# Accelerator-based Computing and Manycore Workshop

SLAC décembre 2009

http://www.lbl.gov/cs/html/manycore.html

# Le sujet

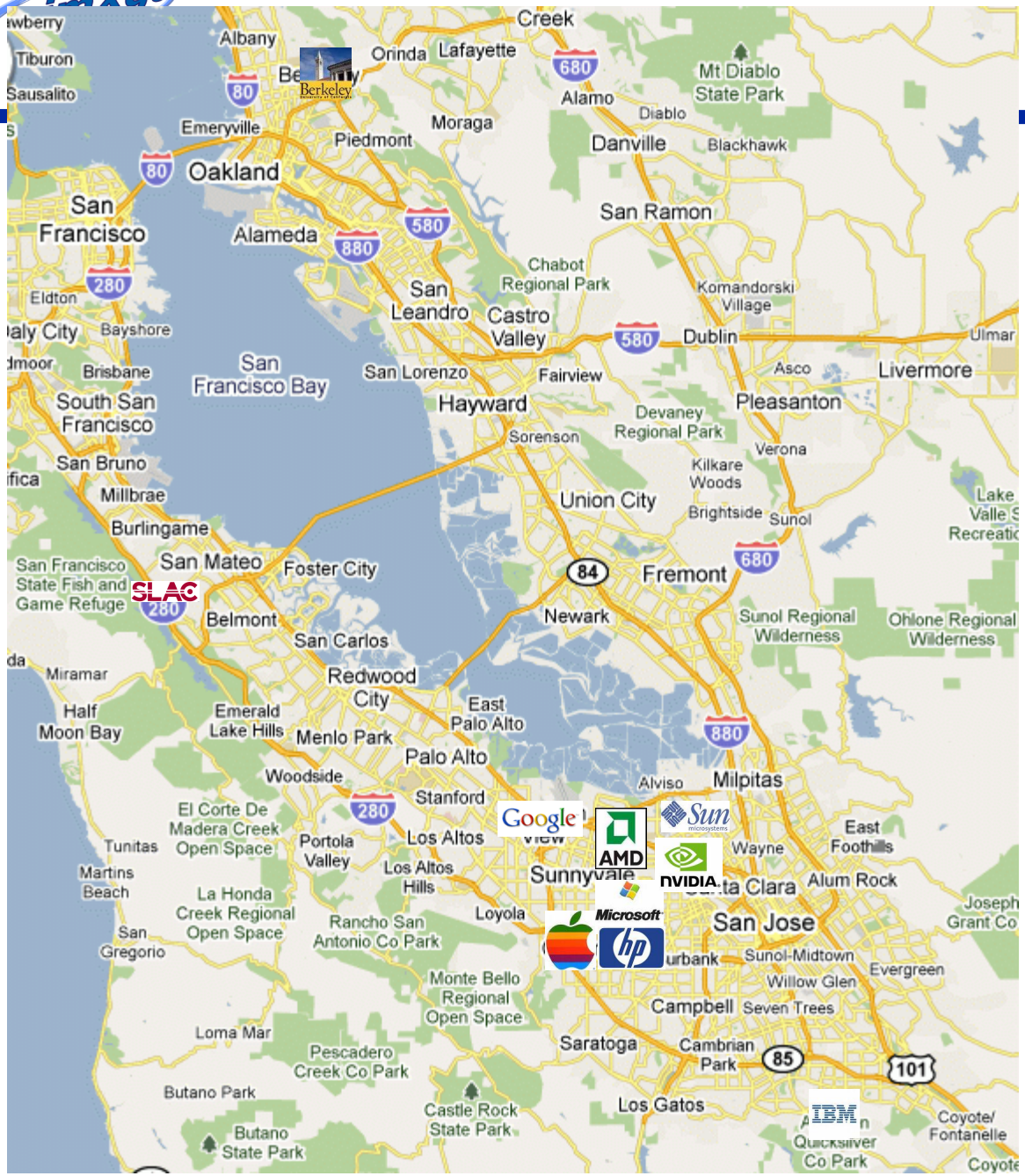
Comment avoir les performances nécessaires
dans les 10 à 15 années à venir

Approche traditionnelle : augmenter la fréquence d'horloge

Mais les limites sont atteintes

Solutions possibles :

La grille
Multicore, manycore
GPU
FPGA
Processeurs spécifiques (Grape)




Au coeur de l'action

# Les participants

40 personnes environ

Activités :

- Centres de calcul
- Cosmologie : N-body simulations (gravité)
- Dynamique moléculaire
- LSST
- Optique adaptative
- Seti
- Murchison Wide-field Array radiotelescope
- Imagerie médicale
- Nvidia
- AMD

Nationalité :

- US
- Chine
- Japon
- Taiwan
- Allemagne
- France

# Agenda 1

| Monday November 30, 2009 | | |
|---|---|---|
| 8:30–9:00 am | Registration / Refreshments | |
| 9:00–9:30 am | Tom Abel<br>Hemant Shukla | Introduction |
| Chair- John Shalf | | |
| 9:30–10:30 am | Kathy Yelick | Keynote (LBNL) |
| 10:00–11:00 am | Break/ Refreshments served | |
| 11:00–11:30 am | Alain Bonissent | Accelerated image reconstruction on a cluster of two AMD GPUs in X-ray CT with nonuniform detector geometry. |
| 11:30-12:00 pm | Keigo Nitadori | High Order N-body integrators on GPU |
| 12:00 – 2:00 pm | Lunch | |
| Chair- Hemant Shukla | | |
| 2:30–3:00 pm | Joel Primack | New University of California AstroComputing Institute |
| 3:00–3:30 pm | Dan Werthimer | Peta-Scale Computing Challenges in SETI and Radio Telescope Arrays |
| 3:30–4:00 pm | John Shalf | Hardware and Software Scaling Challenges for Future Systems |
| 4:00–4:30 pm | Break/ Refreshments | |
| 4:30–5:00 pm | Tom Abel | KIPAC |
| 5:00–5:30 pm | Lyne Jones | LSST |
| 6:30 pm | Dinner: Chef Chu's 1067 N San Antonio Road Los Altos, CA 94022-1300 | |

