

Logiciel responsable (LoRe)

Chaire de recherche financée par le mécénat

Georges-André Silber

<https://mines.paris/lore>

25 septembre 2024

Institut des transformations numériques (ITN)

Centre de recherche en informatique (CRI)

Mines Paris — PSL

APPLICATIONS

CALCULS/IA/HPC/EDGE + MASSES DE DONNÉES

LOGICIELS FONDAMENTAUX

SYSTÈME , BIBLIOTHÈQUES , COMPILATEURS

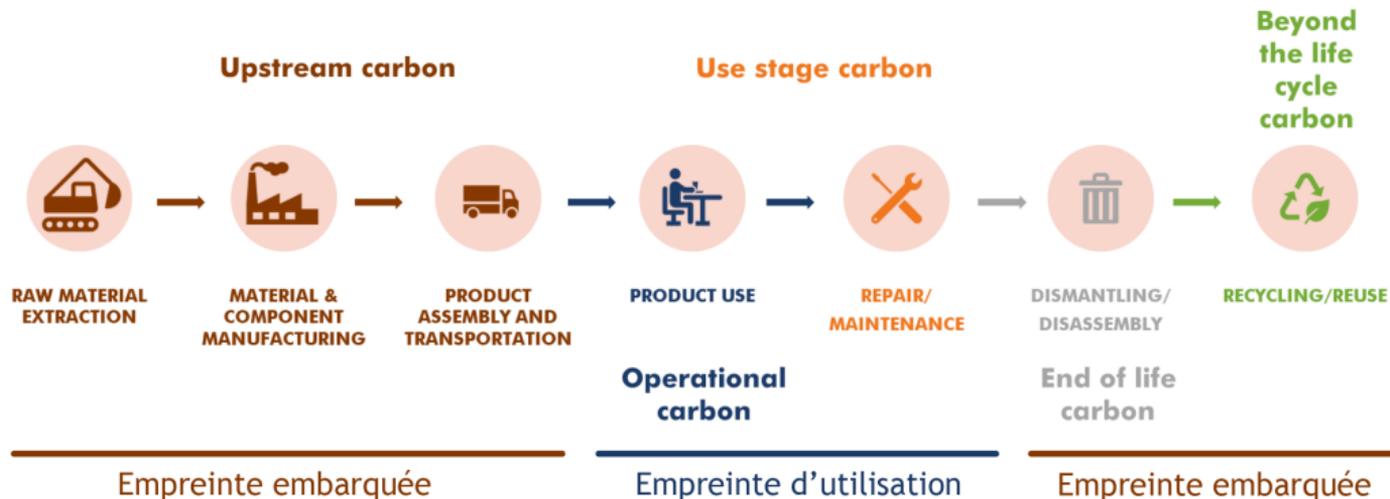
MATÉRIEL

CPU, GPU, RAM, STOCKAGE, RÉSEAU

