

PSI-SIM : System Performance Evaluation Environment for Next-Generation Supercomputers

Inoue, Koji

Faculty of Information Science and Electrical Engineering, Kyushu University

Shibamura, Hidetomo

Institute of Systems & Information Technologies/KYUSHU

Susukita, Ryutaro

Fukuoka Industry, Science & Technology Foundation

Inadomi, Yuichi

Research Institute for Information Technology, Kyushu University

他

<https://hdl.handle.net/2324/9180>

出版情報 : SLRC プレゼンテーション, 2007-11-10. 九州大学システムLSI研究センター
バージョン :
権利関係 :





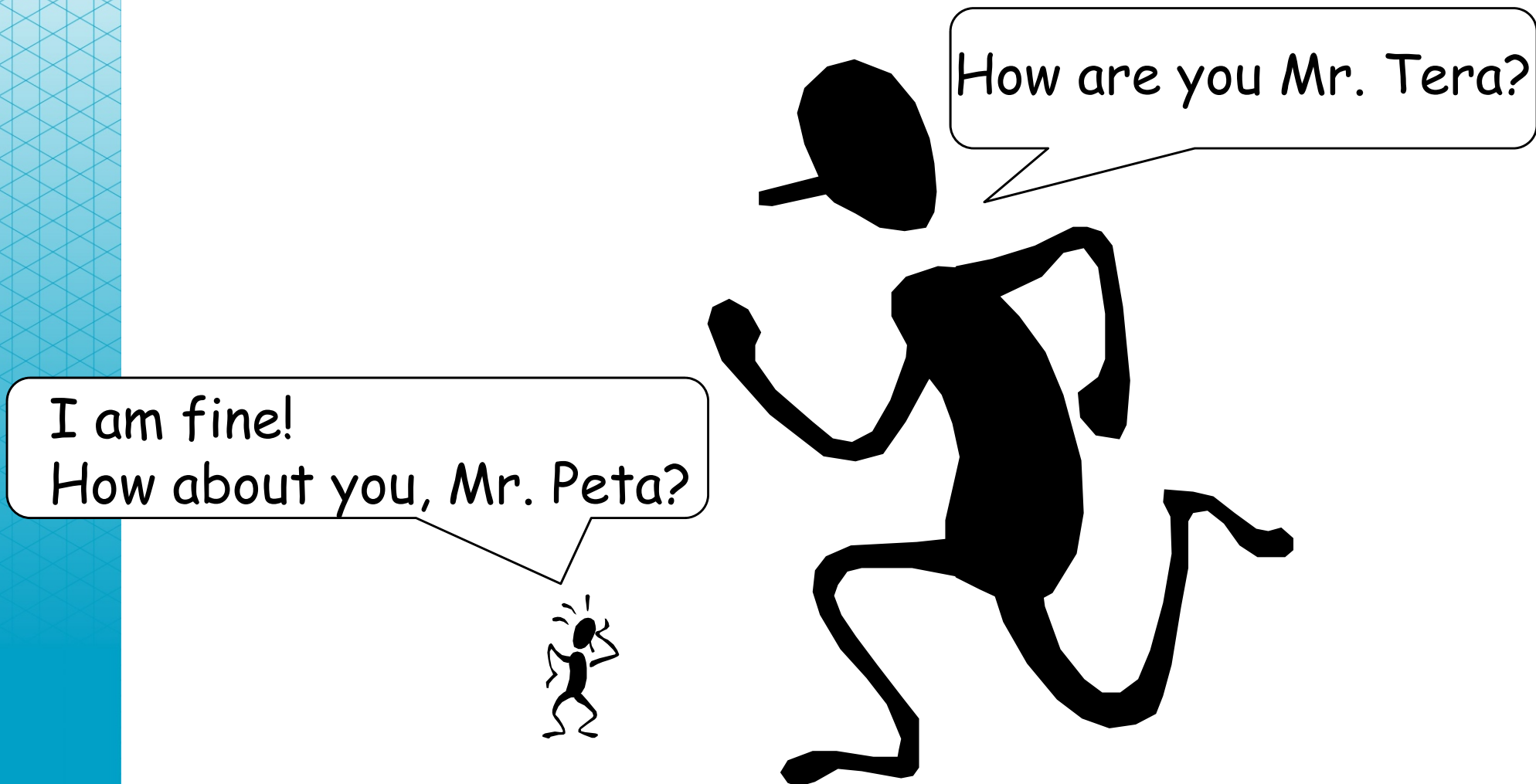
PSI-SIM: System Performance Evaluation Environment for Next-Generation Supercomputers

