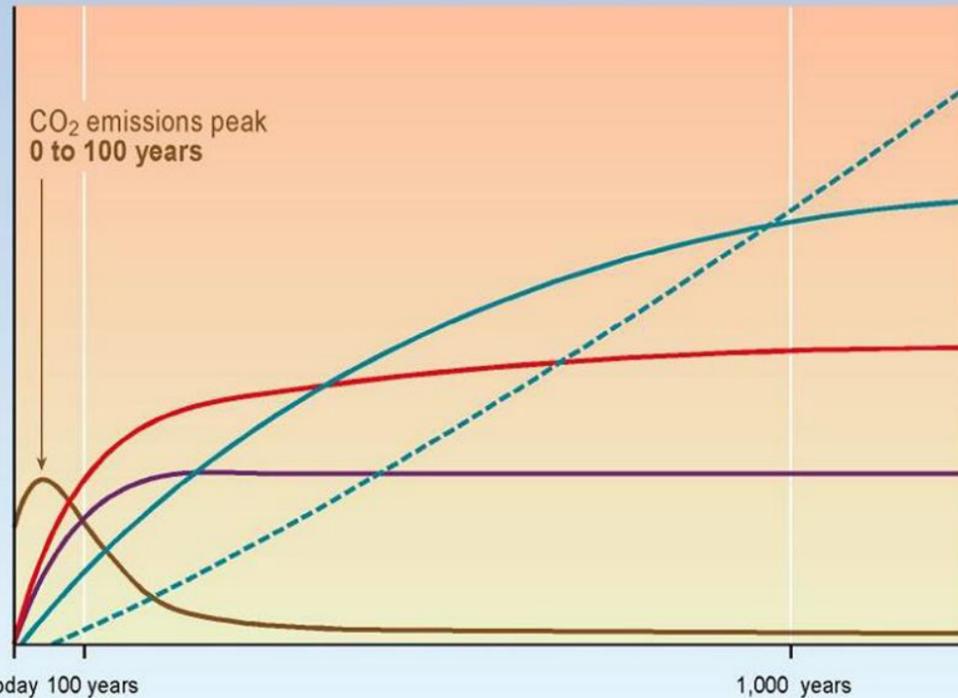
Pierre-Yves Longaretti



Pourquoi s'inquiéter ?

CO₂ concentration, temperature, and sea level continue to rise long after emissions are reduced

Magnitude of response



Time taken to reach equilibrium

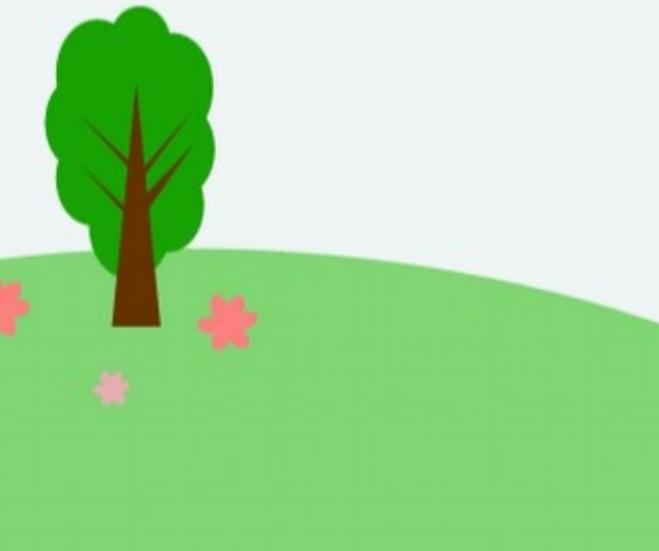
Sea-level rise due to ice melting: **several millennia**

Sea-level rise due to thermal expansion: **centuries to millennia**

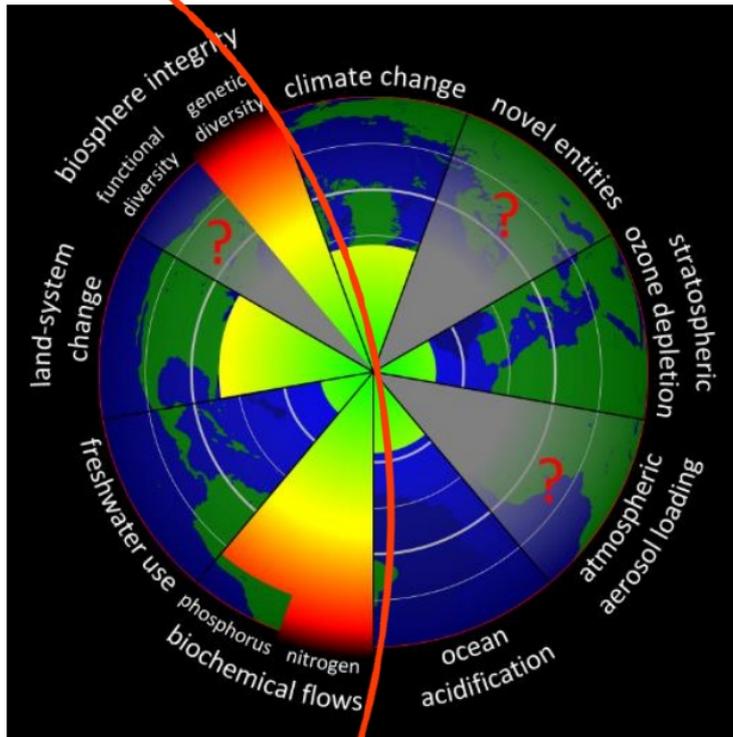
Temperature stabilization: **a few centuries**