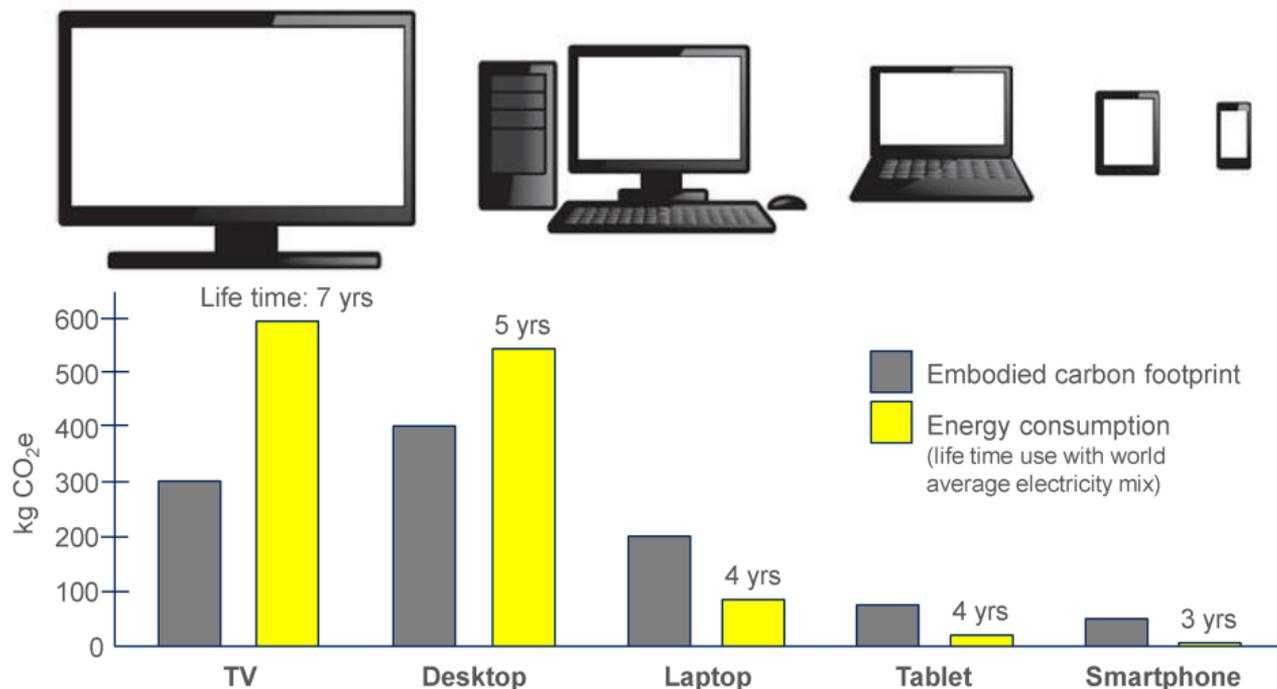
} Chaîne
Logiciel
Responsable

Améliorer les **logiciels libres** fondamentaux pour une **meilleure utilisation du matériel** par les **applications** : utiliser **moins d'énergie**, de **mémoire** et de **temps**.

L'impact environnemental croissant du numérique



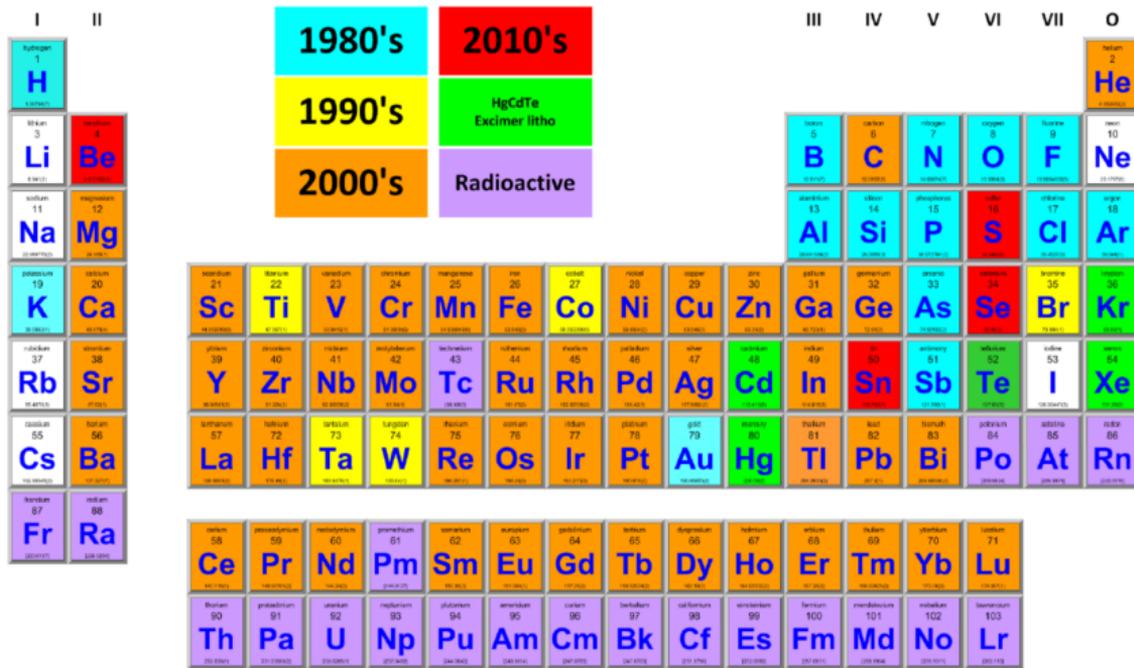
Source : L. Belkhir et al., «Assessing ICT global emissions footprint : Trends to 2040 & recommendations», J. of Cleaner Production, 3/2018.



