XcalableMP

HPCS研究室
佐藤グループ(PA-team)の研究紹介

佐藤
筑波大学　計算科学研究センター

自己紹介

- http://www.hpcs.cs.tsukuba.ac.jp/~msato/top.html.ja
- 専門分野
  - 並列処理、並列システムの性能評価
  - 並列プログラミング言語、コンパイラ、OpenMP
  - グリッドコンピューティング、広域分散処理
  - 並列処理による低電力・省電力化
- 筑波大学　計算科学研究センター　センター長
- 理研計算科学研究機構 プログラミング環境研究チーム・チームリーダ
  「京コンピュータ」の研究開発
- 公開ソフトウエア
  - Omni OpenMP/ XcalableMP コンパイラ http://www.xcalablemp.org/
  - OmniRPCシステム　http://www.omni.hpcc.jp/omniRPC/

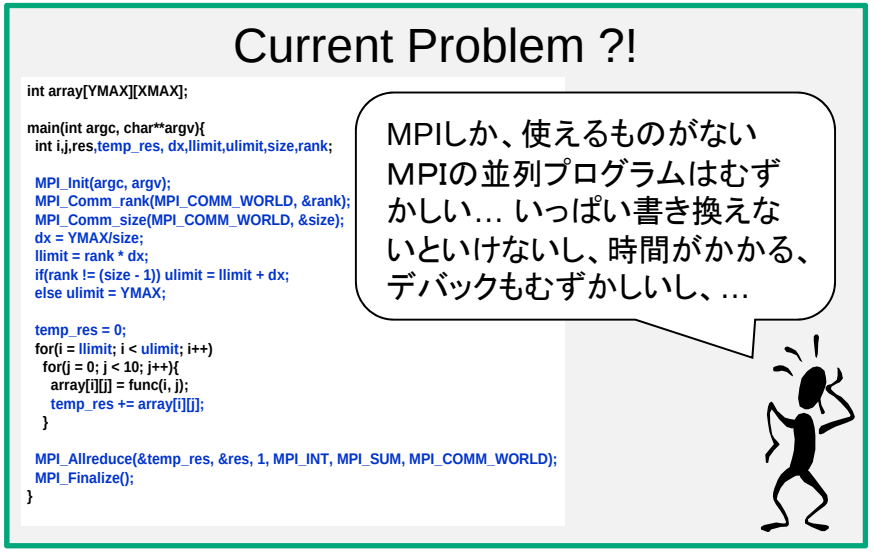
- 担当講義
  - プログラミング入門II（情報科学類）
  - プログラミング言語処理（情報科学類）
  - 分散システム（情報科学類）
  - プログラミング環境特論（システム情報工学研究科）

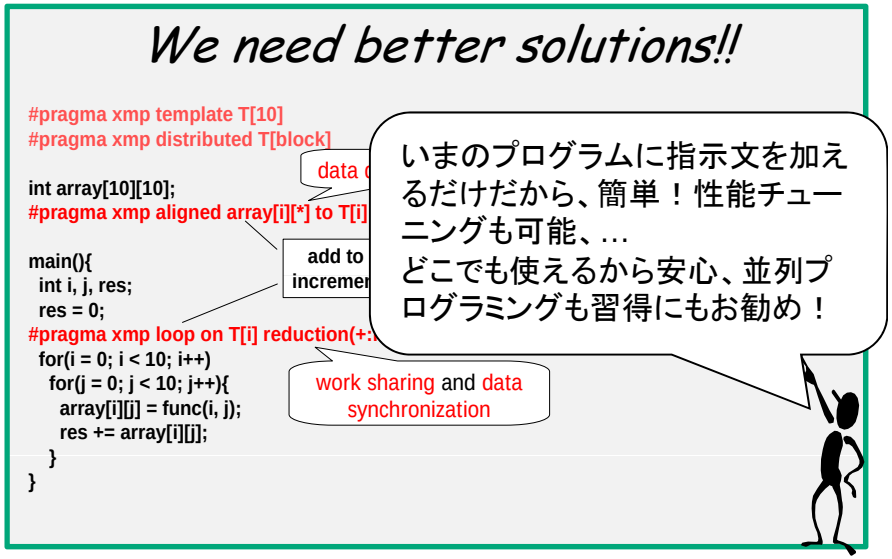
卒業　研究テーマ

- 並列プログラミング言語XcalableMPをベースにした研究
  - コンパイラ
  - 大規模並列 – 性能プロファイル
  - GPU/メニーコア　ー　HA-PACS
  - 耐故障　ー　言語からのアプローチ
- マルチコア・メニーコアの並列プログラミングに関する研究
  - 細粒度並列処理技術 Cilkとか。
  - Work stealing scehduler ...
- 省電力化に関する研究
  - なんでも。

