La fin de la Loi de Moore et les applications HPC

François Bodin,
Université de Rennes 1 / Irisa

UPMC 19 janvier 2017

Introduction

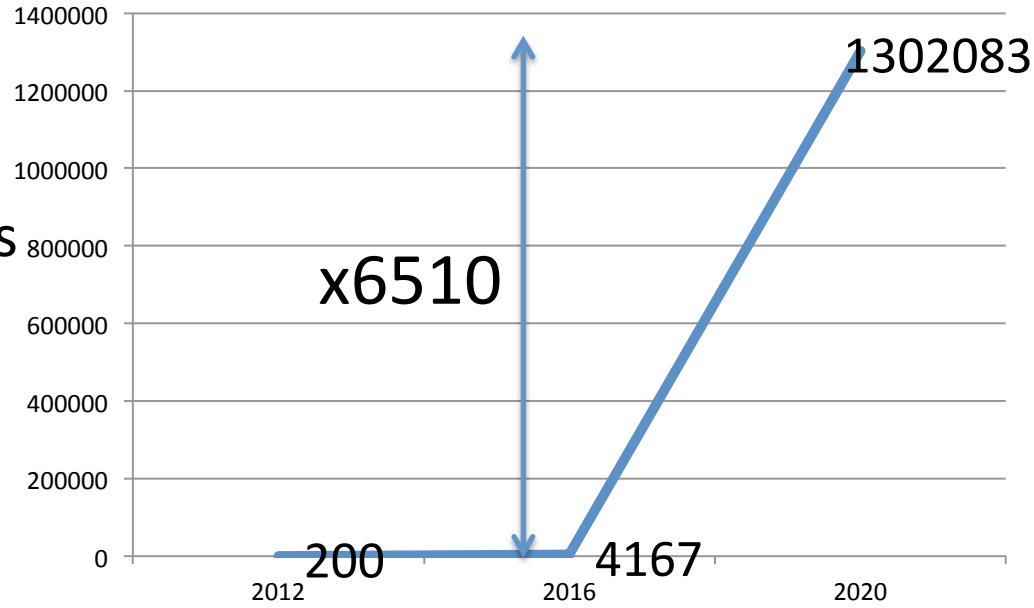
- En 1965, Gordon Moore prédit que le nombre de transistors sur une puce doublerait chaque année
  - En 1975, Gordon Moore revoit cette prévision en indiquant une période de 2 ans
- Depuis plusieurs décennies les progrès (exponentiels) du HPC sont fondés, à parts égales, sur
  - La loi de Moore, les progrès des microarchitectures (pipelining, exécution OoO, prédictions de branchements, hiérarchie mémoire), le parallélisme
  - Les progrès algorithmiques
- La loi de Moore est une loi essentiellement économique dont la remise en cause va bouleverser les équilibres actuels
  - Fin du gain de performance transparent pour le logiciel
  - Symptôme d'une forte évolution technologique

Les « Lois »

- Les lois qui s'essoufflent
  - La loi de Moore : densité des transistors
  - La loi de Dennard (1974): densité énergétique constante
  - La loi de Kryder : observe que la densité de stockage des disques magnétique s'est accrue plus rapidement que la densité des puces
- Les lois qui perdurent
  - La loi de Rock : le coût des fonderies qui double tous les 4 ans
  - La loi Amdahl sur l'accélération
  - La loi de Gustafson sur le « *weak scaling* »

Les contraintes et autres « Murs »

- Le mur de l'énergie, dominé par les mouvements de données
  - Processeur massivement multi-cœurs
  - Technologie accélératrice (e.g., GPU, FPGA)
- Le mur de la bande passante et latence mémoire
- Le déluge de données
- La diminution du MTBF
- Contraintes économiques
  - Efficacité des codes (i.e. passage à l'échelle)
  - Temps de développement (*Time to solution*)



Climate Earth System Modeling
Data produced in total in Gbytes/month-of-simulation
PRACE Scientific for High Performance Computing in Europe

