

Proposal from Theory Group

ALGORITHMICS

Algorithm Theory Research

for Massively Hetero. Parallel Computing

algorithmics \Leftrightarrow mathematics

topic (1) with Fukunaga, Kishimoto, Kobayashi

topic (2) with Okamoto, Onsjö

東京工業大学 グローバルCOEのご紹介

計算世界観の深化と展開

拠点目標

教育: 計算世界観の手法を実践できる人材の育成

研究: 計算を中心とした科学の新たな科学の確立

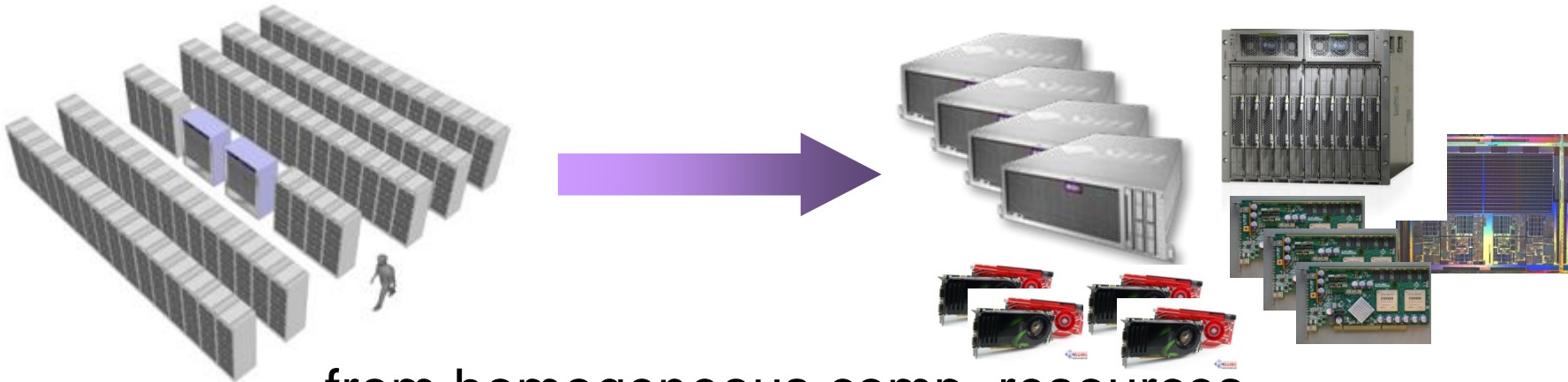
計算世界観とは

計算を中心に科学の対象を見よう！
それにより新しい科学の手法を導こう！

計算 = 処理・判断の組み合わせ

fundamental

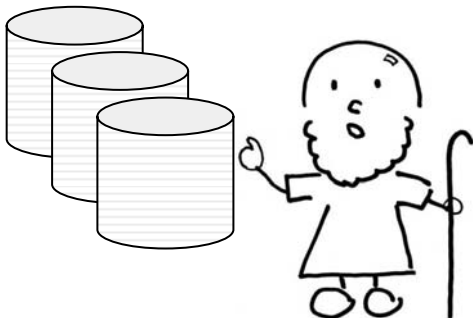
Motivation#1. Paradigm shift in HPC



from homogeneous comp. resources
to massively hetero. comp. resources

Motivation#2. Heavy & hetero.

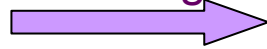
computational demands



Targets and Goals

1. How to handle mass. hetero parallelism

ultimate goal



Autonomic Computing

- run time self error detection → self-healing
- self status recognition → self-configuration