XcalableMP プロジェクト

- 現状と課題
  - 並列プログラムの大半はMPI通信ライブラリによるプログラミング
    - 生産性が悪く、並列化のためのコストが高い。
  - 並列プログラミングの教育のための簡便で標準的な言語がない（MPIでの教育にとどまっている）
  - 研究室のPCクラスタから、センター、ペタコンまでに到るスケーラブルかつポータブルな並列プログラミング言語が求められている
- 目標
  - 既存言語を指示文により拡張し、これからの大規模並列システム（分散メモリシステムと共有メモリノード）でのプログラミングを助け、生産性を向上させる並列プログラミング言語を設計・開発する。
  - 標準化をすることを前提に、ユーザのわかりやすさを第一にどこでも使えるということを重視し、開発ならびに普及活動を進める。

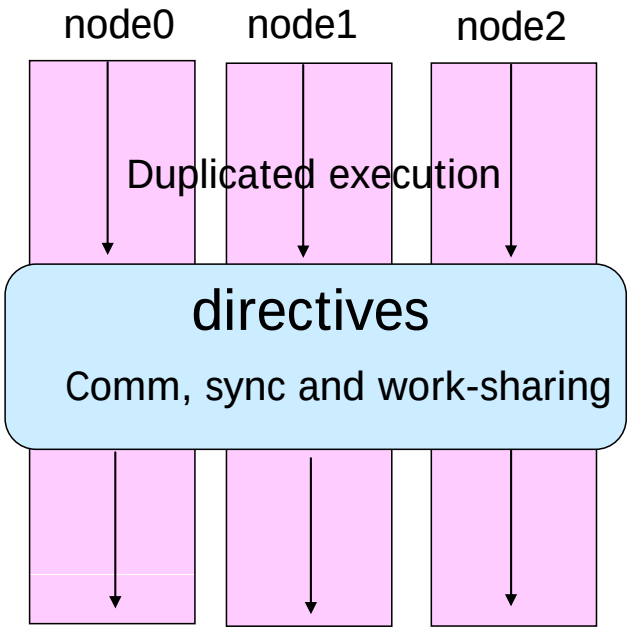




XcalableMP http://www.xcalablemp.org XcalableMP

XcalableMP : directive-based language eXtension
for Scalable and performance-tunable Parallel Programming

- **Directive-based language extensions** for familiar languages F90/C/C++
  - コードの書き換えや教育のコストを抑えること
- **“Scalable” for Distributed Memory Programming**
  - SPMDが基本的な実行モデル。
  - MPIのように、各ノードでスレッドが独立に実行を開始する。
  - 指示文(directive)がなければ、重複実行
  - タスク並列のためのMIMD実行も



- **“performance tunable” for explicit communication and synchronization.**
  - 指示文を実行するときに、Work-sharing や通信・同期がおきる。
  - すべての同期・通信操作は、指示文によって起きる。HPFと異なり、パフォーマンスのチューニングがわかりやすくなる。

Code Example

```
int array[YMAX][XMAX];

#pragma xmp nodes p(4)
#pragma xmp template t(YMAX)
#pragma xmp distribute t(BLOCK) on p
#pragma xmp align array[i][*] to t(i)

main(){
  int i, j, res;
  res = 0;

#pragma xmp loop on t[i]  reduction(+:res)
  for(i = 0; i < 10; i++)
    for(j = 0; j < 10; j++){
      array[i][j] = func(i, j);
      res += array[i][j];
    }
}
```

data distribution

add to the serial code : incremental parallelization

work sharing and data synchronization

XcalableMP コード例 (NPB CG, global view)

```
#pragma xmp nodes p[NPROCS]
#pragma xmp template t[N]
#pragma xmp distributed t[block] on p
...
#pragma xmp aligned [i] to t[i] :: x,z,p,q,r,w
#pragma xmp shadow [*] :: x,z,p,q,r,w
...
```