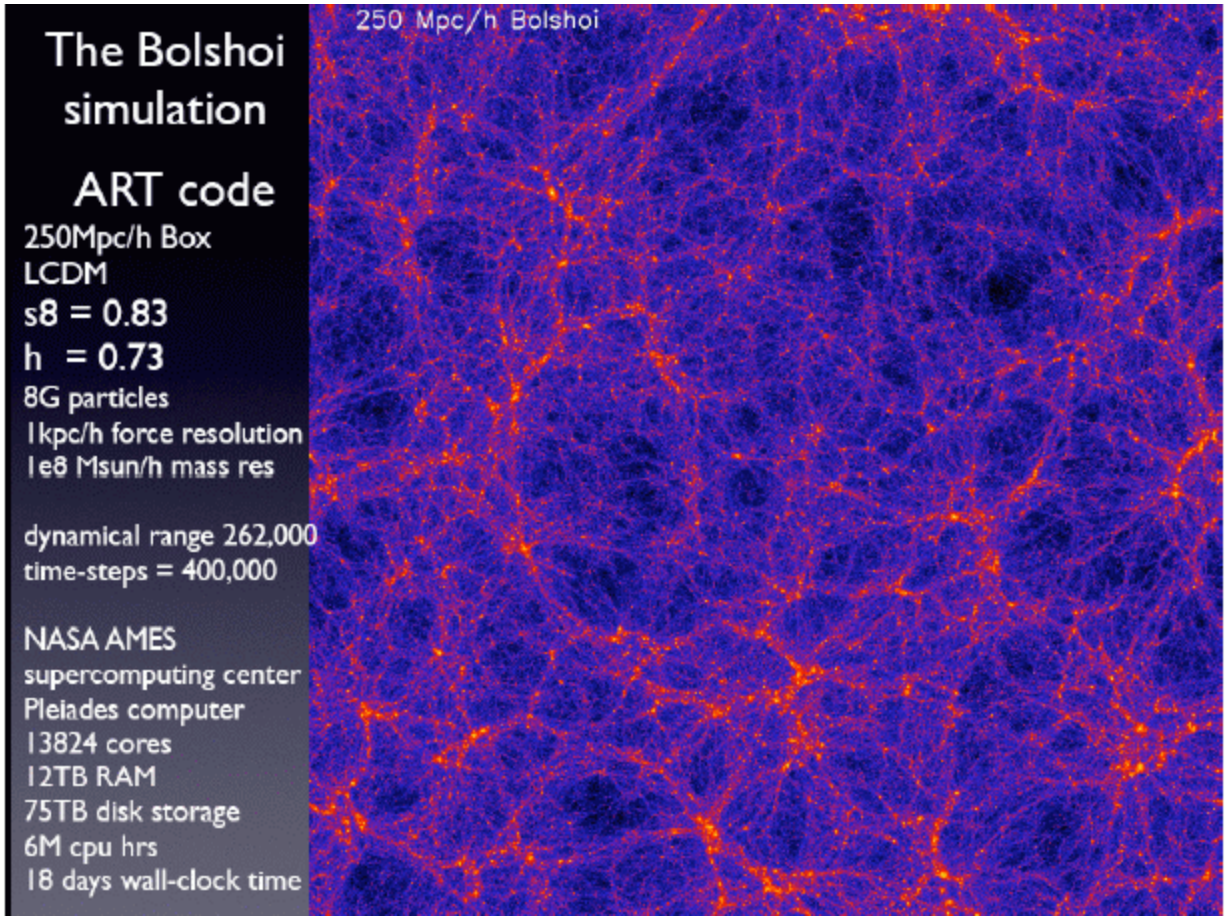
# Agenda 2

| Tuesday December 1, 2009 | | |
|---|---|---|
| 8:30–9:15 am | Refreshments | |
| Chair- Horst Simon | | |
| 9:15–10:00 am | Rainer Spurzem | Keynote (CAS) Green Supercomputing |
| 10:00–10:30 am | Mike Clark | Murchison Wide-field Array (MWA) |
| 10:30–11:00 am | Break/ Refreshments served | |
| 11:00–11:30 am | Don Gavel<br>Marc Reinig | Challenges in real-time control of adaptive optics systems for high resolution ground based astronomy |
| 11:30-12:00 pm | Wang Xiaowei | Lattice Boltzmann simulation of porous media on GPU-based parallel systems |
| 12:00 – 2:00 pm | Lunch | |
| Chair- Tom Abel | | |
| 2:30–3:00 pm | Chen Feiguo | Molecular dynamics simulation of multi-phase flow on micro-scales using CUDA |
| 3:00–3:30 pm | Peter Berczik | On super-massive binary black holes in galactic nuclei, recent results and software issues obtained with the IPE GPU cluster in Beijing. |
| 3:30–4:00 pm | Tsuyoshi Hamada | A Novel Multiple-Walk Parallel Algorithm for the Barnes-Hut Tree-code on GPUs – Towards Cost Effective, High Performance N-Body Simulation |
| 4:00–4:30 pm | Break/ Refreshments | |
| 4:30 pm | Collaboration Meeting | |

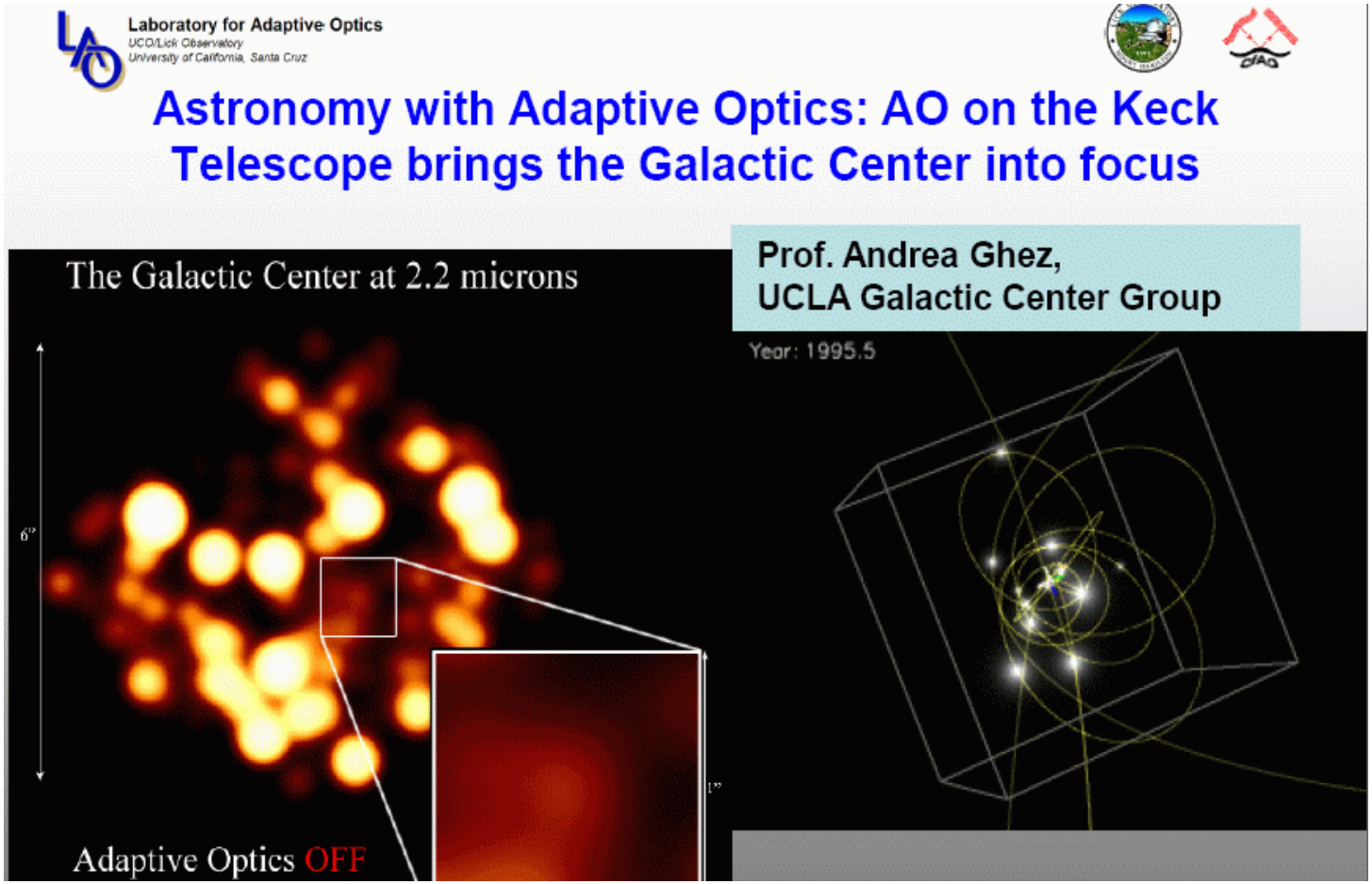
# Agenda 3

| | | |
|---|---|---|
| **Wednesday December 2, 2009** | | |
| 8:30–9:00 am | Refreshments | |
| **Chair- Rainer Spurzem** | | |
| 9:00–10:00 am | Billy Dally | Keynote (Industry) |
| 10:00–10:30 am | Ge Wei | Multi-scale discrete simulation on multi-scale HPC system (an overview of the software and hardware) |
| 10:30–11:00 am | Break/ Refreshments served | |
| 11:00–11:30 am | Jun Makino | GRAPE-DR and next-generation GRAPE |
| 11:30 – 1:30 pm | Lunch | |
| **Chair- Hemant Shukla** | | |
| 1:30–2:00 pm | Guillermo Marcus | Neighbour list generation with GPUs for SPH simulations |
| 2:00–2:30 pm | His-Yu Schive | GAMER: a GPU-Accelerated Adaptive Mesh Refinement Code for Astrophysics |
| 2:30–3:00 pm | Khronos | OpenCL Overview |
| 3:00–3:30 pm | Break/ Refreshments | |
| 3:30–4:00 pm | Udeepta Bordoloi | Heterogeneous computing on AMD using OpenCL™ |
| 4:00–4:30 pm | Rupak Biswas | |
| 4:30–5:00 pm | Tom Abel | |
| 5:00–5:30 pm | Horst Simon | Future and Conclusion |