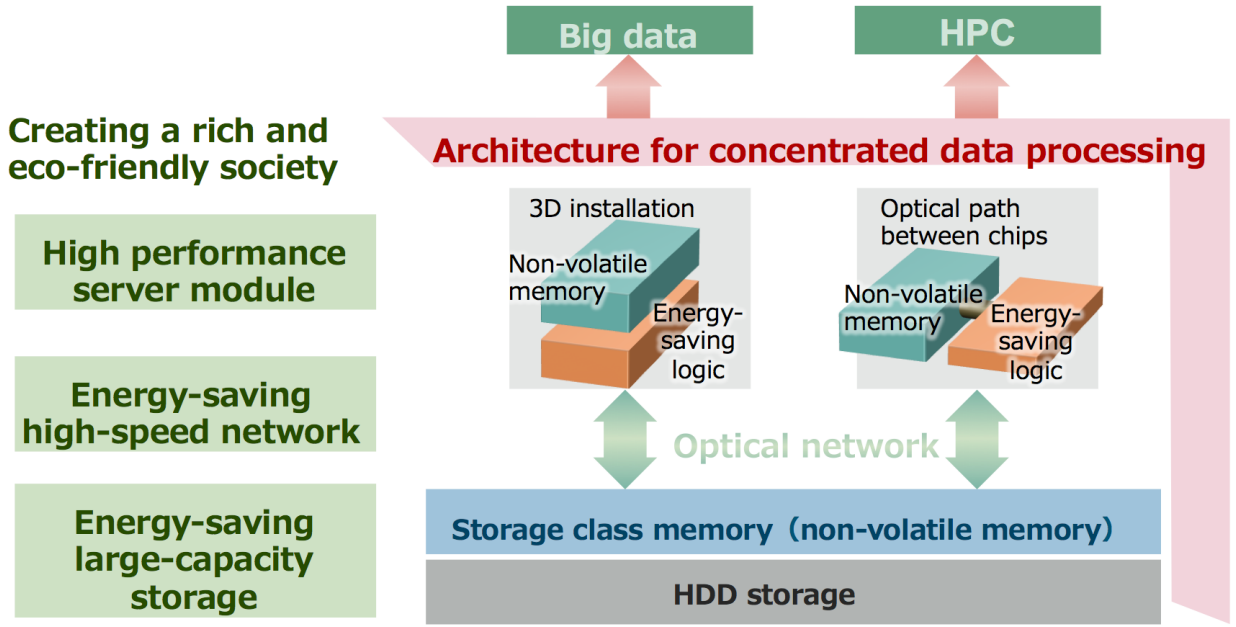
Plan de la présentation

- Survol des évolutions technologiques probables
- Les conséquences du point de vue logiciel
- Les obstacles à l'adaptation / exploitation des nouvelles technologies
- A propos de la convergence calculs-données

LES ÉVOLUTIONS TECHNOLOGIQUES

Evolutions technologiques

- + de cœurs de calcul, hétérogénéité
- Communications photoniques silicium
- 3D *stacking*
- Nouvelles mémoires non volatiles



From **Japanese HPC, Network, Cloud & Big Data Ecosystem circa 2015 onto Post-Moore**
Satoshi Matsuoka Professor Global Scientific Information and Computing (GSIC) Center Tokyo Institute of Technology
Visiting Prof-NII, Visiting Researcher-Riken AICS Fellow, Association for Computing Machinery (ACM) & ISC
BDEC Barcelona Presentation 2015 01 28

Stockage et Entrées/Sorties

- De plus en plus sur le chemin critique et de plus en plus chères[1]
  - Le ratio « E-S / calculs » se dégrade
- Interfaces E/S actuelles inadaptées[2]
  - Manque d'efficacité et de flexibilité
  - Besoin de plus d'intelligence dans la gestion des E/S
  - *Checkpoint-restart* spécifiques à l'application
- Généralisation des mémoires non volatile (NVRAM) [3]

39e Forum ORAP
28 mars 2017

[1] « The technology drivers are tending towards infinitely cheap computing and infinitely expensive data systems! » Bryan Lawrence, NCAS, STFC & The University of Reading

[2] ETP4HPC Strategic Research Agenda achieving HPC leadership in Europe

[3] H. Kryder and Chang Soo Kim, After Hard Drives—What Comes Next? Mark IEEE Transactions on Magnetics, Vol. 45, No. 10, October 2009