CO₂ stabilization: **100 to 300 years**

CO₂ emissions

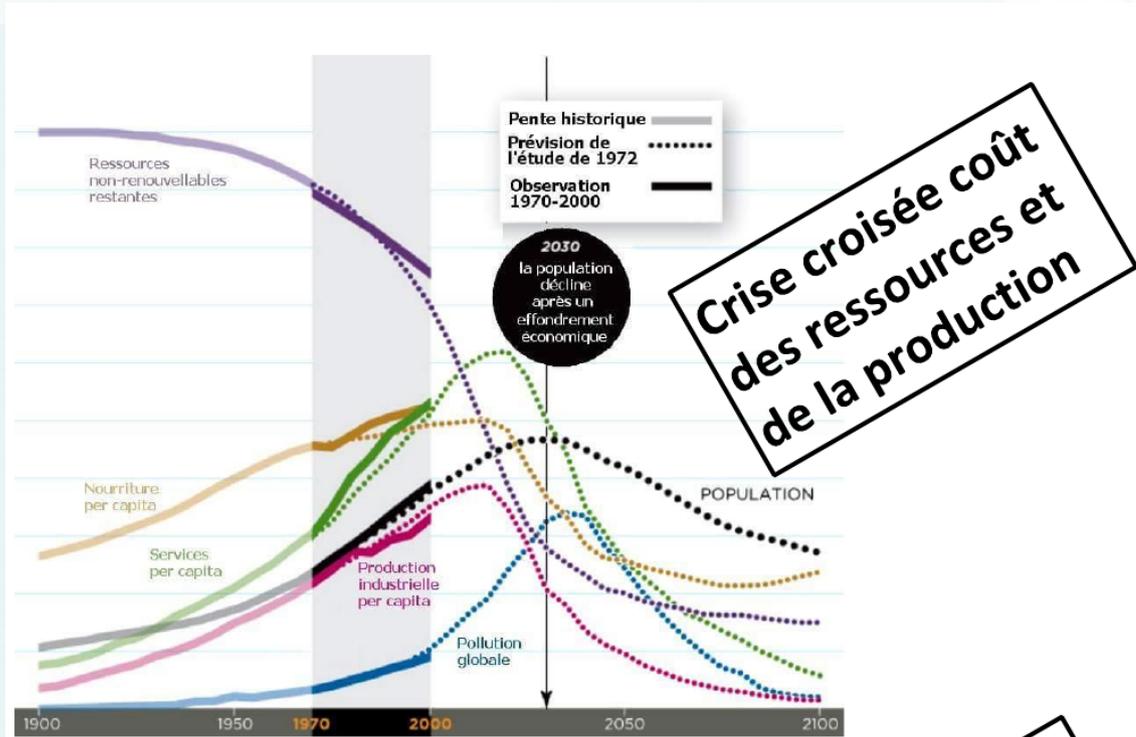


Limites planétaires

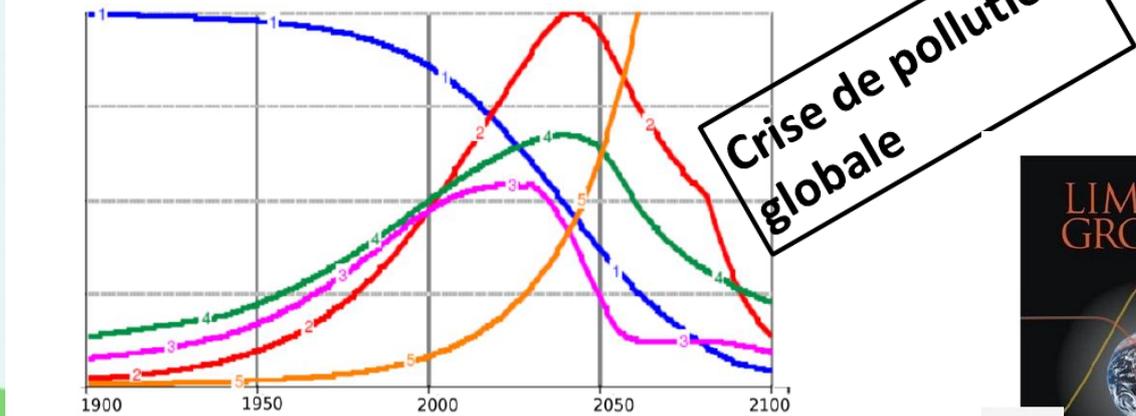


agriculture(s)