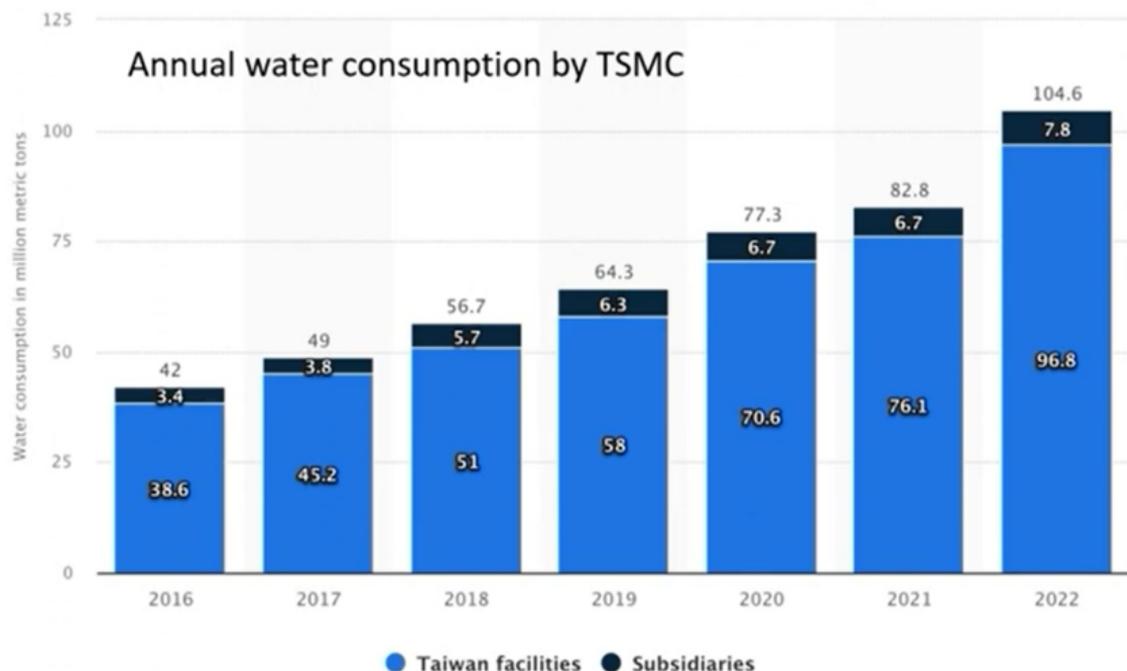
Source : J. Malmodin et D. Lundén, «The Energy and Carbon Footprint of the Global ICT and E&M Sectors 2010–2015», Sustainability, 9/2018.