Spécialisation des CPU et Hétérogénéité

- « *Customization* » du silicium[1]
  - Performance spécifique à un type de code (tout en conservant un jeu d'instructions généraliste)
  - Economie d'énergie (i.e. petits vs gros cœurs)
- Adaptation de la vitesse des cœurs en fonction de la charge
  - Gestion DVFS de plus en plus agressif : peu de threads à fréquence élevée, compromis parallélisme complexe
  - Vitesse de crête d'un CPU difficile à discerner[2]
- Et autres accélérateurs
  - GPU, FPGA, MPPA, …

[1] Ravi Rajwar Martin Dixon Ronak Singhal "Specialized Evolution of the General-Purpose CPU", 7th Biennial Conference on Innovative Data Systems Research (CIDR '15)

[2] Theoretical Peak FLOPS per instruction set on modern Intel CPUs, Romain Dolbeau

CONSÉQUENCES

Quelques conséquences[1]

- Problème de passage à l'échelle des codes
- Spécialisation des APIs / programmation par domaine (DSL)
- Analyse de données in-situ
- Virtualisation des ressources
- Evolution du modèle économique d'exploitation des supercalculateurs CPU.Heure en Watt.Heure

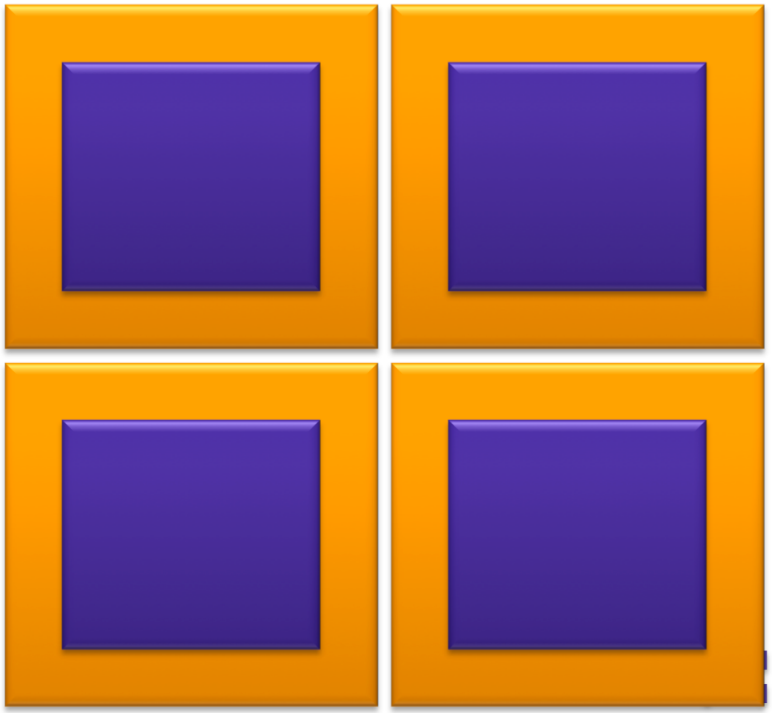
[1] fortement inspiré des recommandations EESI

Evolution des codes

- Nouveaux algorithmes *ultra-scalables*
  - Parallélisation par échanges de messages inadaptée aux multi-cœurs
  - Besoin d'être capable d'exploiter l'hétérogénéité des architectures (multi-cœurs, accélérateurs)
  - Augmentation de l'importance des aspects *runtime*
- Programmation applicative *hétérogène*
  - Analyses de données (e.g., Java, R, etc.)
  - Simulations numérique (e.g., Fortran, C++)
  - Mélange de bibliothèques et autres API / langages
  - Interactivité, visualisation des données