# Activités : cosmologie : N-body simulation




Jo Primak, UC Santa Cruz

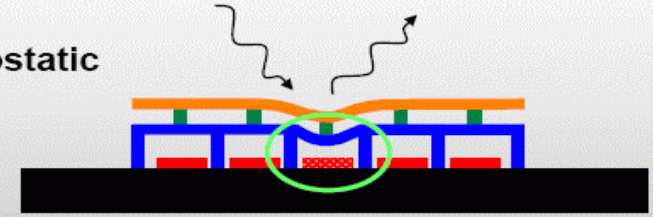



D. Gavel and M. Reining
Laboratory for Adaptative Optics
UC Santa Cruz

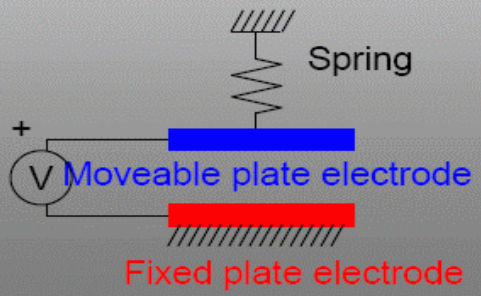
Laboratory for Adaptive Optics
UCO/Lick Observatory
University of California, Santa Cruz

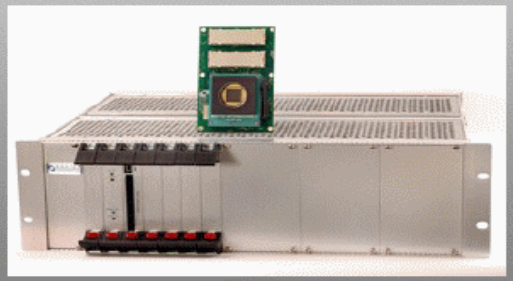
## Wavefront phase is corrected with a deformable mirror

MEMS deformable mirror with electrostatic actuators



Simplified actuator model:

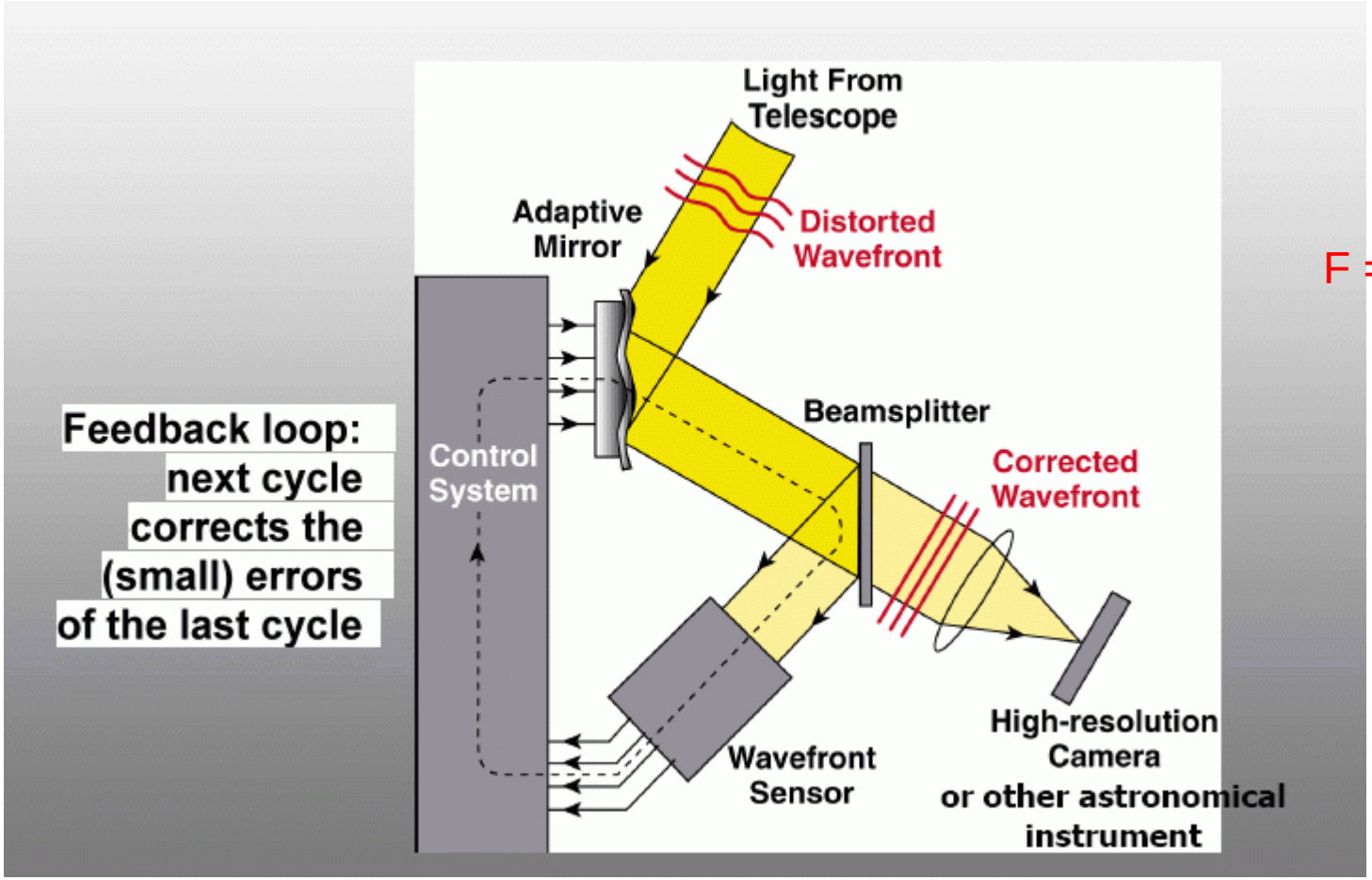


Diagrams and photo courtesy Boston Micromachines Corporation

D. Gavel and M. Reining
Laboratory for Adaptative Optics
UC Santa Cruz

# Activités : optique adaptative

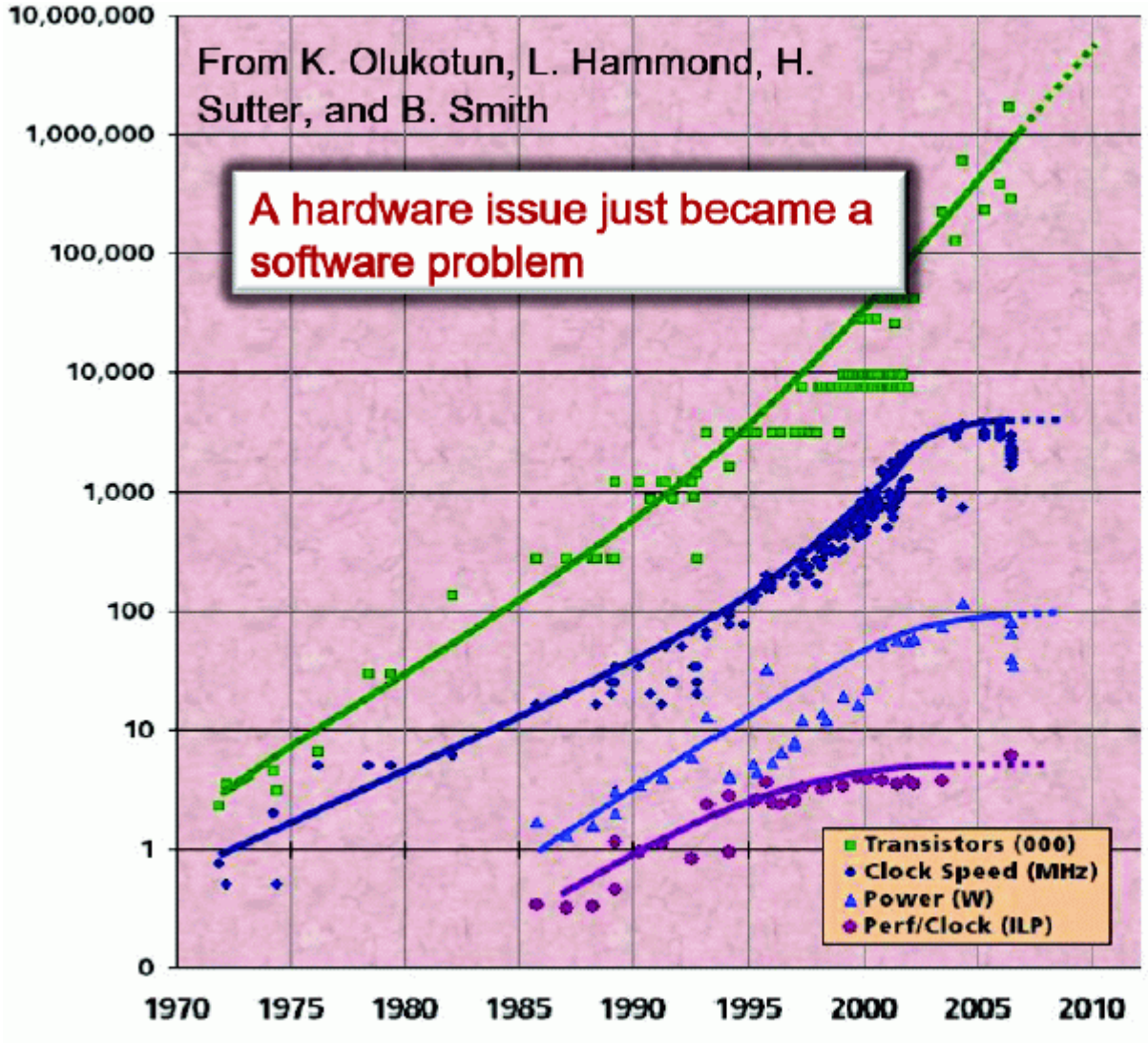



F = 1kHz

D. Gavel and M. Reining
Laboratory for Adaptative Optics
UC Santa Cruz

# Augmenter la fréquence pour plus de performances

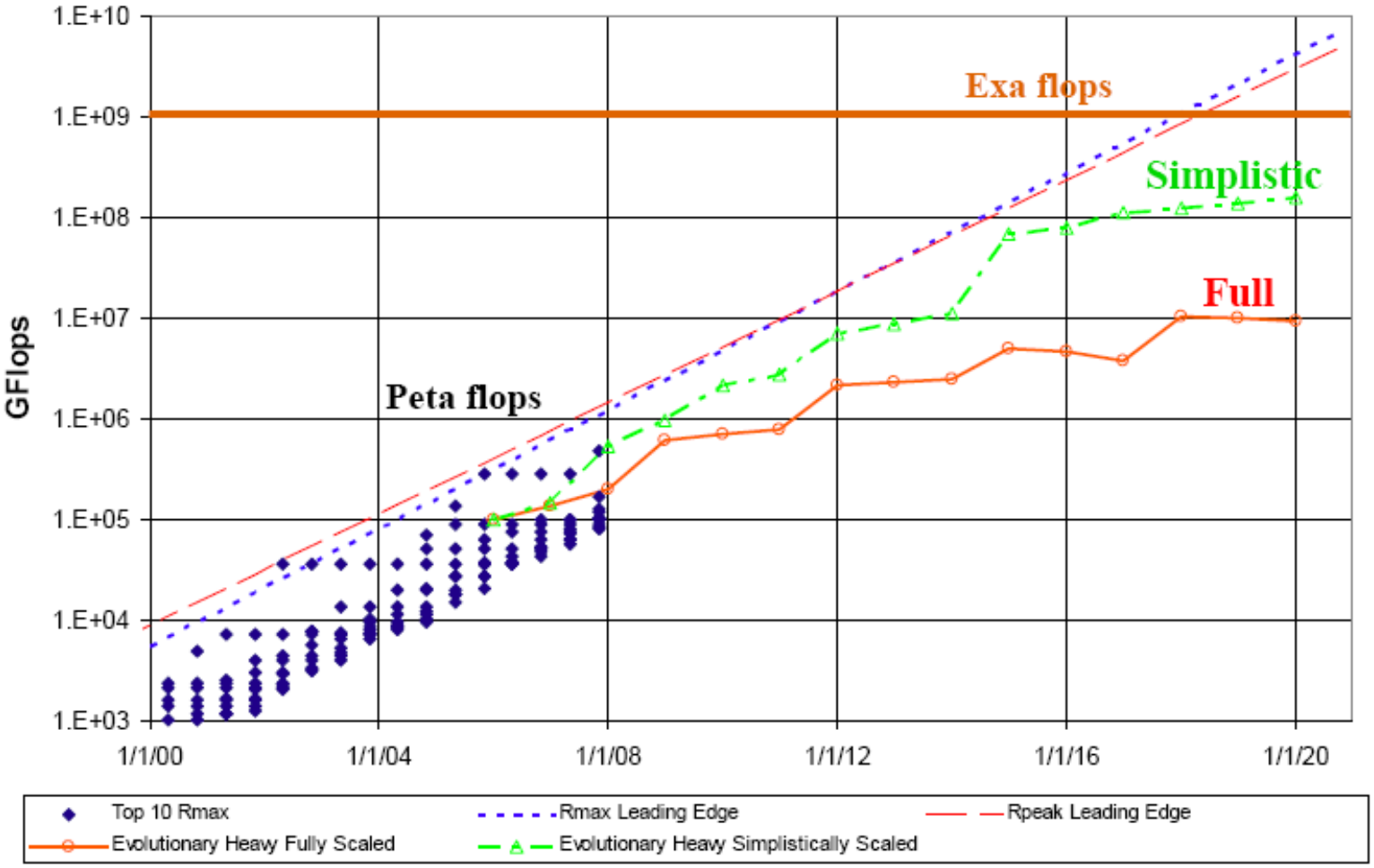
# La course à la fréquence est terminée




- In the "old days" it was: each year processors would become faster
- Today the clock speed is fixed or getting slower
- Things are still doubling every 18 -24 months
- Moore's Law reinterpretated.
  - Number of cores e, double every Study 18-24 months