ノードの形状の定義

Templateの定義とデータ分散を定義

データの分散は、templateにalignデータの同期のためのshadowを定義、この場合はfull shadow

Work sharing ループの分散

データの同期

```
/* code fragment from conj_grad in NPB CG */
sum = 0.0;
#pragma xmp loop on t[j] reduction(+:sum)
    for (j = 1; j <= lastcol-firstcol+1; j++) {
        sum = sum + r[j]*r[j];
    }
    rho = sum;
for (cgit = 1; cgit <= cgitmax; cgit++) {
#pragma xmp reflect p
#pragma xmp loop on t[j]
        for (j = 1; j <= lastrow-firstrow+1; j++) {
            sum = 0.0;
            for (k = rowstr[j]; k <= rowstr[j+1]-1; k+
                sum = sum + a[k]*p[colidx[k]];
            }
            w[j] = sum;
        }
#pragma xmp loop on t[j]
        for (j = 1; j <= lastcol-firstcol+1; j++) {
            q[j] = w[j];
        }
```

HPCCベンチマークのプログラミングと性能

- HPC Challenge Benchmark Class2
  - 新しい並列プログラミング言語での記述性と性能を競うカテゴリ
    - Class1はシステム性能
  - 4つのベンチマーク
    - STREAM
    - Random Access
    - HPL
    - FFT
  - 今年は、Awardは、性能はIBM(X10 and UPC), 記述性はCary (Chapel) になった

SC09 HPCC Class2 でFinalist!

Nicknamed the "K computer"

Kei (京) represents the numerical unit of 10 Peta (10^{16}) in the Japanese language, representing the system's performance goal of 10 Petaflops. The Chinese character 京 can also be used to mean " a large gateway" so it could also be associated with the concept of a new gateway to computational science.

一、	十、	百、	千、	万、	億、	兆、	京、	垓、	杼、	穰、	溝、	澗、	正、	載、	極、
10^{0}	10^{1}	10^{2}	10^{3}	10^{4}	10^{8}	10^{12}	10^{16}	10^{20}	10^{24}	10^{28}	10^{32}	10^{36}	10^{40}	10^{44}	10^{48}

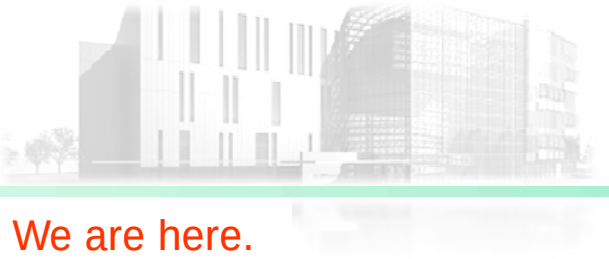
恒河沙、	阿僧祇、	那由他、	不可思議、	無量大数
10^{52}	10^{56}	10^{60}	10^{64}	10^{68}