Multicoeurs et décomposition de domaine

32x32x32 cell domain
ghost cells 2
ghost cells / domain cells = **0.42**



16x16x16 cell domain
ghost cells 2
ghost cells / domain cells = **0.95**

1 process → 8 processes

Modèle SPMD inadapté
→ MPI + X

Spécialisation des APIs / Langages

- Langages et APIs spécifiques à un domaine (DSL)
  - Ne pas exposer la complexité, haute productivité[1]
  - Organiser les interactions informatiques / applicatifs
  - Une difficulté intrinsèque avec leur modèle économique
- Langages dynamiques (e.g. Pyhton, Julia, etc.)
  - Tant qu'a faire un saut technologique !
- Nouvelles architectures logicielles
  - Flexibilité pour s'adapter à un large panel d'architectures
  - Bien séparer les aspects implémentations des aspects applicatifs
  - Exploitation des accélérateurs, multi/many-cœurs
  - Nouvelles hiérarchies mémoires (e.g. mémoires persistantes)

[1] Cherry Pancake, « There's an inherent tension between portability and performance » In « Usable HPC -- An Oxymoron? »

Analyse de données in-situ

- Impossible à l'avenir de sortir l'ensemble des données des machines pour analyse a posteriori
  - Processus de découverte à clarifier
  - Analyse *on-the-fly*
  - Visualisation, HPC interactif
- Introduction de problématique de type *Big-Data* dans le HPC
  - Algorithmiques et structures de données différentes
- *Workflows* complexes[1]
  - Orchestration / chorégraphie des traitements numériques, E/S et gestion et exploitation des données
- NVRAM + interconnexions photoniques rendent possibles les analyses *in-situ*

[1]Applications and Workload Management Systems, Ewa Deelman, USC, BDEC 2015

HPC vs Big Data

	HPC	Big-Data
compute on	double x;	int n; int *p;
huge //ism	definitely	why not
memory wall	high&thick	= or >
lower layer	SIMD+wide L/S	(SIMD+)wide L/S
code	for() {...}	while() {...if...if...}
data	A[i][j][k++]	p->q->r; p=p->s
SIMD friendly	yes hopefully	no *in general*

Extrait de BDEC 2015, **Finding Regularity in Irregularities**

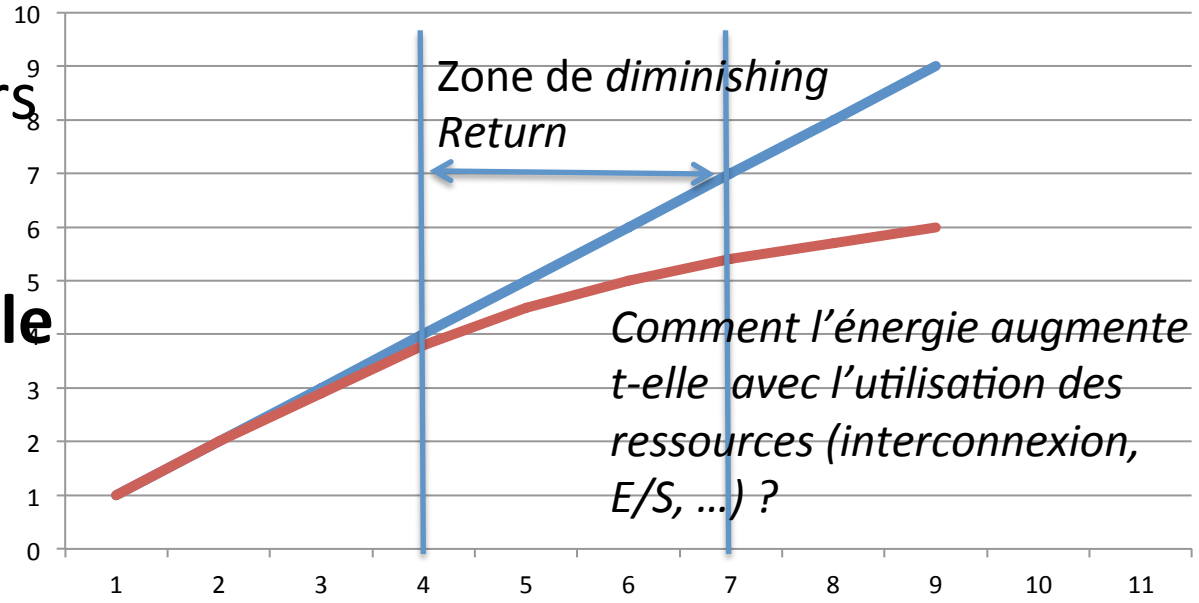
Virtualisation des ressources

- Evolutions nécessaires des piles logicielles et gestion de ressources pour accommoder les nouveaux besoins sur les nœuds
  - Gestion et exploitation des données
  - Entrées / Sorties, visualisation, résilience (Exascale)
  - Gestion de workflow avec des tâches hétérogènes
- Exemples
  - Containers[1] (e.g. *TSUBAME 3.0*)
  - Python, Java etc. pour l'analyse et la gestion de workflow