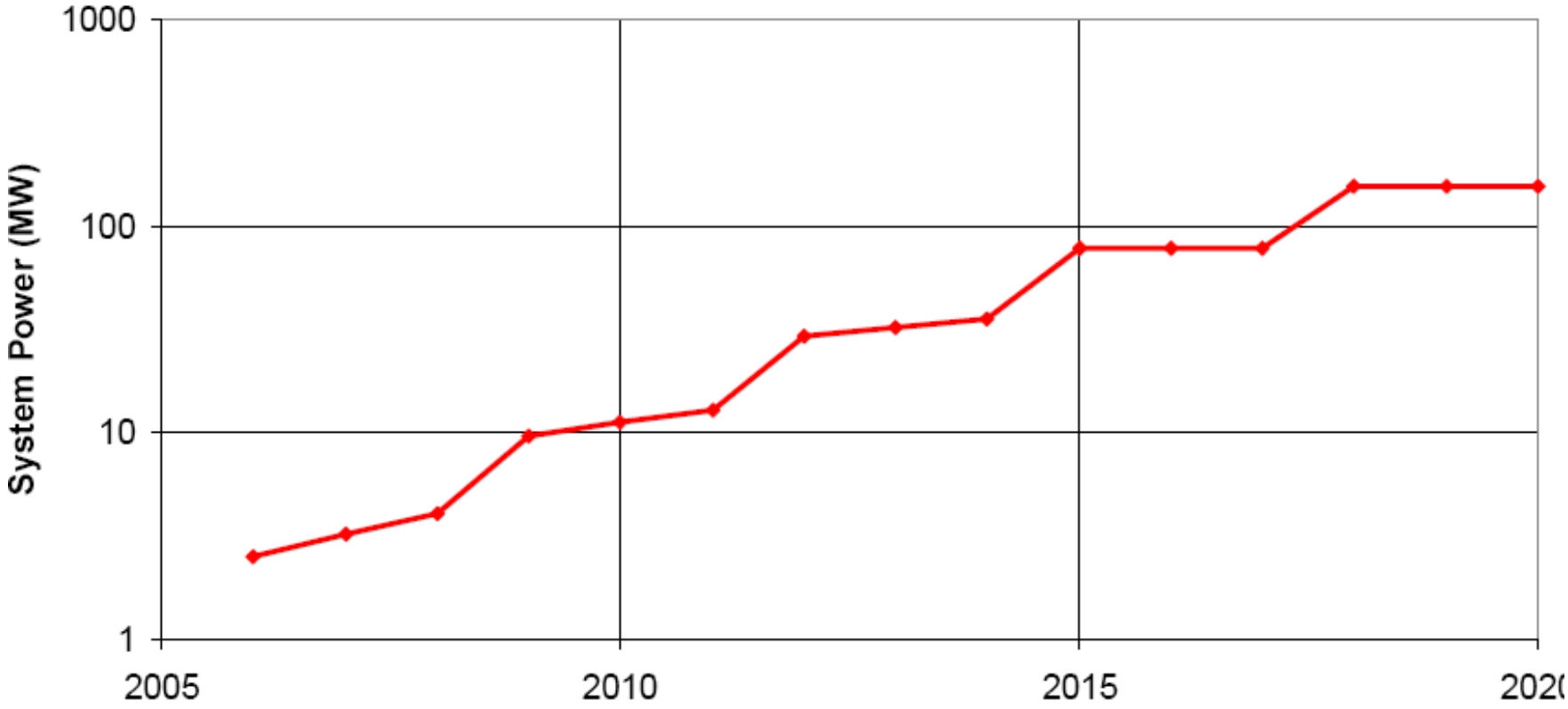
8

# Cannot continue Performance Scaling with Current Approach



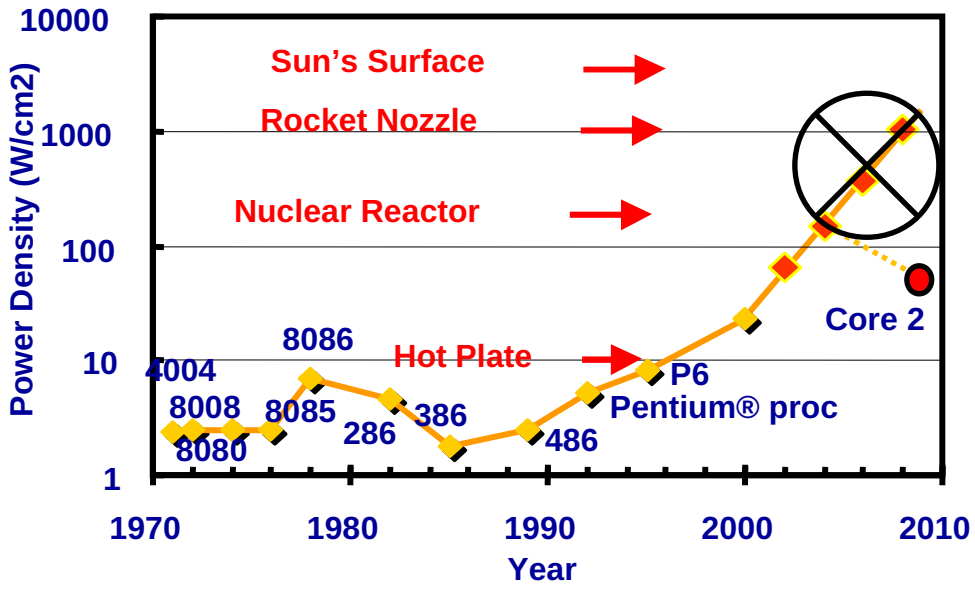

From Peter Kogge, DARPA Exascale Study

# … and the power costs will still be staggering




From Peter Kogge,
DARPA Exascale Study

# Parallelism to Recover Performance

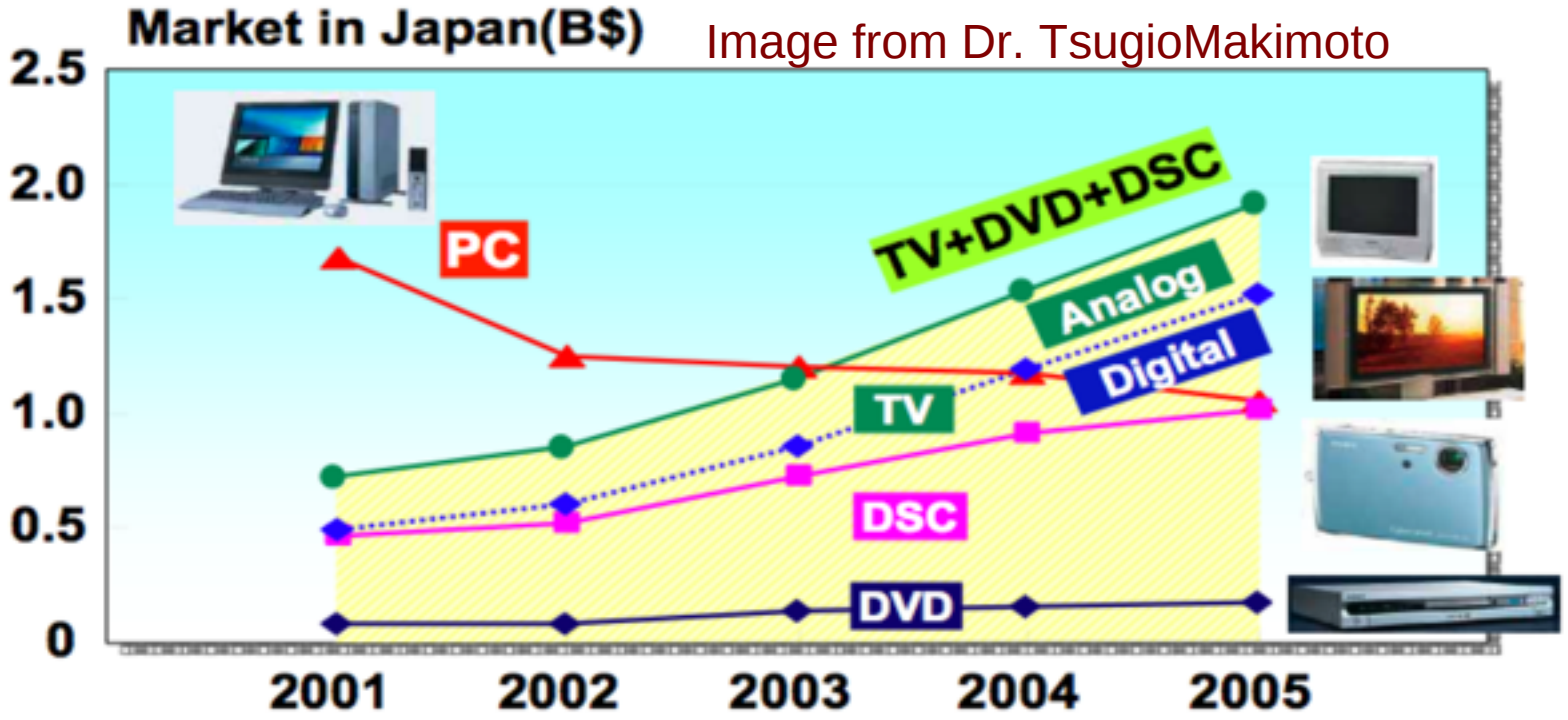



**Source: S. Borkar (Intel )**

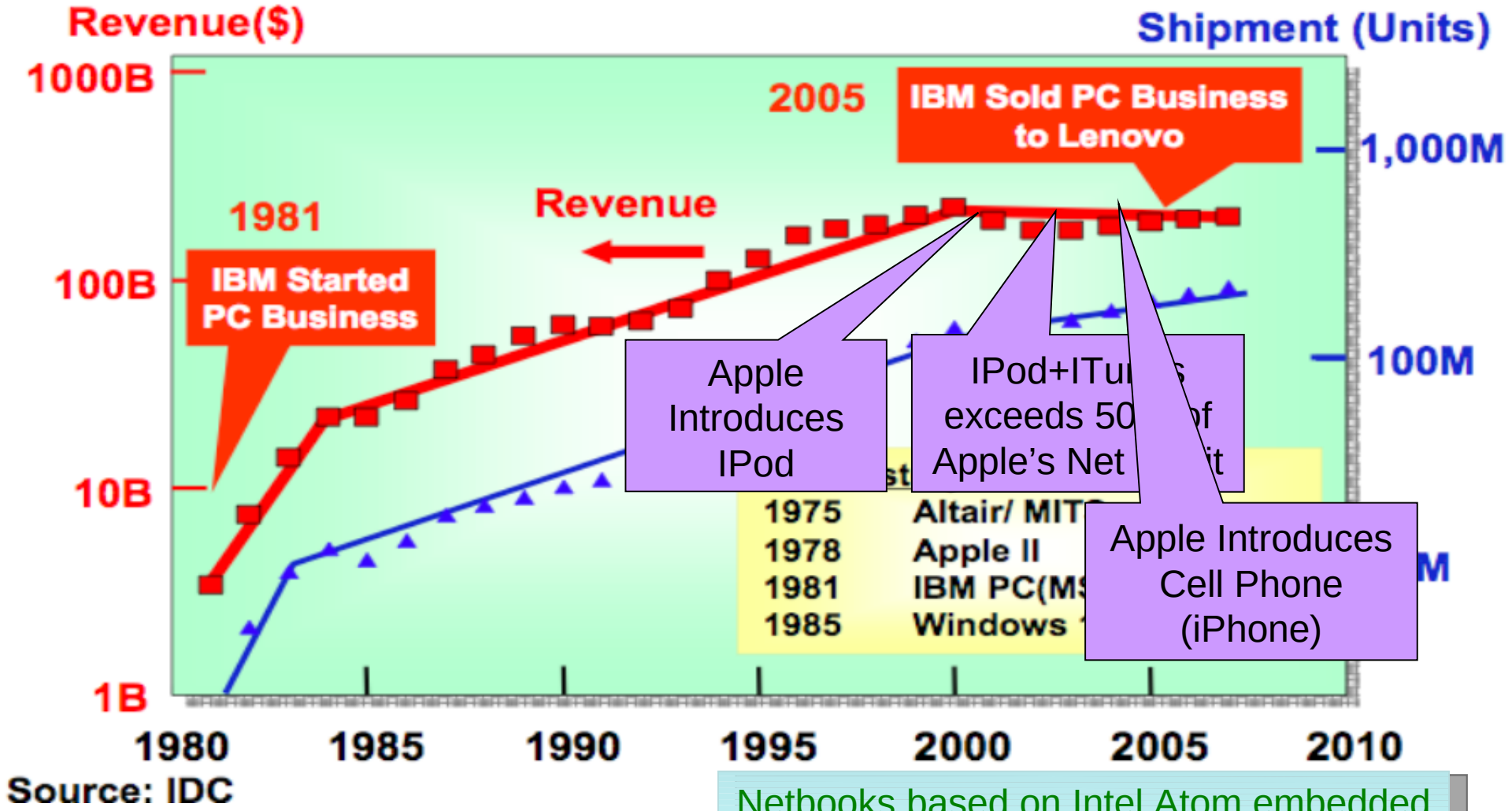
- Computing performance is now limited by power dissipation. This has forced the move to parallelism as principal means of increasing performance without increasing energy per operation.