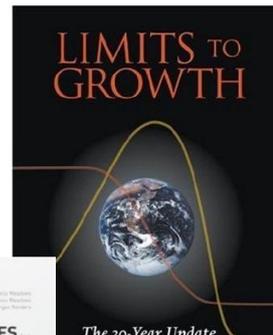
pollutions



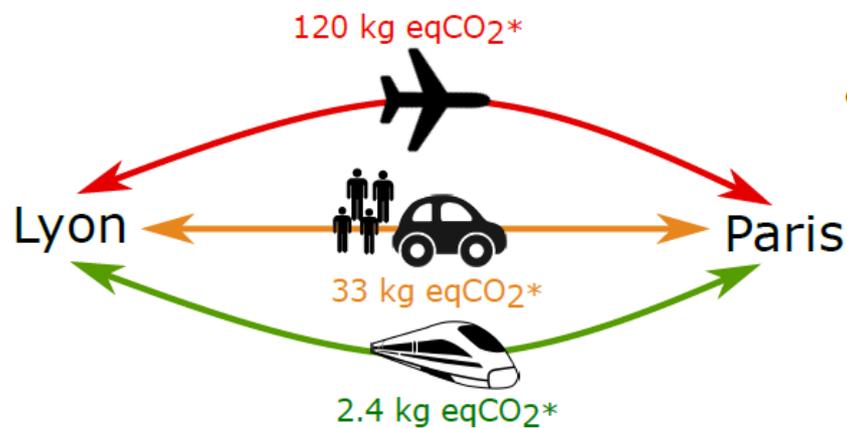
Crise croisée coût des ressources et de la production



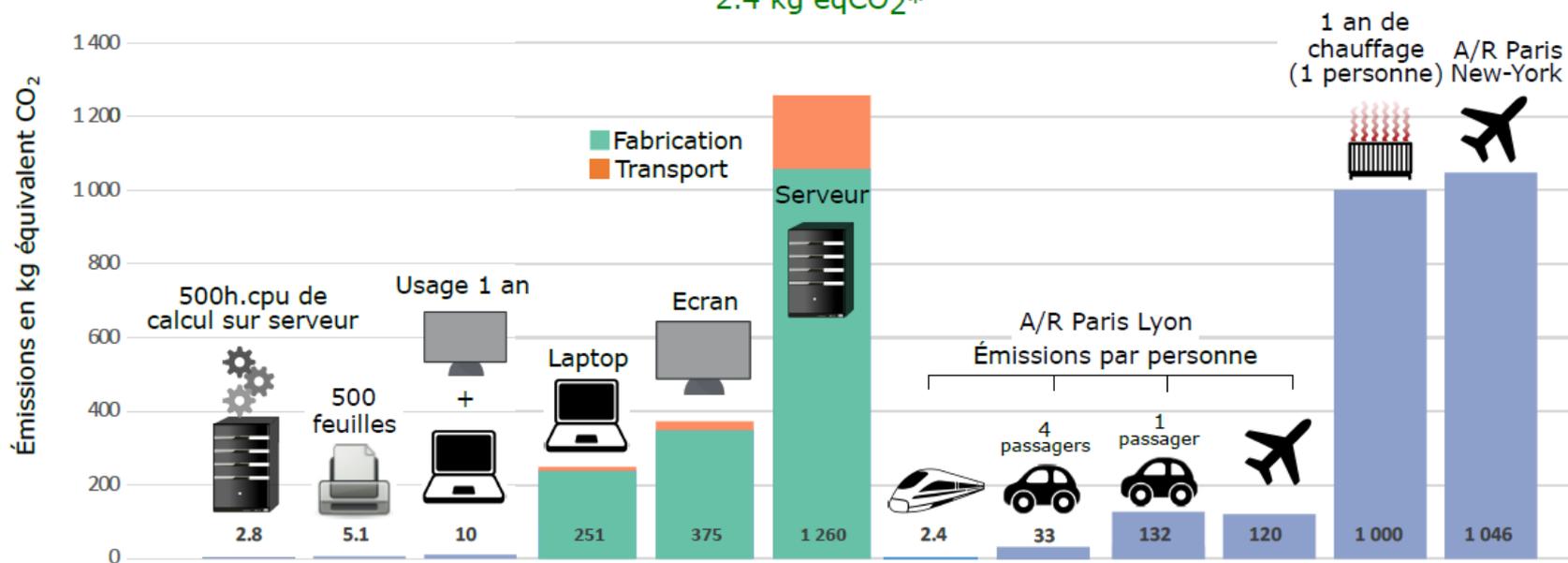
Crise de pollution globale



Comparatif d'émissions équivalent CO₂



* Émissions pour un A/R en kg équivalent CO₂/voyageur
Sources dans le mail associé