[1] Virtualisation légère fondée sur l'isolation pour cloisonner les applications, e.g. Docker

Changement de modèle économique CPU.Heure → Watt.Heure

- La mesure Watt.Heure n'est pas seulement limitée à la partie calcul, inclue l'interconnexion, les E/S, etc.
- Difficultés à incarner la métrique dans le processus de développement
- Besoin d'outils pour un retour aux développeurs et utilisateurs
- Centrée sur l'efficacité **globale** des codes
- Evolution à long-terme



Courbe d'accélération typique

LES OBSTACLES

Obstacles

- Culturels
- Economiques
- Transitionnels

Obstacles culturels

- Maîtrise des algorithmes numériques actuels
  - Validation / compréhension des nouveaux algorithmes complexes
- Pratiques de développement / mises en œuvre
  - Manque d'usage de techniques issues du génie logiciel
  - Fortran → programmation objets, langages dynamiques
  - Echanges de messages vs graphes de tâches dynamiques (schémas de parallélisation hybrides)
- Interdisciplinarité
  - Science des données et science des calculs
- Organisationnel
  - Gestion de l'allocation de ressources de calcul etc.
  - Introduction d'une tierce partie → maintenance et support

Obstacles économiques

- Difficultés à estimer le retour sur investissement
  - Pour des investissements conséquents
  - Efficacité de l'usage des calculateurs
- Déterminer les invariants sur lesquels concevoir un plan de développement et/ou migration
  - Abstractions plus stables que les technologies
  - Algorithmes qui doivent exhiber du parallélisme sous diverses formes (tâches, SIMD/T, données)
  - Standards ? Bibliothèques ?
  - Quelles architectures ?