# Leveraging Commodity Hardware Design Flow

- 1990s - R&D computing hardware dominated by desktop/COTS
  - Had to learn how to use COTS technology for HPC
- 2010 - R&D investments moving rapidly to consumer electronics/ embedded processing
  - Must learn how to leverage embedded processor technology for future HPC systems



# Consumer Electronics has Replaced PCs as the Dominant Market Force in CPU Design!!




From Tsugio Makimoto: ISC2006

Netbooks based on Intel Atom embedded processor is the fastest growing portion of "laptop" market.

# Solutions pour le calcul parallèle

# How Small is “Small”

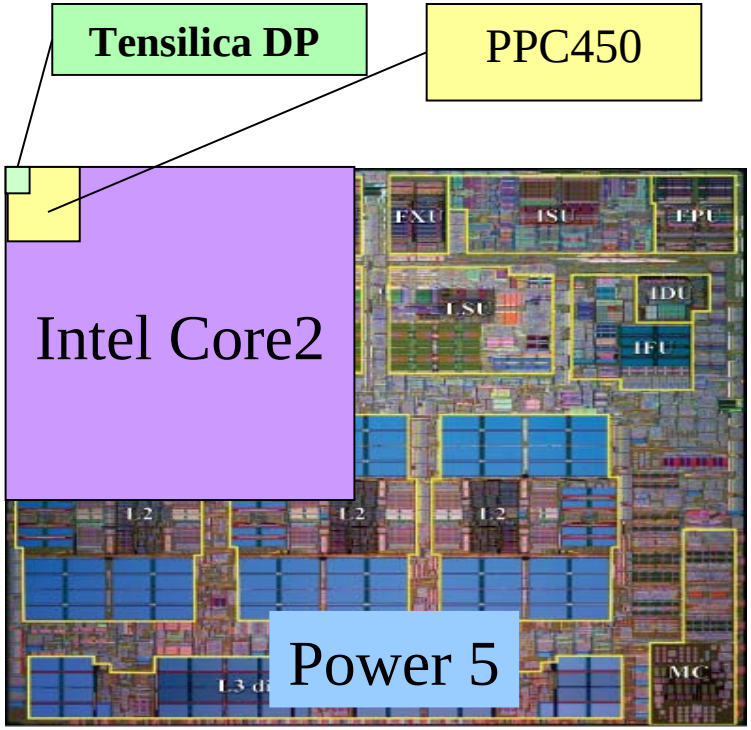



Figure 2

POWER5 die photo showing major chip components. (FXU: fixed-point execution unit; ISU: instruction sequencing unit; IDU: instruction decode unit; LSU: load/store unit; IFU: instruction fetch unit; FPU: floating-point unit; MC: memory controller.) Reprinted with permission from [2].

- IBM Power5 (server)
  - 120W@1900MHz
  - Baseline
- Intel Core2 sc (laptop) :
  - 15W@1000MHz
  - *4x more FLOPs/watt than baseline*
- IBM PPC 450 (automobiles - BG/P)
  - 0.625W@800MHz
  - 90x more
- TensilicaXTensa(Moto Razor) :
  - 0.09W@600MHz
  - 400x more

# Tensilica Xtensa

Computer on-chip
- Entièrement configurable par l'utilisateur;
- Jeu d'instructions modulable : réduire ou ajouter;
- Le compilateur est généré en même temps que le chip est configuré
- Intégré dans Eclipse

Il faut ensuite aller voir un fondeur, produire les chips,
cabler les circuits etc...

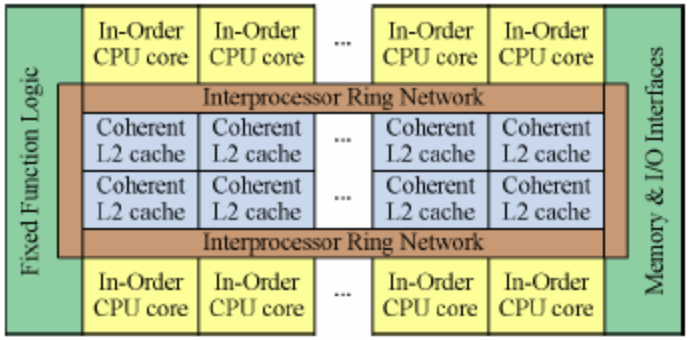
Valable si on a besoin d'un grand nombre.

Envisagé par NERSC
National Energy Research Scientific Computing Center, DOE-LBL

# Intel : Larrabee

32 coeurs
+ reseau local on-chip

= manycore




Principal objectif : high performance graphics

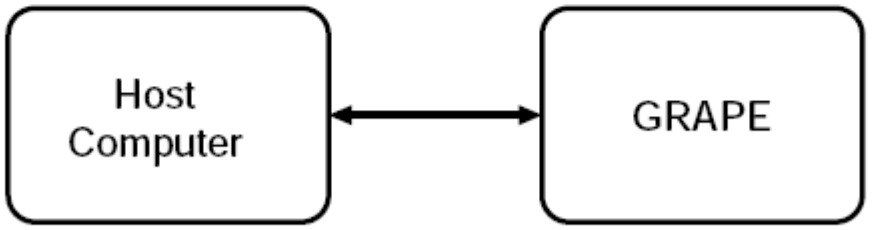
Mais jeu d'instructions X86 complet : peut faire du calcul haute performances

Status : 2009 repoussé à 2010

# Grape : un ordinateur sur mesure

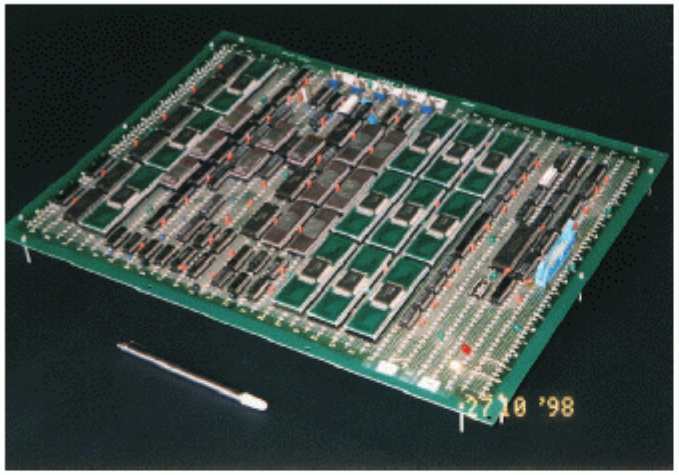
# Basic concept (As of 1988)