2. How to make use of mass. parallelism



comp.
model

ultimate goal

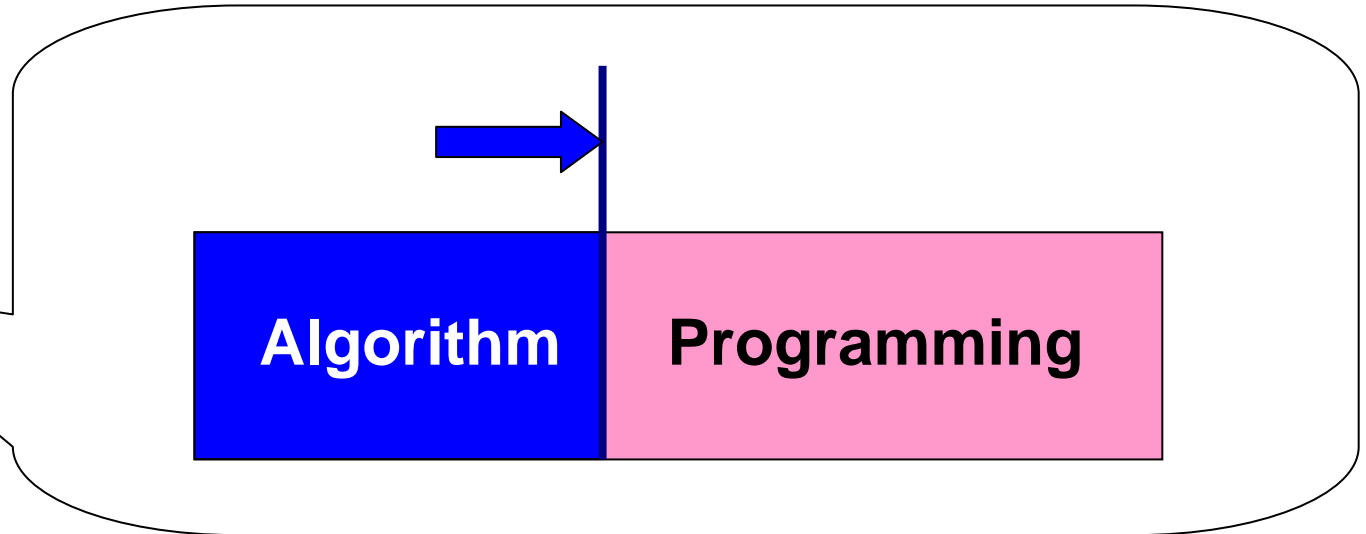


Opt. Parallel Computing

- (1) revisit of theoretical model: BSP, LogP
- (2) new model for new computing, e.g., GPU

New Algorithmic Research for Opt. Parallel Computing

Why new models ??



- (1) Revisit of theoretical model: BSP, LogP
- (2) New model for new computing, e.g., GPU

Topic (1) Revisit of theoretical model

parallel comp. model

S
How to make use of
HUGE memory space ??

of proc.s p

2. unbalanced

Target impact comparable to
Google's MapReduce

1. **cost underestimate** → unrealistic
model cost \ll actual cost

← task must be
distributed evenly

2. **difficult to design** requirement

New Algorithmic Framework

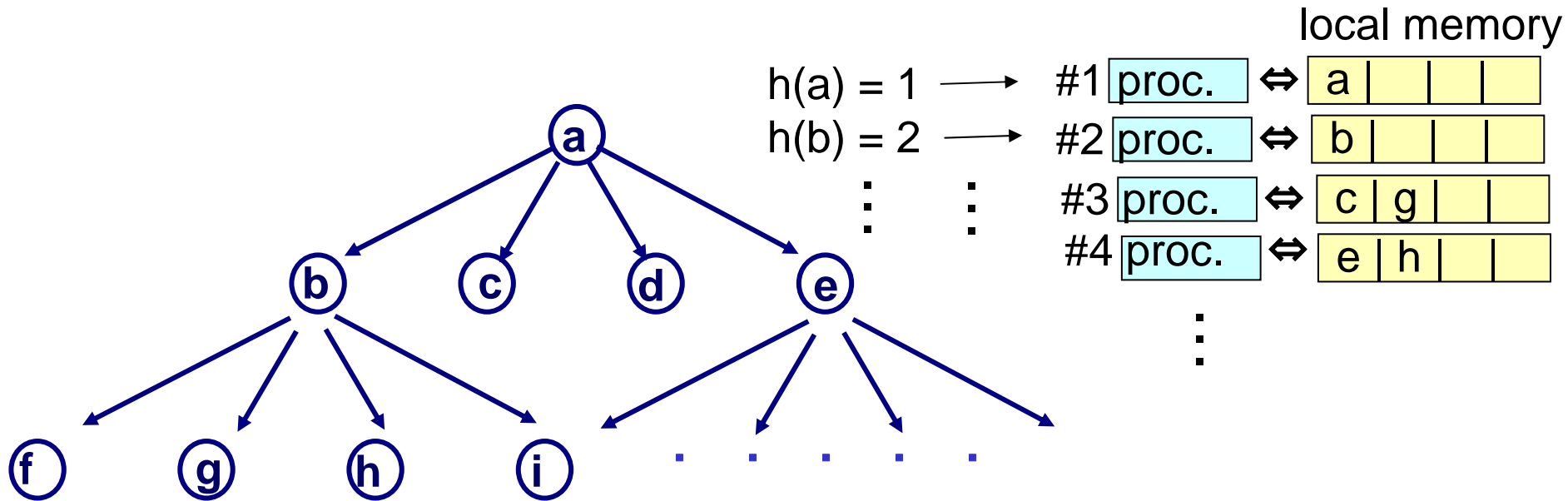
**Hash Distributed
Asynchronous Computing**

Extend

Hash Distributed Computing

New Generic Parallel Algorithm for Dynamic Prog. best paper award

- asynchronous parallelism
- use huge (distributed) memory space



gen. rec. dag

Topic (2) New model for new computing

GPU computing

my understanding
(based on Testa)

Points of interest:

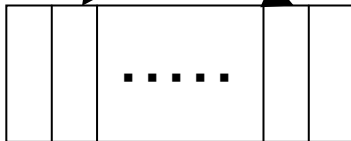
- + very fast local memory (cache)
- + **local parallelism by local proc.s**
- + **stream processing (?)**
- **bank conflict \Rightarrow slow down**

global memory

access cost T_{ext} \leftrightarrow block size B

shared memory (cache)

shared memory size M



local processors

warp size

of processors W

access cost T_{sh}

to the shared memory

no bank conflict case