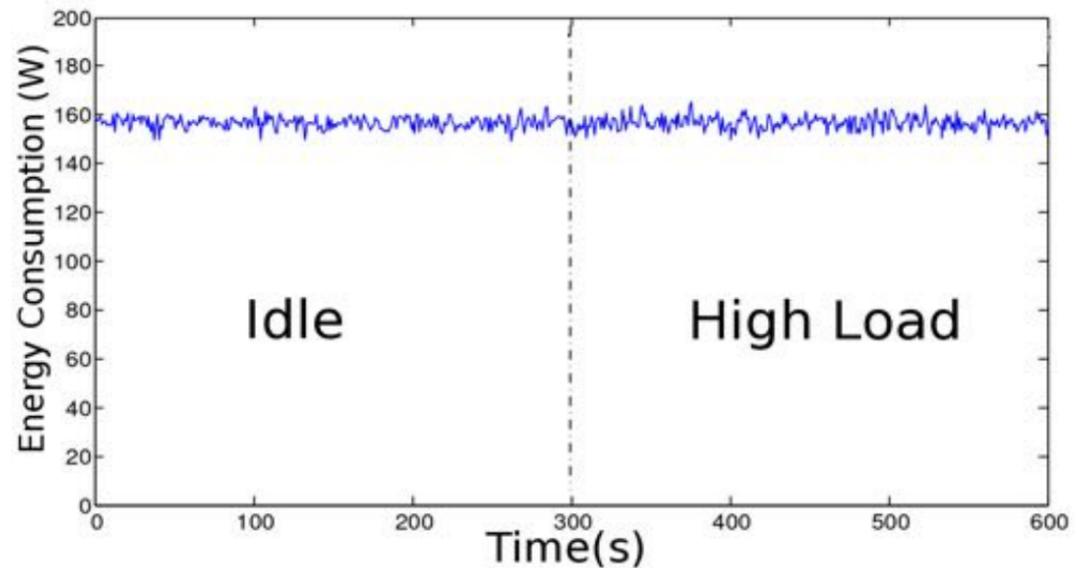
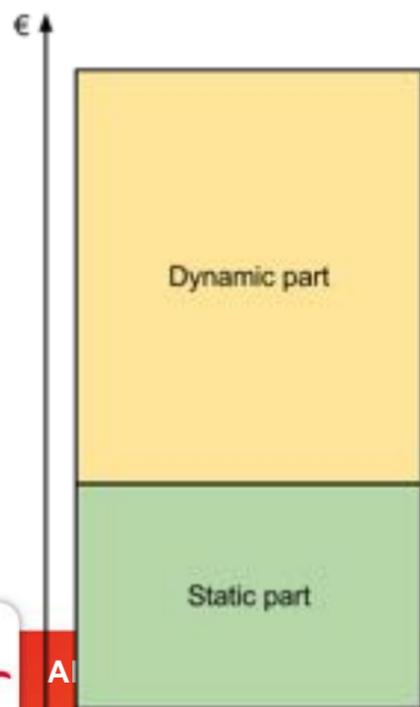
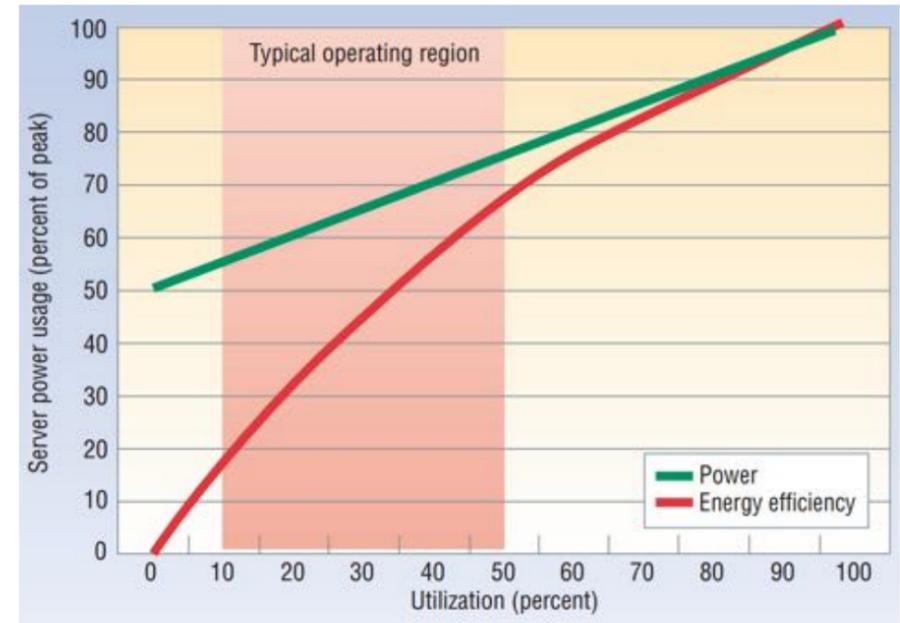


Pas/Peu de Proportionalité Energétique

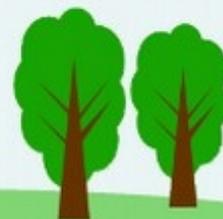
Luiz André Barroso and Urs Hölzle, « The case for Energy-Proportional Computing », IEEE Computer, 2007