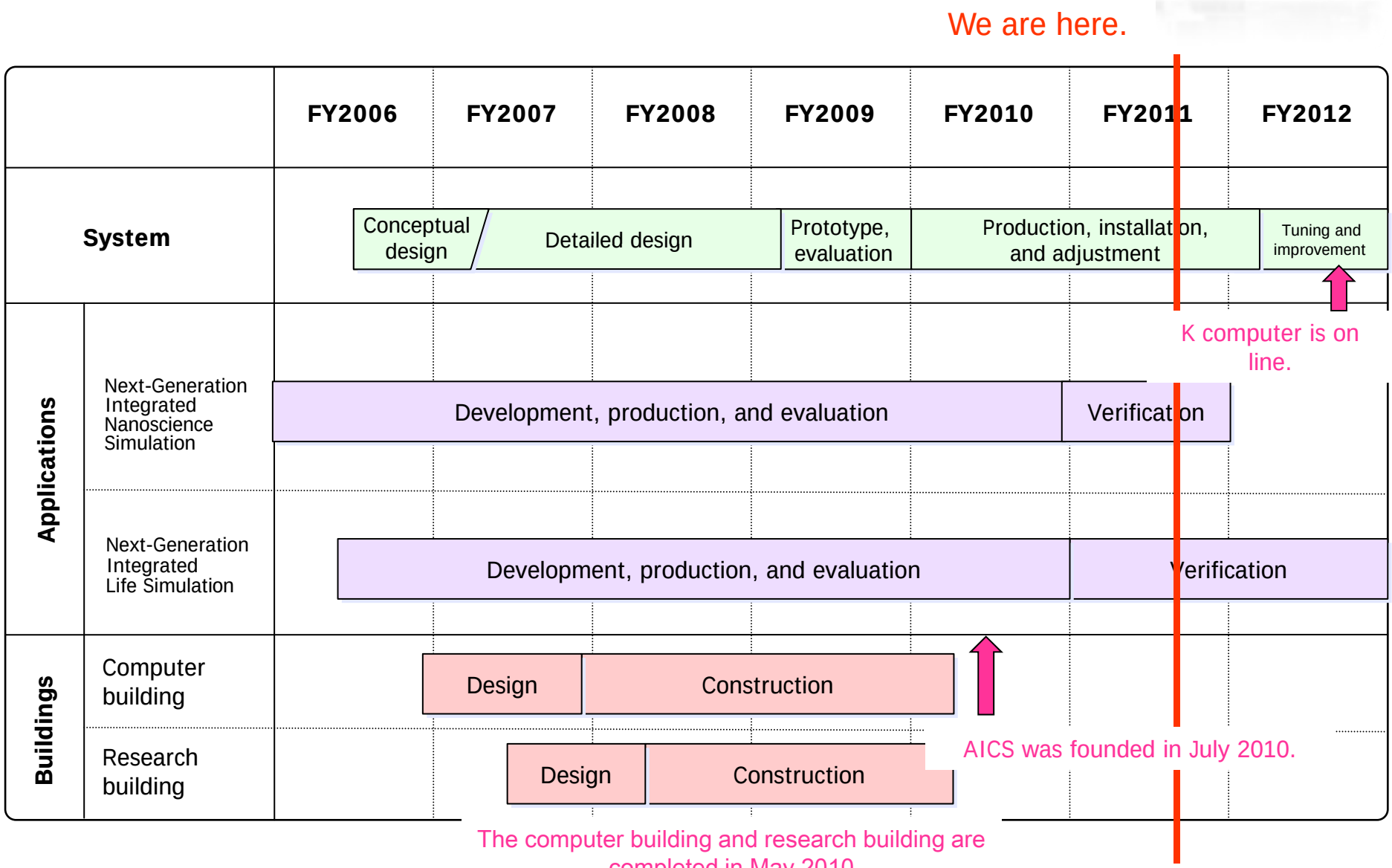
K computer Delivery Began in Late September 2010

- The first eight racks of the K computer were delivered to Kobe from Fujitsu on September 28, 2010. More than 800 racks are required for a 10 Peta Flops Performance.
- A computer rack weighs about 1,300 kg in average. The rack contains 96 water-cooled Fujitsu SPARC64 VIIIfx CPU chips, each of which performs 128 GFlops, interconnected with the 3D Torus network that Fujitsu named Tofu.

Photo of First

Schedule of development





RIKEN
Advanced Institute for Computational Science (AICS)