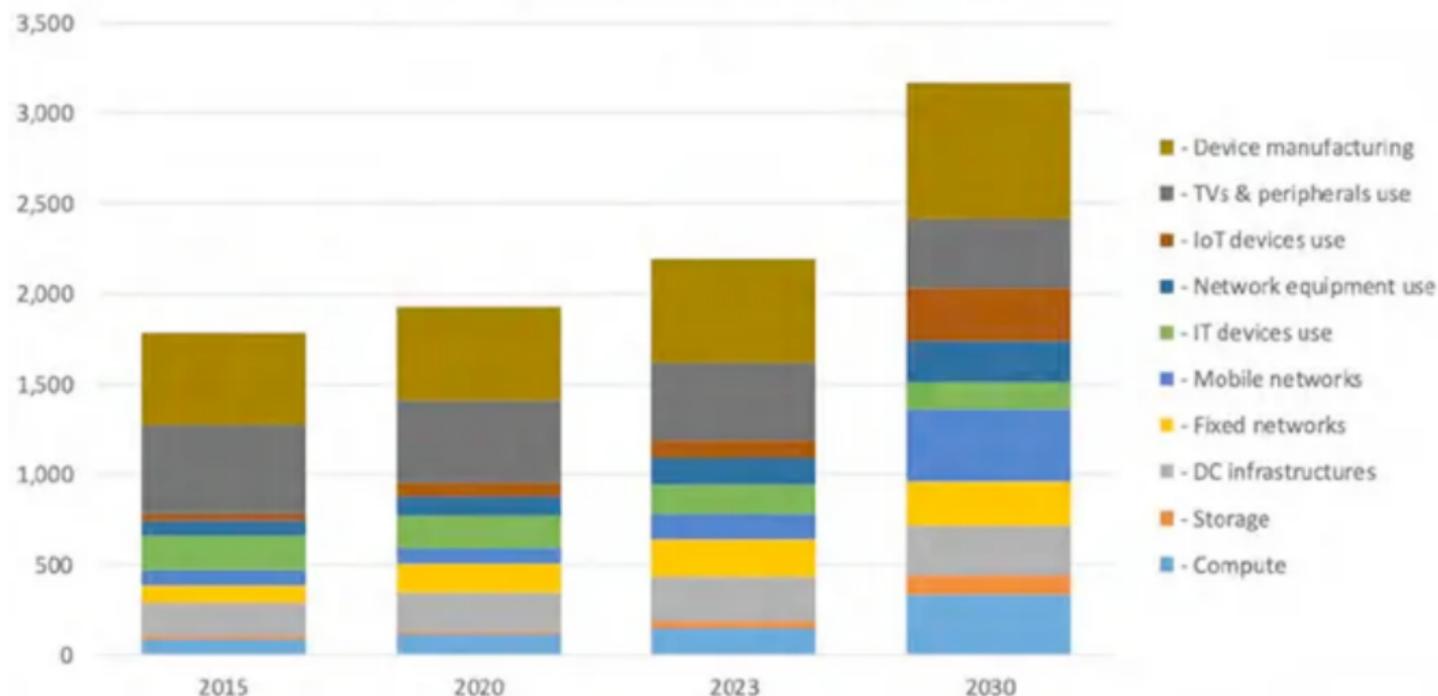
Crédit : Arnout Fierens (<https://arnulf.be>).



Source : T. Ernst, L.-Å. Ragnarsson, et J.-P. Raskin, «Sustainable materials and production», HiPEAC Vision 2024, janv. 2024.



Source : L. Eeckhout, «Sustainable computer systems», HiPEAC 2024 Keynote 2, 23 février 2024.
<https://www.youtube.com/watch?v=jUw7QQ-a00Q>



Source : L. Eeckhout, «Sustainable computer systems», HiPEAC 2024 Keynote 2, 23 février 2024.
<https://www.youtube.com/watch?v=jUw7QQ-a00Q>

Le logiciel : un fort potentiel d'amélioration

```
# Exemple: multiplication de matrices
for i in range(4096):
    for j in range(4096):
        for k in range(4096):
            C[i][j] += A[i][k] * B[k][j]
```

#	Method	Time (s)	Gflop/s	Speedup	Rel. sp.	% peak
1	Python	25 552,48	0,005	1	—	0,00
2	Java	2 372,68	0,058	11	10,8	0,01
3	C	542,67	0,253	47	4,4	0,03
4	Parallel loops	69,80	1,969	366	7,8	0,24
5	Parallel divide and conquer	3,80	36,180	6 727	18,4	4,33
6	plus vectorization	1,10	124,914	23 224	3,5	14,96
7	plus AVX intrinsics	0,41	337,812	62 806	2,7	40,45