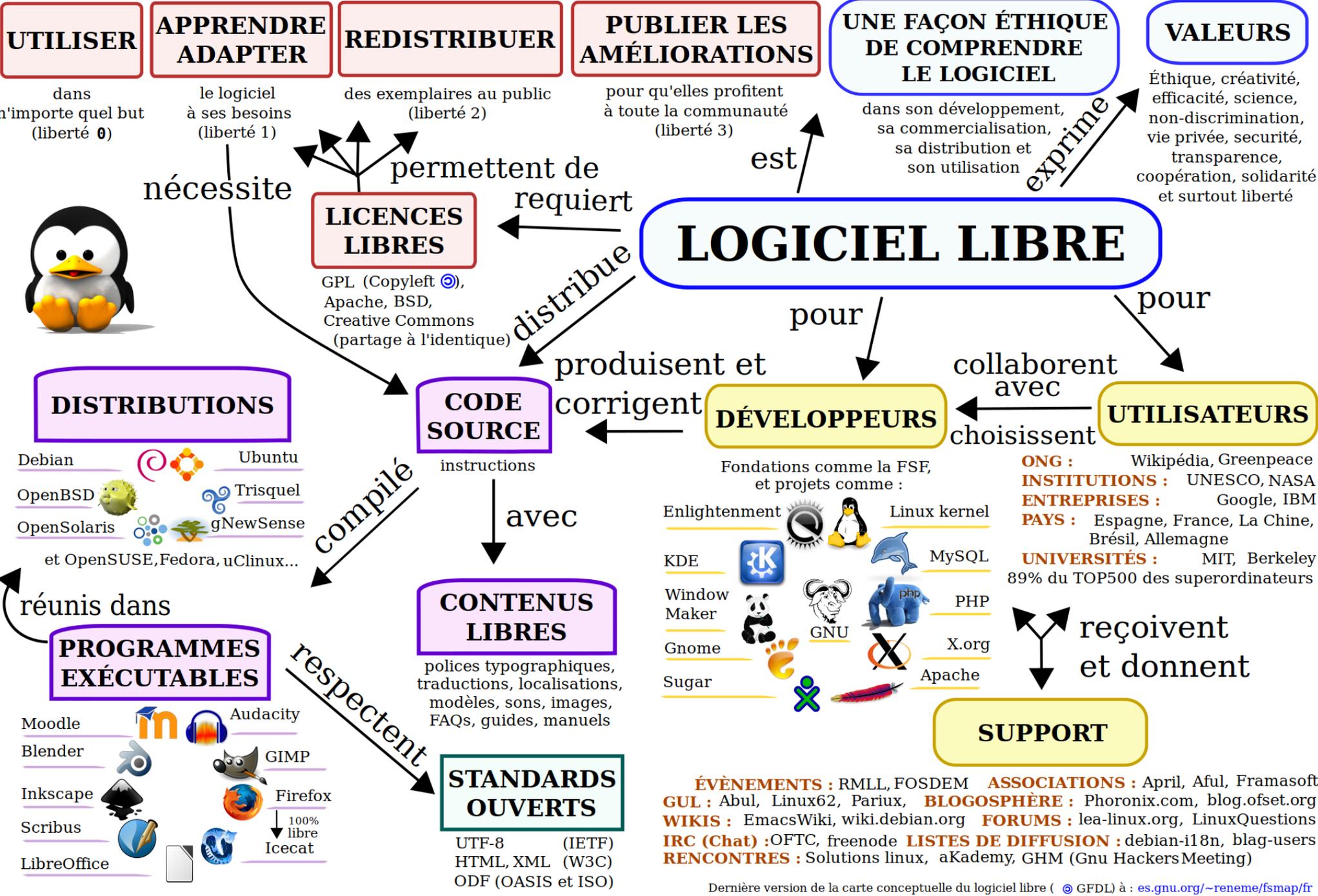
Serveurs : Consommation *idle* (a vide / statique) importante – faiblement proportionnelle

Réseau : consommation constante insensible à l'usage



Des pistes ?





UTILISER
dans
n'importe quel but
(liberté 0)

**APPRENDRE
ADAPTER**
le logiciel
à ses besoins
(liberté 1)

REDISTRIBUER
des exemplaires au public
(liberté 2)

**PUBLIER LES
AMÉLIORATIONS**
pour qu'elles profitent
à toute la communauté
(liberté 3)

**UNE FAÇON ÉTHIQUE
DE COMPRENDRE
LE LOGICIEL**
dans son développement,
sa commercialisation,
sa distribution et
son utilisation

VALEURS
Éthique, créativité,
efficacité, science,
non-discrimination,
vie privée, sécurité,
transparence,
coopération, solidarité
et surtout liberté

LOGICIEL LIBRE

LICENCES LIBRES
GPL (Copyleft ©),
Apache, BSD,
Creative Commons
(partage à l'identique)

DISTRIBUTIONS

Debian, Ubuntu, OpenBSD, Trisquel, OpenSolaris, gNewSense, et OpenSUSE, Fedora, uClinux...

CODE SOURCE

instructions
avec

CONTENUS LIBRES
polices typographiques,
traductions, localisations,
modèles, sons, images,
FAQs, guides, manuels

PROGRAMMES EXÉCUTABLES

Moodle, Audacity, Blender, GIMP, Inkscape, Firefox, Scribus, LibreOffice, Icecat

STANDARDS OUVERTS
UTF-8 (IETF)
HTML, XML (W3C)
ODF (OASIS et ISO)

produisent et corrigent

DÉVELOPPEURS

Fondations comme la FSF, et projets comme :
Enlightenment, Linux kernel, KDE, MySQL, Window Maker, PHP, Gnome, X.org, Sugar, Apache