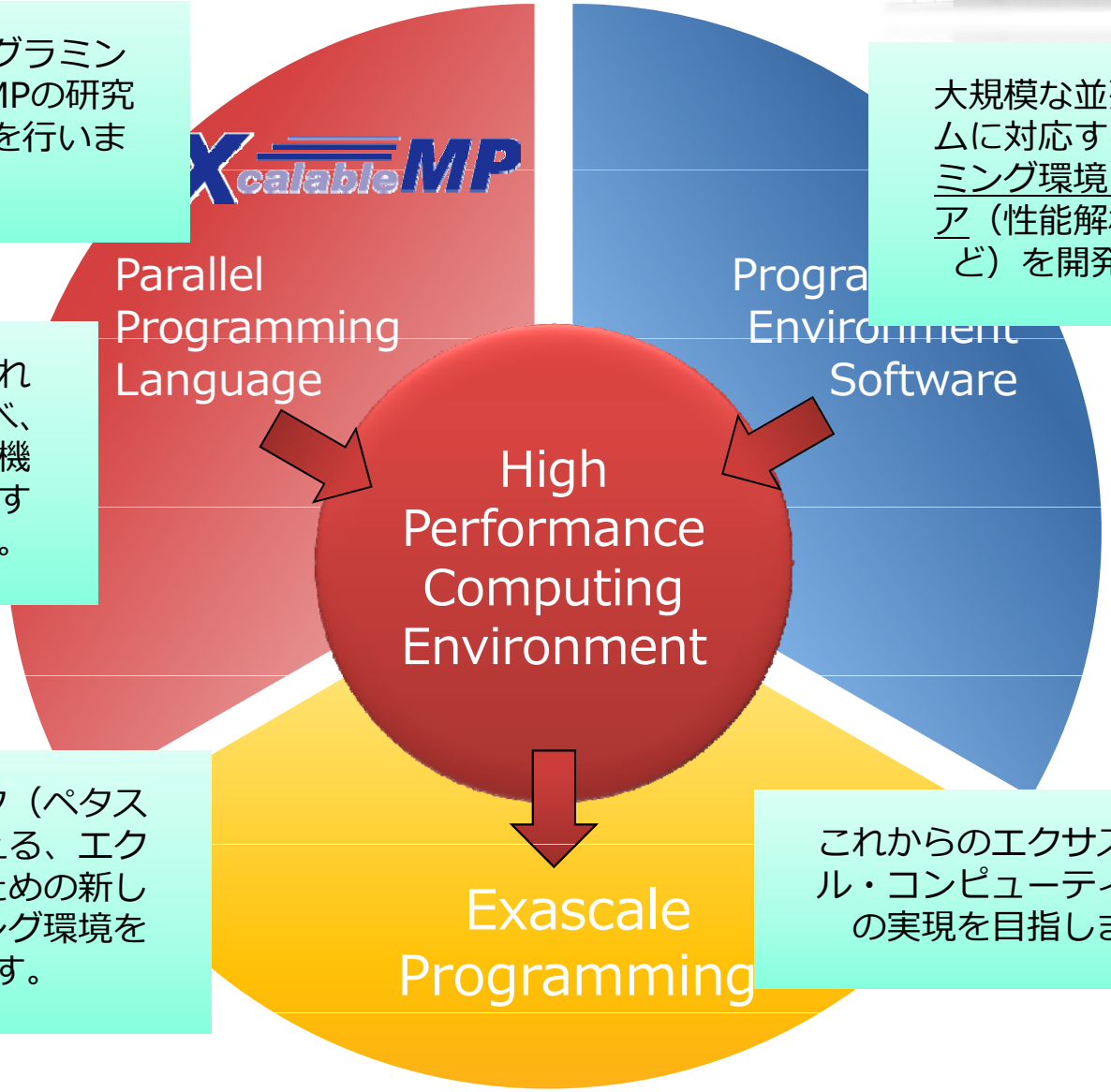
- The institute have been established at the NGS in Kobe (started in October 2010)
- Missions
  - To run the K computer efficiently for users of wide research areas
  - Carry out the leading edge of computational science technologies and contribute for COE of computational science in Japan
  - Propose the future directions of HPC in Japan and conduct it.
- Topics
  - Promoting strong collaborations between computational and computer scientists, working with core-organizations of each fields together.
  - Fostering young scientists who exploit both computational and computer science
  - Research for new concepts for HPC in the future beyond the NGS (this is, exascale?)

AICS RIKEN Advanced Ins e

http://www.aics.riken.jp/

プログラミング環境研究チームの研究概要

新しい並列プログラミング言語XcalableMPの研究開発・普及活動を行います。



大規模な並列プログラムに対応するプログラミング環境ソフトウェア（性能解析ツールなど）を開発します。

現在広く用いられている方法に比べ、簡単に並列計算機の性能を引き出すことができます。

京コンピュータ（ペタスケール）を超える、エクサスケールのための新しいプログラミング環境を研究します。

これからのエクサスケール・コンピューティングの実現を目指します

卒業　研究テーマ

- 並列プログラミング言語XcalableMPをベースにした研究
  - コンパイラ
  - 大規模並列 – 性能プロファイル
  - GPU/メニーコア　ー　HA-PACS
  - 耐故障　ー　言語からのアプローチ
  - フランスとの共同研究（フランスに行きたいひと！）
- マルチコア・メニーコアの並列プログラミングに関する研究
  - 細粒度並列処理技術 Cilkとか。
  - Work stealing scehduler ...
- 省電力化に関する研究
  - 応、相談