

# Conf Tech

# KALRAY

→ Grenoble, Pionnier en ManyCore

LISP  
TRUC HORRIBLE

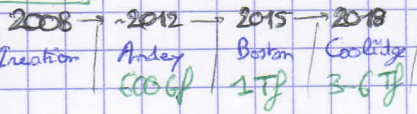
## Terms:

Hand	Soft	Business	Admin
Front (EP) Back (MC)	Core (OS, Drivers) Support	Embedded (con) Data Center (strategie)	II

## avantages

- low latency
- productivity
- low consumption
- flexibility tasks
- diverse app on same MPPA

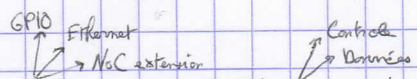
## histoire:



## productions

- boards intégrant chips
- many cores mppa

## A. MPPA BOSTON - Processor Architecture



Manycore Processor = 16 × Compute Clusters + 2 IO Clusters + 2 Network on chip

= 16 Cores + Interface N/C + DMA + Local Mem (L2 cache)

= Pipeline 9 stages VLIW

ARCHI

SOFT

OUTLS: \* Prog Env: C/C++ (ex style), OpenCL (gen style), APIs for Network, Crypto, Réseau de Neurones, Automotive

(avec ou sans RT à côté) (avec flash à côté pour prog) (réseau existant antérieur, possibilité d'exécuter plusieurs réseaux en m temps)

\* OS: Linux sur Quad Core (spécialisé), Pharo sur clusters, Communication Libraries, Libraries Optimisées

(low coars, controle, ordonnanceur) (NoC OS ou bare-metal, OpenMP) (com entre clusters, ext) (OpenSSL, lib c)

\* Simulation, Debug: Build Debug Simulation System Trace

(gcc, lib c, bin utils) (gdb, breakpoint) (cycle-based, trace acc) (linux trace like)

\* Traces: avec archi spécialisée pour sortir les données

⚠ Pas de cohérence de caches! Mémoire Distribuée! → Difficulté de programmation

## B. MPPA: Memories

CACHE

→ Cohérence de cache = peu variable protégée → invalidation de la donnée → Mais ici pas de cohérence hardware

→ Comment? → Invalidation software (méthode dédiée)

→ By-pass (directement en mem) ⊕ pas touches au cache (juste ce qu'il faut dans le cache = opt)

→ Toolchain: Compilateur // Assembleur // Éditeur de Lien → Pour MPPA ?

→ Comment? → Tout réécrit (long, complexe)

→ Open source ⊕ Re-targeting

- Assembleur [Assembly → Elf] lui prendra instructions, opérandes, relocation (et pas connues) 64 bits
- Linker [Elf, 0 → Executable] met tout en mem puis remplit les trous plus avec de la 0? ajout de sort 64 bits
- Compilateur [Source → Assembly] avec target et optimisation, utilisations pile, registres... (archi & registres)

Lisp pour logique, combiner tout pour pipeline = bundle

## C. MPPA COOLIDGE

- ⊕ Clock Freq ↑
- ⊕ Optimisé pour data centers et embedded
- ⊕ 64 bits natifs
- ⊕ Cache coherence
- ⊕ Memory Map unifiée
- ⊕ de clusters (80)
- ⊕ Débit Ethernet/Bus
- ⊕ Low Power
- ⊕ x2 taille cache, 4-way
- ⊕ Sécurité avec secure core
- ⊕ Doublement possible avec PCI et N/C reliés
- ⊕ Scalability supportée