collaborent avec
choisissent

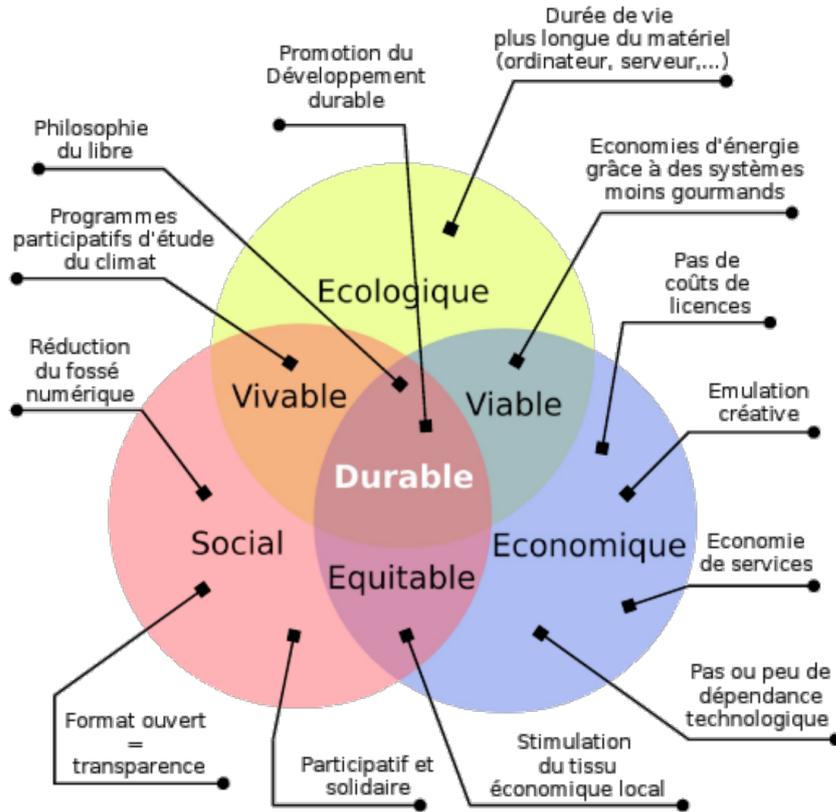
UTILISATEURS

ONG : Wikipédia, Greenpeace
INSTITUTIONS : UNESCO, NASA
ENTREPRISES : Google, IBM
PAYS : Espagne, France, La Chine, Brésil, Allemagne
UNIVERSITÉS : MIT, Berkeley
89% du TOP500 des superordinateurs

reçoivent et donnent
SUPPORT

ÉVÈNEMENTS : RMLL, FOSDEM **ASSOCIATIONS :** April, Aful, Framasoft
GUL : Abul, Linux62, Pariux, **BLOGOSPHERE :** Phoronix.com, blog.ofset.org
WIKIS : EmacsWiki, wiki.debian.org **FORUMS :** lea-linux.org, LinuxQuestions
IRC (Chat) : OFTC, freenode **LISTES DE DIFFUSION :** debian-i18n, blag-users
RENCONTRES : Solutions linux, aKademy, GHM (Gnu Hackers Meeting)

Dernière version de la carte conceptuelle du logiciel libre (© GFDL) à : es.gnu.org/~reneme/fsmmap/fr
Auteur : René Méroü (h@es.gnu.org) Matériel didactique traductions et commentaires V.5 2011-07-13



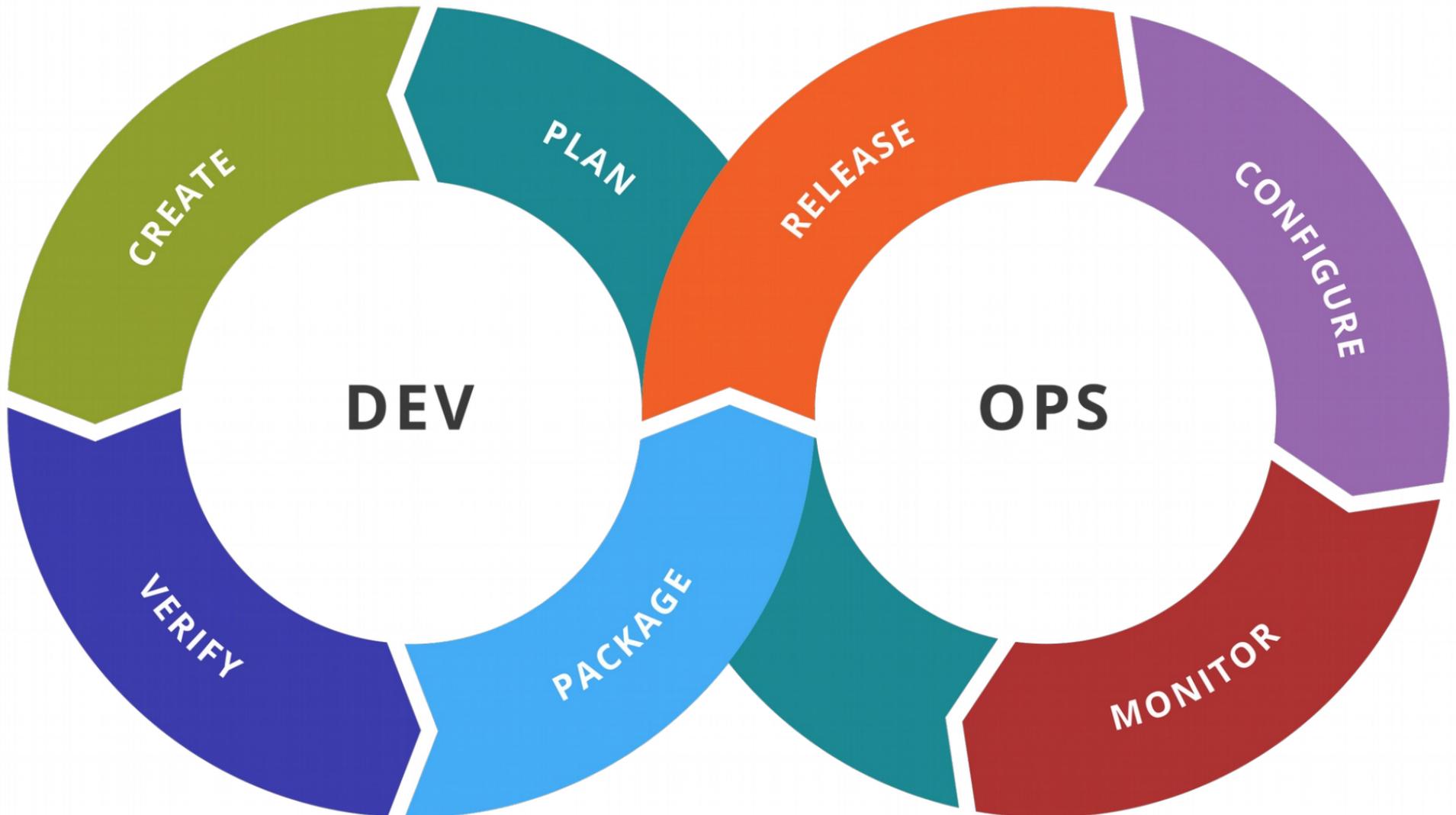
Logiciels libres / Développement durable