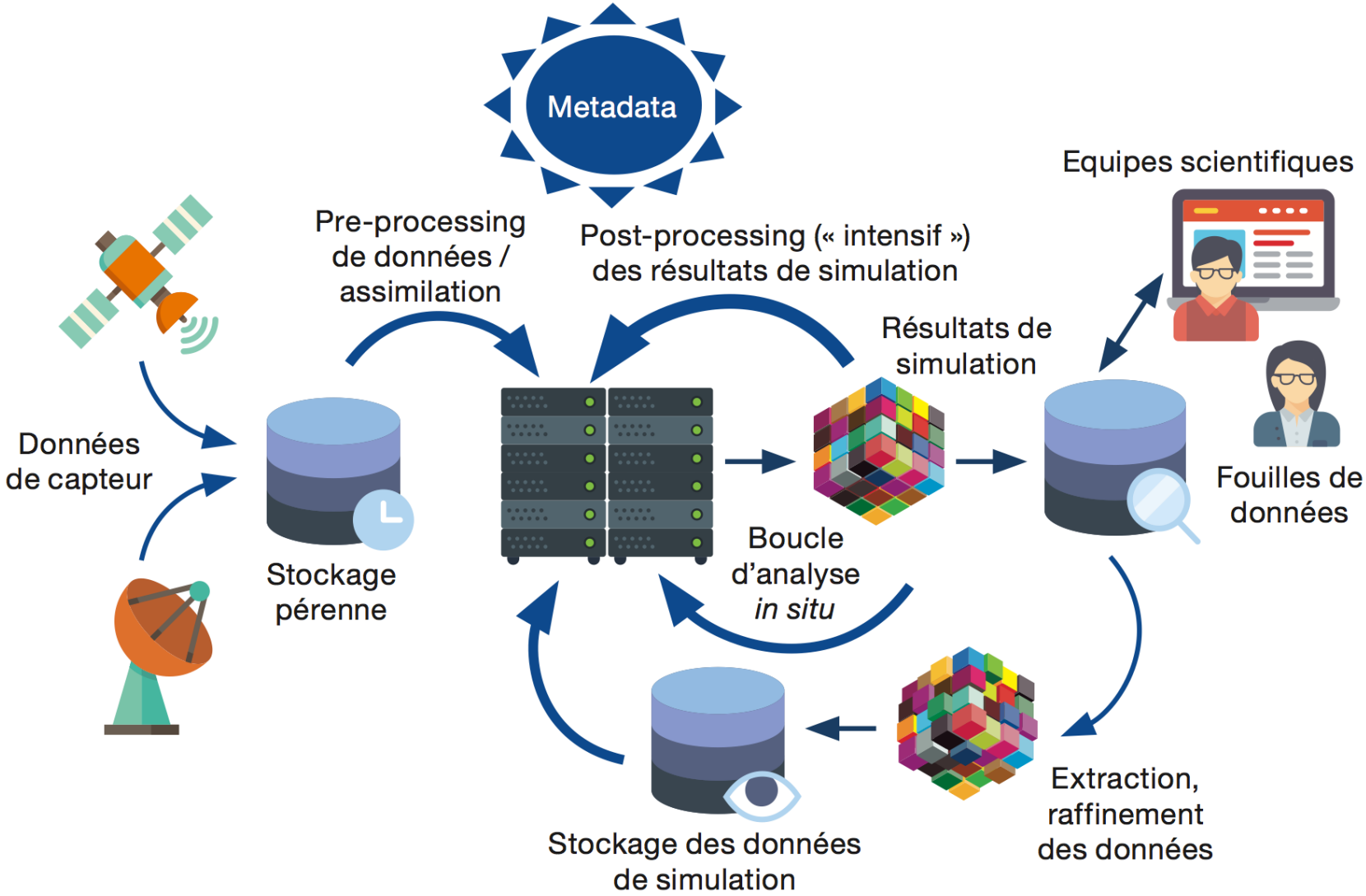
Obstacles transitionnels

- Productions vs nouveaux codes
  - Comment préserver l'exploitation
    - Beaucoup de codes *legacy* ne passeront pas à l'échelle
  - Nouveaux codes en compétition avec les anciens
    - Quels efforts ? Validation ?
    - Coût de transition élevé
  - Comment mettre en place des approches par co-design
- Organisation, support et formation des équipes[1]
  - Plus d'interdisciplinarité
  - Exploitation de l'écosystème (e.g. CoE)

[1] Gordon Bell : « Training and applications will still be the dominant limiters to massive parallelism », Frontiers of Massively Parallel Computation '92