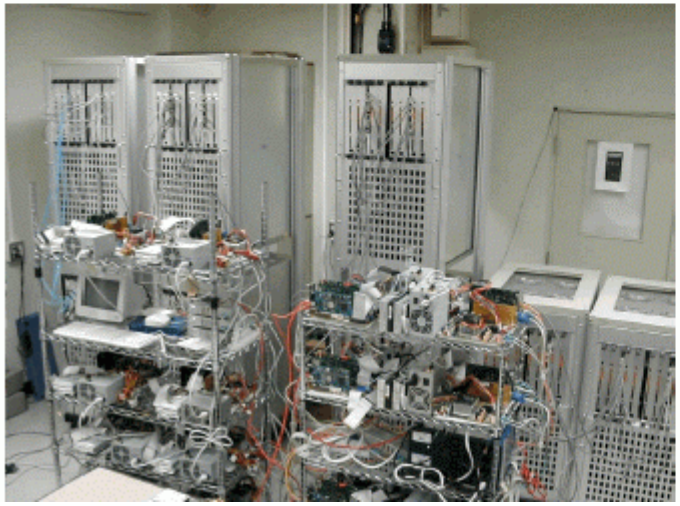
- With $N$-body simulation, almost all calculation goes to the calculation of particle-particle interaction.
- This is true even for schemes like Barnes-Hut treecode or FMM.
- A simple hardware which calculates the particle-particle interaction can accelerate overall calculation.
- Original Idea: Chikada (1988)




Accelerator-based computing two decades ago

# GRAPE-1 to GRAPE-6



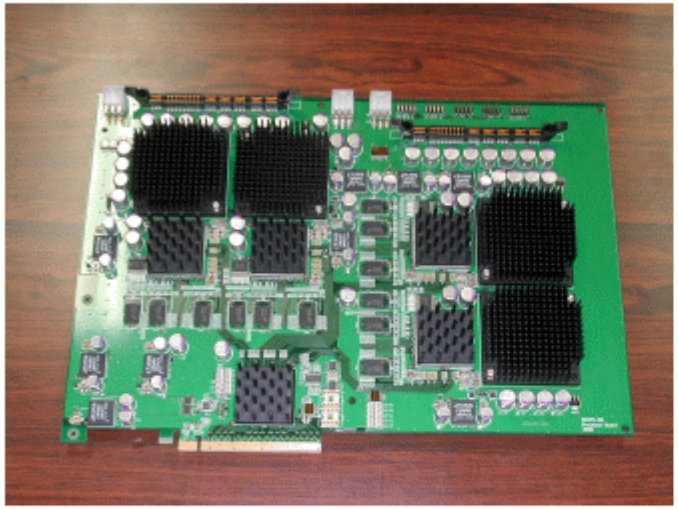


GRAPE-1: 1989, 308Mflops

GRAPE-4: 1995, 1.08Tflops

GRAPE-6: 2002, 64Tflops

# Processor board

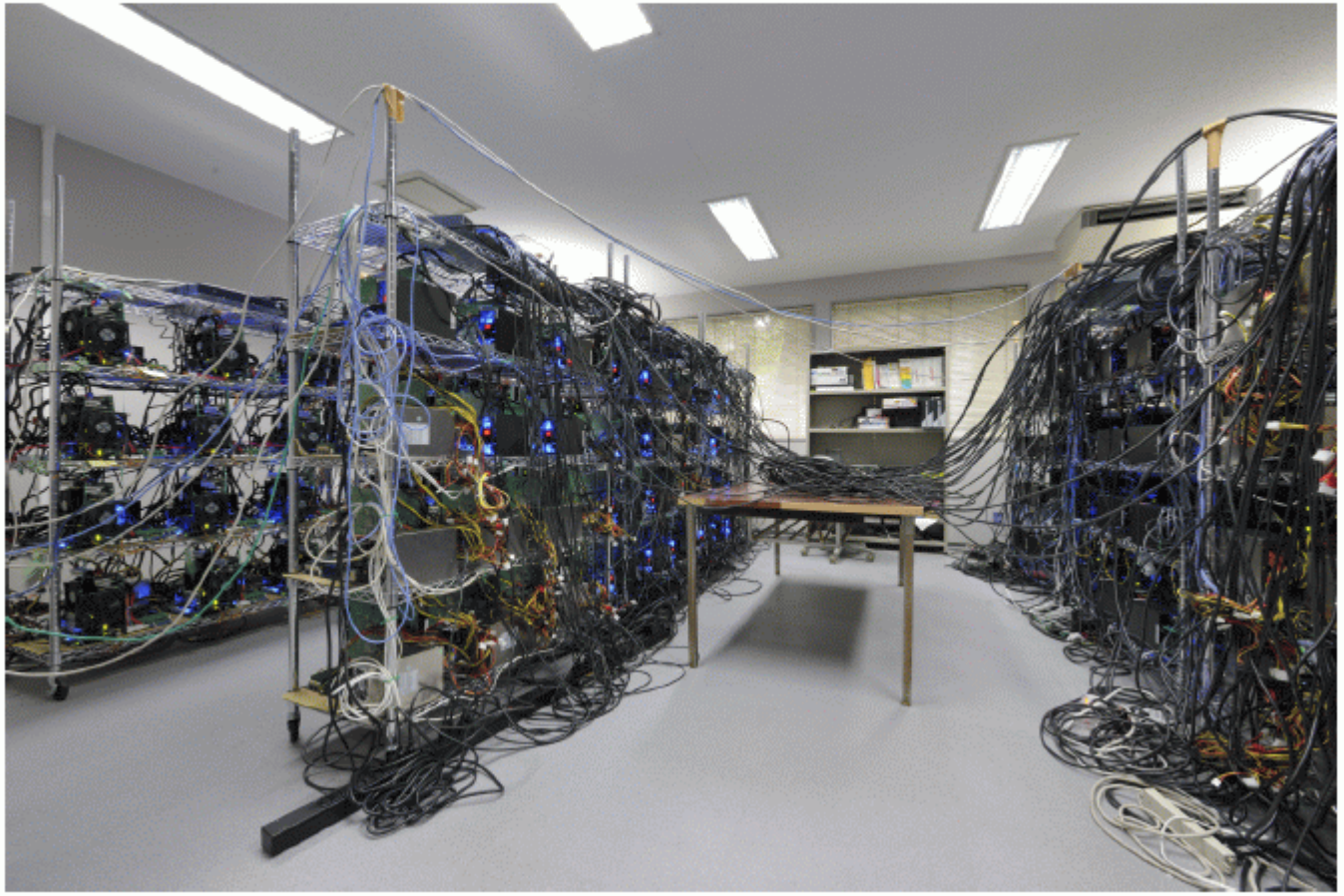


PCIe x16 (Gen 1) interface
Altera Arria GX as DRAM controller/communication interface

- Around 200W power consumption
- Not quite running at 500MHz yet... (FPGA design not optimized yet)
- 900Gflops DP peak (450MHz clock)
- Available from K&F Computing Research

# Alternatives : Processeur spécialisé

## GRAPE-DR cluster system



# "Problem" with GRAPE approach

- Chip development cost becomes too high.

| Year | Machine | Chip initial cost | process |
|---|---|---|---|
| 1992 | GRAPE-4 | 200K$ | 1$\mu$m |
| 1997 | GRAPE-6 | 1M$ | 250nm |
| 2004 | GRAPE-DR | 4M$ | 90nm |
| 2010? | GDR2? | $>$ 10M$ | 45nm? |

Initial cost should be 1/4 or less of the total budget.
How we can continue?

# GPU et GPGPU

## Graphic processing unit
## Et
## General Purpose Processing Unit

# Alternatives : GPU


**Exemple : 9270 AMD**

**Dérivé des cartes vidéo**

**2GB on board memory DDR5**
**1.2 Tflops single precision**
**240 Gflops double precision**

800 stream cores
PCIExpress x16

Deux constructeurs :
NVIDIA 66% du marché
AMD/ATI 33%

Mais combien font du calcul scientifique ?

