

# ゆとりのためのHPC

筑波大学 大学院 システム情報工学研究科  
李 珍泌 (イー ジンピル)

# 自己紹介

- 氏名 : 李 珍泌 (イー ジンピル)
- 所属 : 筑波大学 HPCS研究室 博士課程2年次
- XcalableMP処理系の実装
  - C, Fortran + XcalableMP directive
  - 理想 : 分散メモリ + マルチコア +  $\alpha$  in XcalableMP

```
#pragma xmp template t(0:(N - 1))  
#pragma xmp distribute t(block) on p  
#pragma xmp align array[i] with t(i)  
  
#pragma xmp loop on t(i)  
for (int i = 0; i < N; i++) {  
    array[i] = ... ;  
}
```

# きっかけ

- あるSNSにて
  - 今更C言語なんて低レベルな言語に(略  
→ なんだまたゆとりか
- 研究室内の言語シェアの変動
  - C言語からC++
  - シェルスクリプトからRuby
  - 今から勉強するなら?: Java, Ruby, C++
- webで大人気: Haskell, Go, . . .

# プログラミング言語 (HPC)

- OpenMP            C, C++, Fortran
  - MPI                C, C++, Fortran
  - CUDA              C, C++, Fortran
- 
- C, C++, Fortran以外は？
  - 実質CとFortranしか選択肢がないのでは？
    - C++ではどうか？
      - OpenMP, Pthreads: Cのインターフェイスと同様

# MPI

## ■ MPI\_Reduce in C

- `int MPI_Reduce(void* sendbuf, void* recvbuf, int count, MPI_Datatype datatype, MPI_Op op, int root, MPI_Comm comm);`

## ■ MPI\_Reduce in C++

- `void MPI::Comm::Reduce(const void* sendbuf, void* recvbuf, int count, const MPI::Datatype& datatype, const MPI::Op& op, int root);`

→ MPI C++ binding: 必要最小限の拡張

---

# CUDA

- Fortran

- PGI CUDA Fortranなど

- C++

- C言語の関数として呼び出す

```
extern "C" void cudaKernel(...) { // in CUDA src
```

```
int cppFunc() { // in C++ src
```

```
...
```

```
cudaKernel(...);
```

# 他には(モデルケース)

- boost::thread, mpi
  - BoostによるAPI wrapping

```
void printFunc(int i) { cout << "input: " << i; }  
  
boost::thread printThread(boost::bind(printFunc, i));  
  
printThread.join();
```

- CUDA for ...
  - language bindings: Python, Java, .NET, Ruby
  - ML for GPU (SWoPP2010)

---

# Fortran

- もちろん重要
- CSの学生にはあまり人気がない
- ~ Fortran 95
  - こちらがメジャー？
- Fortran 2003 ~
  - Cとの連動、オブジェクト指向など、現代的



# つまり、

- プログラミング言語の偏り
- もう少し選択肢が増えてもいいのでは
  - CSの学生が慣れ親しんだ環境でHPCしたい！
  - 並列プログラミング環境の整備
- もうひとつ、
  - 教科書が欲しいです