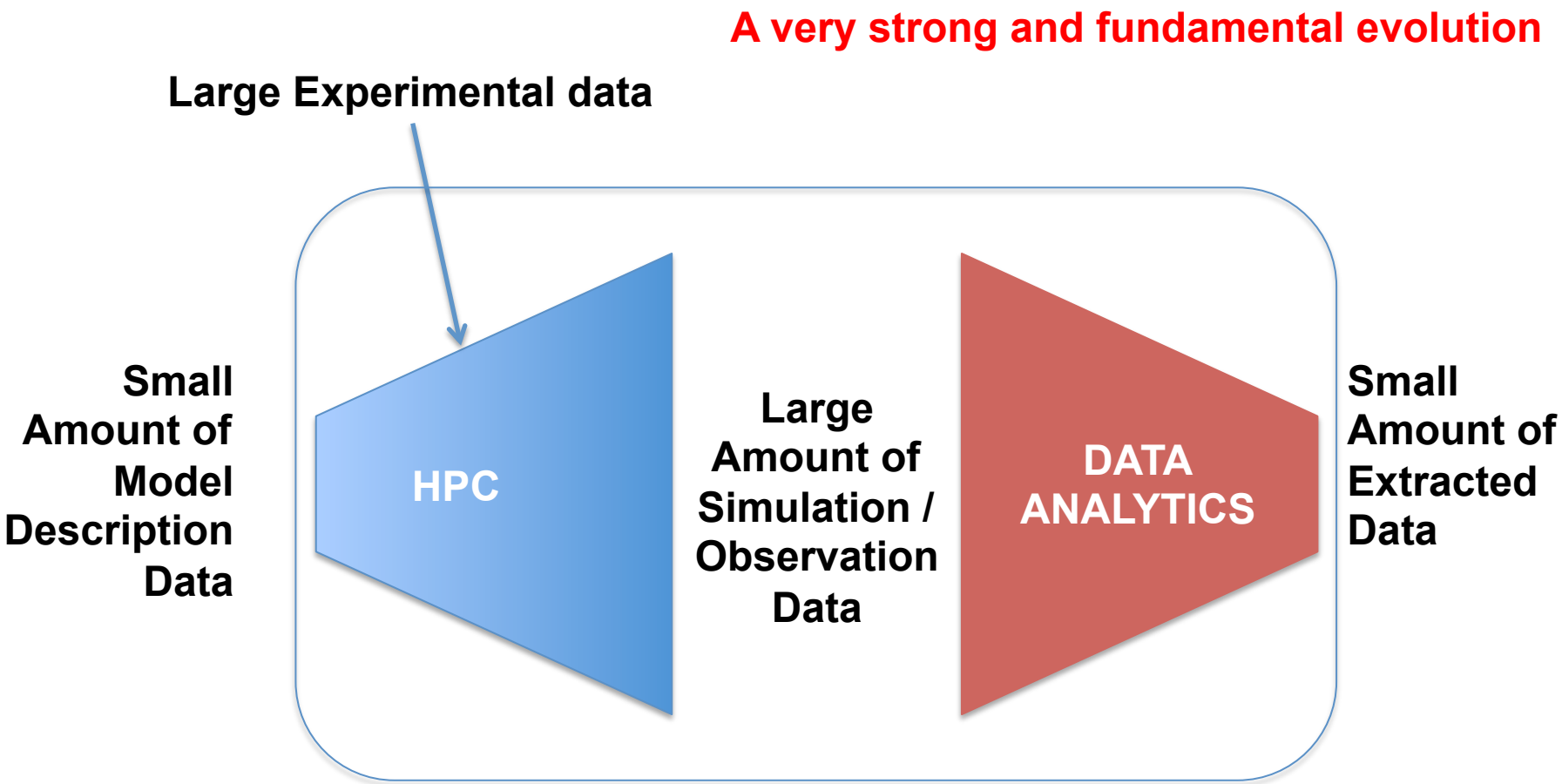
A PROPOS DE LA CONVERGENCE CALCULS-DONNÉES

Cycle de vie des données dans une application scientifique



Extrait de *« Les big data à découvert »,* CNRS Éditions, à paraître

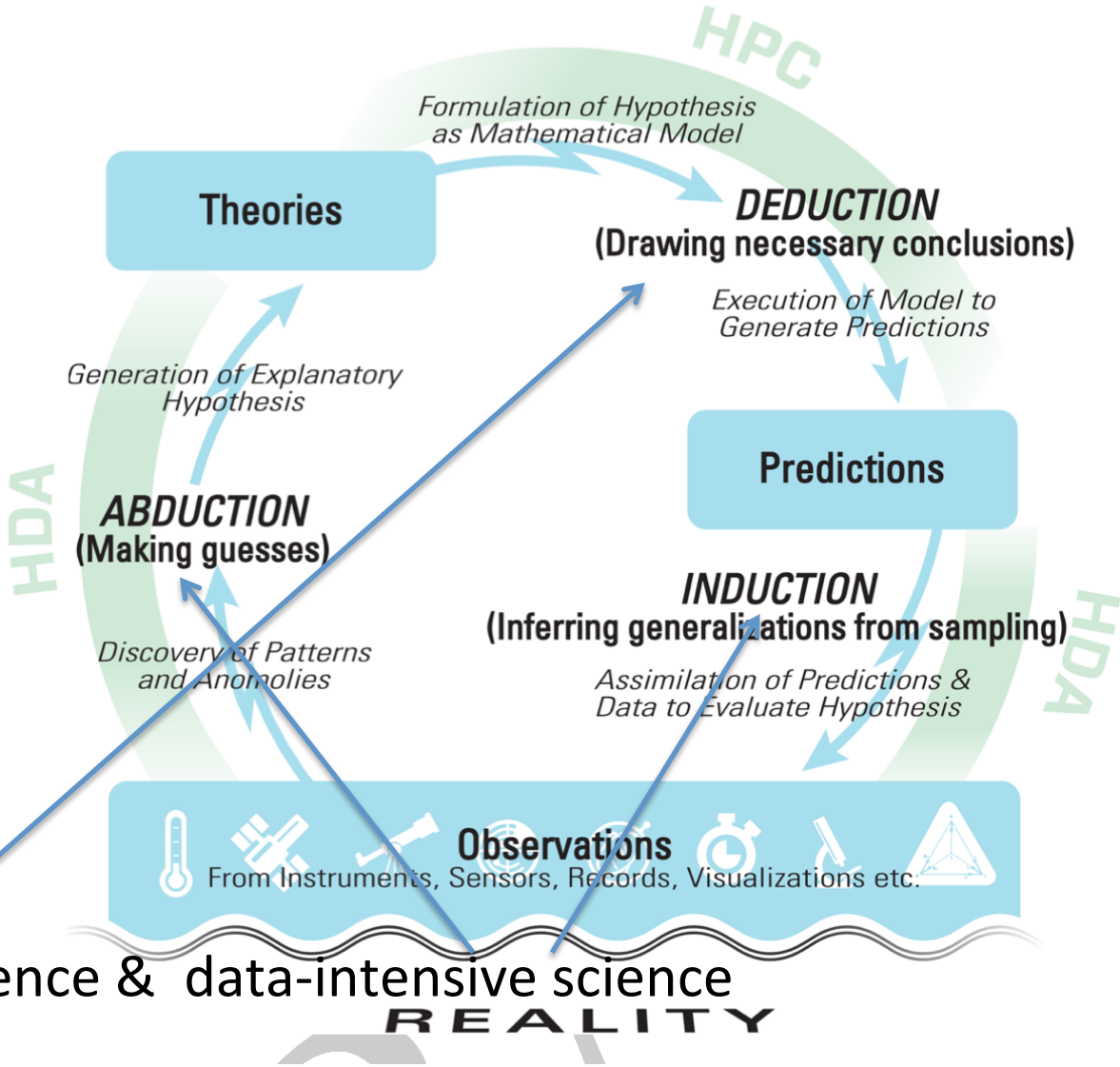
HPC and Data Analytics Convergence



A converged scientific process

From the BDEC "Pathways to Convergence" Report
BDEC Committee
November 16, 2016

This model as a plausible hypothesis discovery for physical laws.



Combines computational science & data-intensive science

Some consequences on cyberinfrastructure convergence

- Application workflows are more complex and more heterogeneous
- Compute intensive data exploration is fundamental
  - Need to invest in tools for data analysis, data mining, machine learning and data visualization
- Keeping data tractable may strongly influence systems organizations
  - Small cluster can analyze data where it was collected (edge computing)
  - Raw data could be transferred a supercomputing center for analysis.

Conclusion

- La fin de la Loi de Moore va imposer une transition profonde du HPC
  - Liée aussi à l'apparition de nouvelles technologies
- De nombreux codes HPC vont devoir évoluer
  - Scalabilité accrue et qui prenne en compte l'hétérogénéité des architectures
  - Nouvelles architectures logicielles
  - Elargissement du champs algorithmique avec l'ajout de l'analyse des données in-situ, interactivité
  - Changement des équilibres architecturaux : E/S, NVRAM, photonique, DVFS
  - Les architectures restent des cibles mouvantes
- Le cycle de découverte scientifique est amené à évoluer fortement

Conseils de lecture

- Conférence BDEC : http://www.exascale.org
- Projet EXDCI: http://www.exdci.eu
- ETP4HPC Agenda stratégique : http://www.etp4hpc.eu/strategy/strategic-research-agenda
- PRACE Scientific Case for HPC: http://www.prace-ri.eu/prace-the-scientific-case-for-hpc/
- Conférence de Bruno Bachimont, « De l'épistémologie de la mesure à celle de la donnée » https://www.youtube.com/watch?v=XcvIGRc4IvA,