# Alternatives : GPU

Programmation des GPU

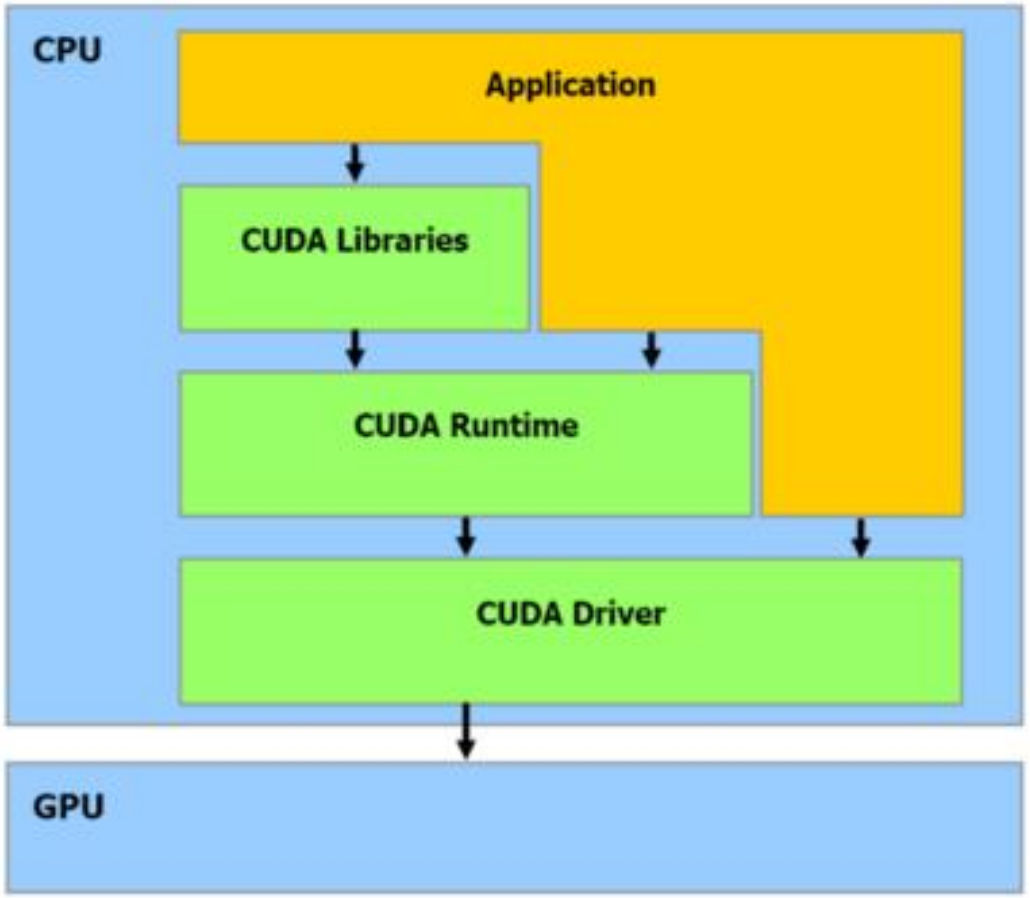
Ne fonctionne que sur certains problèmes bien spécifiques

- SIMD : Single Instruction Multiple Device, tous les processeurs font la même chose
- Ils ont aussi besoin d'accéder à la mémoire (rapidement)
  → divers niveaux de cache
- La mémoire du GPU est différente de celle du CPU, elles ne communiquent que via le but PciExpress, opérations coûteuses que le programmeur doit gérer

Environnement de programmation →

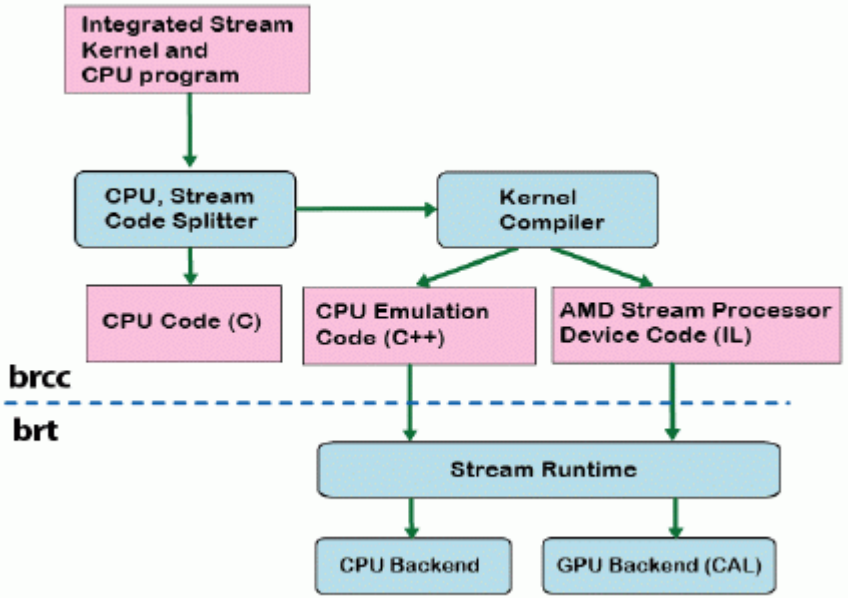
# Sdk Nvidia

Possibilité de contrôler
L'utilisation des processeurs
Et de la mémoire



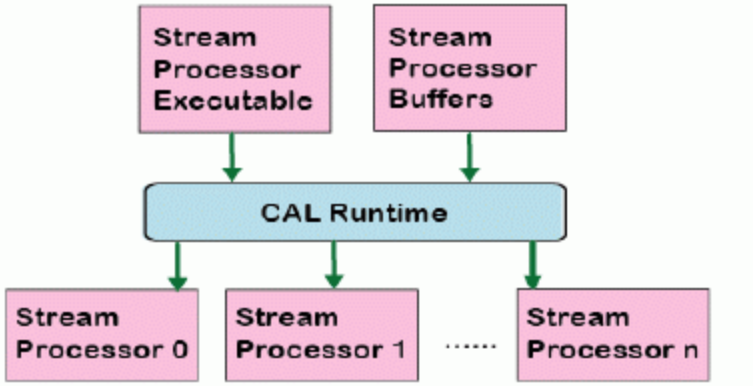
# Sdk AMD

Stream computing
Simple : 5 fonctions seulement
Fonctionalités limitées




Equivalent à CUDA driver

Nombreuses possibilités
Complexe
Doc illisible



# La reconstruction tomographique

Pourquoi avoir choisi AMD et brook+ alors que tous utlisent Nvidia et CUDA ?

A l'époque la seule option pour :

- 2GB de mémoire,
- 500 Megaflops, aujourd'hui 1.2 Teraflops
- Double precision

Extrêmement facile à programmer, important pour une démarche exploratoire

**Coup de chance, brook+ parfaitement adapté à ce problème.**
**C'est un cas particulier**
**Temps d'exécution 1.5mn contre 4h sur CPU**

# OpenCL

## Open Computing Language

## Le calcul parallèle facile...

# OpenCL : l'avenir


Open standards for parallel programming

# OpenCL : l'avenir

## Design Goals of OpenCL

- Use all computational resources in system
  - CPUs, GPUs, and other processors as peers
  - Data- and task- parallel compute model
- Efficient parallel programming model
  - Based on C99
  - Abstract the specifics of underlying hardware
- Specify accuracy of floating-point computations
- Desktop and Handheld Profiles

# OpenCL : l'avenir

Extensions à C permettant de spécifier :

- Les données qui sont indépendantes les unes des autres;
- La fonction qui s'execute sur ces données (kernel)
- La synchronisation (points de rendez-vous);
- Les éléments de hardware sur lesquelles ont veut exécuter

Le compilateur fait le reste, et prépare l'exécution

Disponibilité

Apple utilise maintenant openCL pour tout le graphique (snow leopard)

Nvidia et AMD offrent des versions beta en téléchargement gratuit

**Adapté à tout hardware : gpu, cpu, multithread ...**

Performances ? Bonnes d'après les supporters

# Alternatives : FPGA

Grandes capacités d'Entrées-sorties;
Bonnes performances pour le calcul en entier et fixed point
Pas si bon en virgule flottante

Difficile à programmer (et debugger)

Mais les choses s'améliorent avec des environnements de développement, Émulateur et debugger performant

**Il y a des utilisateurs heureux pour des applications de calcul intensif (de préférence avec flux de données important, exemple optique adaptative)**

# Quelle est la bonne option?

Il n'y a plus de solulion générale.

Si le problème est parallelisable par nature, le GPU est une bonne option

Sinon, multicore

Eventuellement le tout en cluster pour encore plus de performances

OpenCL devrait permettre de maitriser toutes ces architectures, en se chargeant
D'une partie de l'optimisation. A quel prix?
OpenCL deviendra-t-il réellement un standard?

Les fabricants de GPU cherchent à ouvrir le domaine d'utilisation :
plus de fonctionnalités, mais les performances seront affectées.

# Conclusion

C'est un domaine en évolution rapide.

Si vous avez besoin de grande puissance de calcul, pensez au GPU.

Si votre problème n'est pas aisément parallelisable, multithreading, multiprocessing

**Dans tous les cas, openCL est votre ami**

**Ne pas oublier les fpga, surtout si vous êtes I/O limited et si vous pouvez eviter Les calculs en float**