- Utilisent souvent moins de ressources matérielles
- Évitent l'obsolescence programmée
- Modèle économique basé sur le service et non la rente
- Méritocratie
- Favorise : l'amélioration participative, les formats libres, l'adaptation à des besoins, la créativité et l'innovation

Patates durables ou convergence entre logiciels libres et développement durable

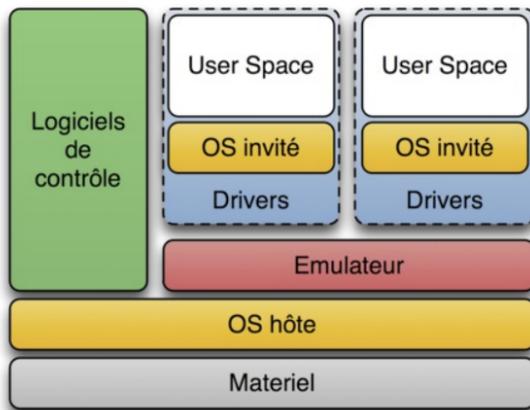


Le logiciel aujourd'hui : agilité, devops

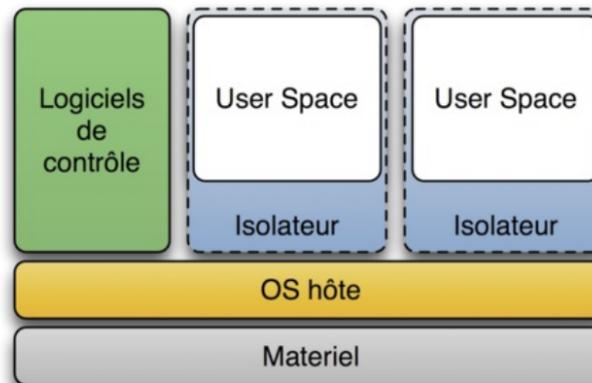


Impacts de la Virtualisation (VM, docker, k8s)

- ▶ 2 approches : hyperviseur (vmware) vs isolation (lxc, docker)
- ▶ Avantages : consolidation sur machine hôte (partage + allocation des ressources), répartition charge, supervision?
- ▶ Inconvénients : transferts images via réseau (docker hub), complexité croissante, accélération (docker) => effet rebond ?



vmware



docker



Image ecoinfo_anf

- 1.66GB
- Build : ubuntu 18.04 (64.2MB) + 510 Mb (apt)

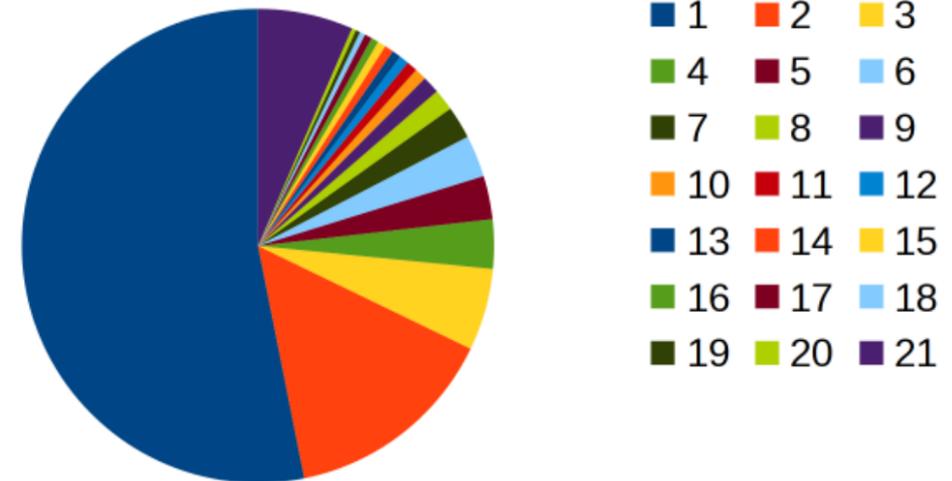
Impact Intégration Continue (Gitlab CI)



- ▶ Forge Gitlab : CI basé sur docker
- ▶ Exemple : Gricad-gitlab (UGA)

- 77 000 jobs / 460 projects
- 292j (durée totale)
- Top 20 (5 projets ~ 75%)