Code	#	Machine	Country	Rpeak (Tflop/s)	Rmax (Tflop/s)	% Rpeak	MW	Eff. (Gflop/s/W)
HPL	1	Frontier	USA	1685	1 102	65	21.1	52.23
HPL	2	Fugaku	Japan	537	442	82	29.9	14.78
HPL	3	LUMI	Finland	428	309	72	6.0	51.38
HPCG	1	Fugaku	Japan	537	16	3,0	29.9	0.54
HPCG	2	Frontier	USA	1685	14,05	0,8	21.1	0.67
HPCG	3	LUMI	Finland	428	3,41	0,8	6.0	0.57

Un chaire sur le logiciel responsable

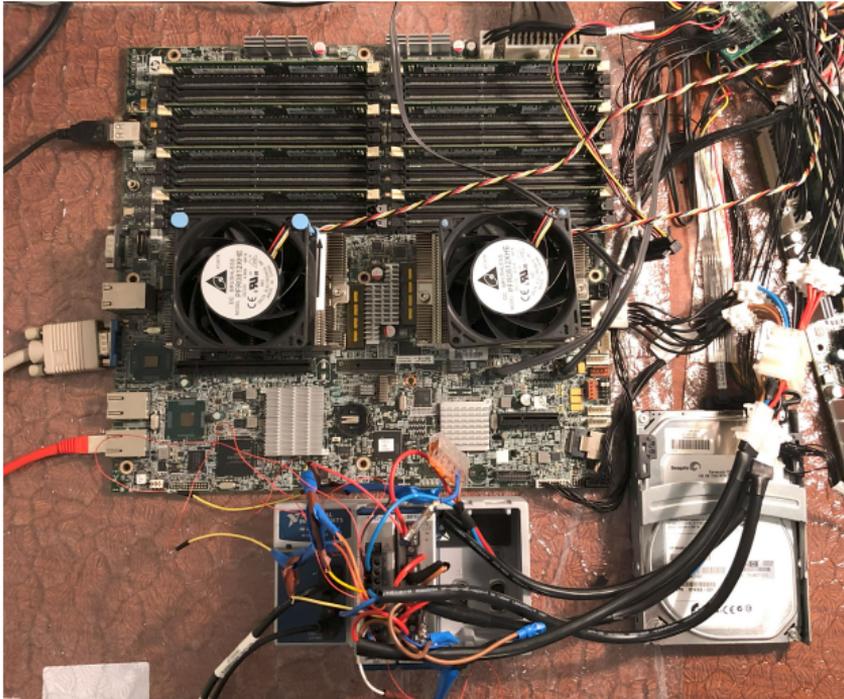
Rendre les **logiciels libres fondamentaux** plus **frugaux** et plus **efficaces**

1. **Mesurer** et **comprendre** finement l'utilisation de l'énergie par le logiciel
2. Créer des **modèles de coûts** en énergie, temps, mémoire
3. Utiliser ces modèles dans les **compilateurs optimisants**
4. Générer **automatiquement** du code parallèle **frugal** et **efficace**
5. **Réduire** la **taille des données** et la **quantité de calculs**
6. Rendre visibles les coûts en **énergie, temps, mémoire**
7. **Stratégies** de **placement** de tâches **sensible à l'énergie**

<https://mines.paris/lore>

- Firmware et système d'exploitation (CoreBoot, Linux)
- Virtualisation (KVM, Kubernetes, Docker, OpenShift)
- Compilateurs optimisants (GCC, LLVM, rust)
- Bibliothèques (libc/libc++, libm, BLAS/LAPACK)
- Packages (numpy, pytorch, tensorflow, jax, xla, keras, Scikit-learn)
- Interpréteurs (Python, NodeJS, GraalVM, OpenJDK)
- Langages améliorés : Python, JavaScript, C, C++, Rust, Java, Fortran

Sélection de **cas d'usages** et d'**optimisations** avec les entreprises.