Gitlab CI 20 top projects

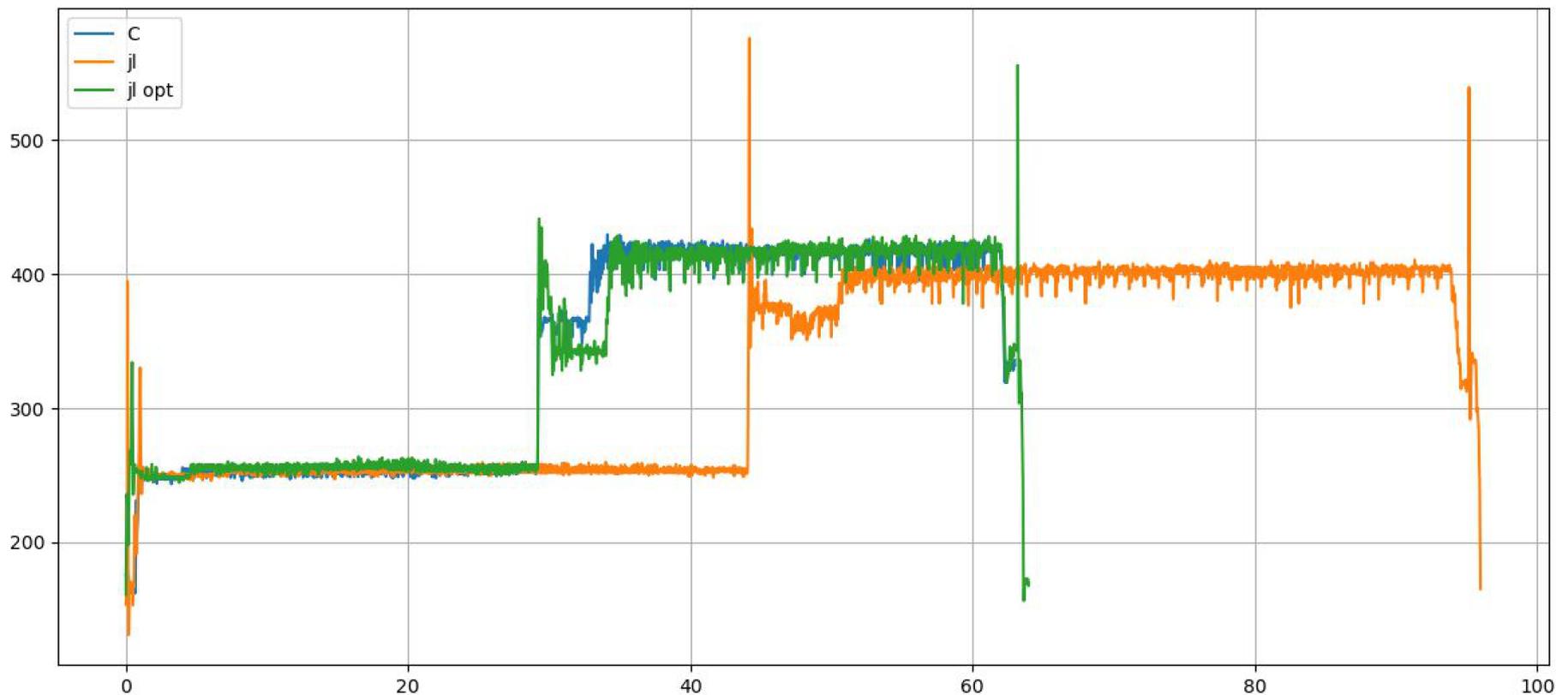


- ***Métrique durée n'indique rien sur cpu + io + network !***

- ▶ Avantages : build + tests + packaging (+ déploiement?)
- ▶ Attention à ne pas en abuser, pistes : tag no_ci, job schedule differed, periodic, branches ...

Etude perf / conso Languages (HPC)

- Analyse approfondie sur Grid5000 (temps / énergie) sur les langages : **Fortran**, **Python**, **Julia**, C, Java, Go, Rust



Plus concrètement

- ▶ Répondre au besoin et pas à côté : Ingénierie des exigences, formaliser le besoin
- ▶ Dégradation de la « qualité » du service en fonction de la charge ou/et du choix de l'utilisateur
- ▶ Penser au matériel utilisé dans la phase d'usage :
 - algorithme, options de compilateurs/environnement, hiérarchie mémoire
- ▶ Agir sur et avec le Hardware (fréquence, tension, architectures big/little)
- ▶ S'appuyer sur des librairies/bibliothèques/langages/os optimisées
- ▶ Sobriété : halte à l'inflation (VM, volumes, savoir arrêter un service, supprimer les données : cycle de vie de la donnée)
- ▶ Attention à la complexité : interdépendances, workflow des différentes couches logicielles, simplicité = réutilisabilité
- ▶ Attention à la paternité
- ▶ Guide pour le web : <https://greencodelab.github.io/>

Conclusion / Perspectives ?

- Adopter des Bonnes pratiques ?
- Sobriété fonctionnelle et matérielle
- Optimiser CPU / mémoire, minimiser les transferts de données (au plus près)
- Attention à l'Accélération, la Complexité, l'inflation:
 - transferts réseau
 - docker, VM (hw)
 - CI / mises à jour / déploiement

Sans réfléchir, tu coderas

Comme débogger printf, tu utiliseras

Pour améliorer tes performances du nouveau matériel, tu optimiseras

Fréquemment la mémoire, tu alloueras

Ton logiciel, tu surdimensionneras

Du printf, tu abuseras

Sur un jeu de données complet, toujours tu testeras

Toujours dans le même langage, tu coderas

Dans les conférences aussi tu resteras

Jamais ton code tu ne partageras