- 20 points de mesure
- Peu de perturbations
- Sondes "maison"
- Calibration
- Benchmarks "maison"
- Microbenchmarks / Picobenchmarks
- *bare metal*
 - comprendre
 - modéliser

- Conception de modèles de coûts statiques pour prédire le temps, la mémoire et l'énergie consommés par un logiciel (*axe 2 : modèles*)
- Intégration de modèles de coûts dans un environnement de développement (*axe 6 : visibilité*)
- Optimisations de compilateurs dirigées par l'énergie (*axe 3 : compilateurs*)
- Génération automatique de code optimisé pour le temps et l'énergie (*axe 4 : génération automatique de code optimisé*)
- Traduction de C++ à Rust (*axe 4 : génération automatique de code optimisé*)
- Compression et réduction de la taille des nombres flottants (*axe 5 : réduction de la taille*)
- Migration de conteneurs sensible à l'énergie sur le cloud (*axe 7 : placement*)

- 30 ans d'**expérience** dans le domaine de l'**optimisation du logiciel**
- Contributions **scientifiques** majeures : *tiling*, interprocédural, énergie ...
- Nombreuses contributions **logicielles**, dont l'outil PIPS
- Transferts dans l'**industrie**, par des contrats et anciens élèves
- Contributions à des **compilateurs** industriels, dont GCC



git://gcc.gnu.org / gcc.git / blob

summary | shortlog | log | commit | commitdiff | tree | blame | history | raw | HEAD

commit ? search: re

RISC-V: Add testcases for vector truncate after .SAT_SUB

[gcc.git] / gcc / tree-loop-distribution.cc

```
1 /* Loop distribution.
2  Copyright (C) 2006-2024 Free Software Foundation, Inc.
3  Contributed by Georges-Andre Silber <Georges-Andre.Silber@ensmp.fr>
4  and Sebastian Pop <sebastian.pop@amd.com>.
5
6 This file is part of GCC.
7
```

- Nouvelle option des **compilateurs** pour **optimiser l'énergie** (gcc -Oe)
- Intégration aux environnements de développements de nouveaux outils pour **visualiser l'énergie** consommée par le code, au moment de sa **conception**
- Nouveaux mécanismes intégrés aux **systèmes d'exploitation** (Linux) et aux orchestrateurs de **conteneurs** (Kubernetes) pour faire fonctionner les systèmes en **minimisant l'énergie**
- Changement sociétal : un **usage prolongé du matériel** (> 10 ans?)

- Chaire de **5 ans** renouvelable
- 1 **responsable scientifique** permanent à temps plein
- 5 **enseignants-chercheurs** en informatique du CRI collaborant à la chaire
- 230 **enseignants-chercheurs** pluri-disciplinaires des mines en appui
- 25 **stages ingénieur** de 6 mois
- 6 **thèses** de trois ans
- 8 **post-docs** de deux ans
- 4 postes d'**ingénieur de recherche** pendant 5 ans
- **matériel** de mesure et de développement logiciel
- **diffusion des connaissances** (publications, conférences)

- **Projets d'ingénierie** en 2A des mines : **informatique soutenable**
- *CodeMineur* : stages rémunérés de 6 mois pour les élèves. Modèle : [Google Summer of Code](#). **Développer** un point particulier dans un **logiciel libre fondamental**. Sujets définis avec les entreprises participant à la chaire
- **Séminaires, tutoriels** en ligne, organisation de **formations** auprès des entreprises adhérentes

- **Partenaires recherchés** : entreprises produisant du logiciel ou utilisant des logiciels fondamentaux open source
- **Modalités d'adhésion** : cotisation modulable de 100 k€ par an (réduction d'impôts de 60 %)
- **Objectif de financement** : 5 partenaires. Budget total de 500 k€/an, soit 2,5 M€ pour 5 ans
- **Gouvernance** : programme scientifique, comité d'orientation incluant les entreprises mécènes. Rapport scientifique annuel
- **Dissémination** : contributions aux logiciels libres. Publications dans des journaux et conférences. Séminaires et workshops ouverts aux mécènes
- **Recherche partenariale** : approfondissement de certains sujets industriels par contrat de partenariat ou thèse CIFRE avec PI spécifique
- **Contact** : Georges-André Silber
georges-andre.silber@minesparis.psl.eu +33 6 47 11 52 08