K. Inoue, H. Shibamura, R. Susukita, Y.
Inadomi, H. Honda, Y. Yu, and M. Aoyagi

Kyusyu University, ISIT, IST

Background

- "Peta" is tremendous!
 - Compared with "Giga or Tera" scale machines



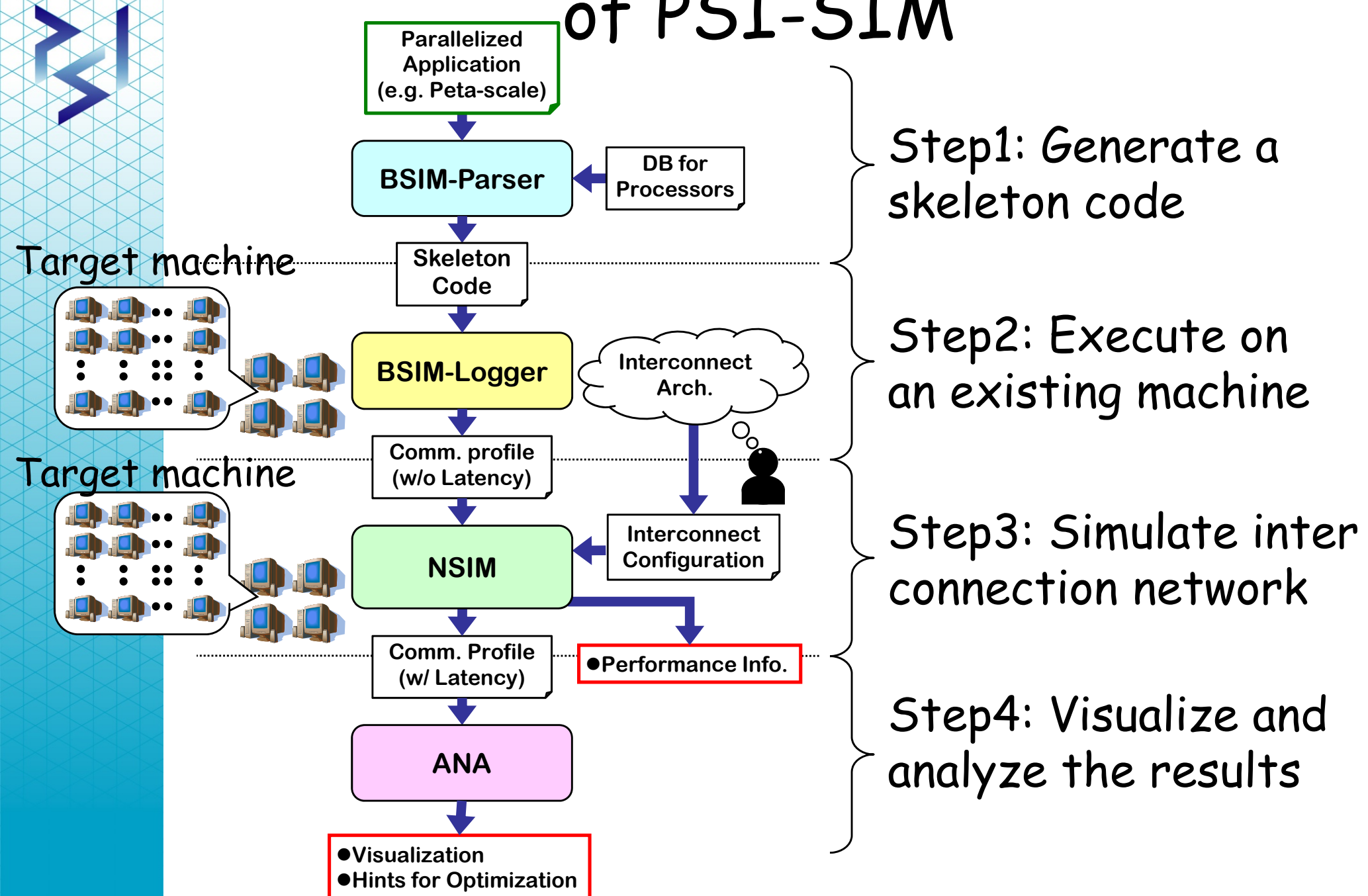
Background

- “Peta” is tremendous!
 - Compared with “Giga or Tera” scale machines
- If you would like to develop a “Peta-Scale” supercomputer, it is required to...
 - Explore the design space both of computation nodes and inter-connection network!
 - Verify the effective performance to be achieved!
- So, we need a performance evaluation environment for peta-scale supercomputers!

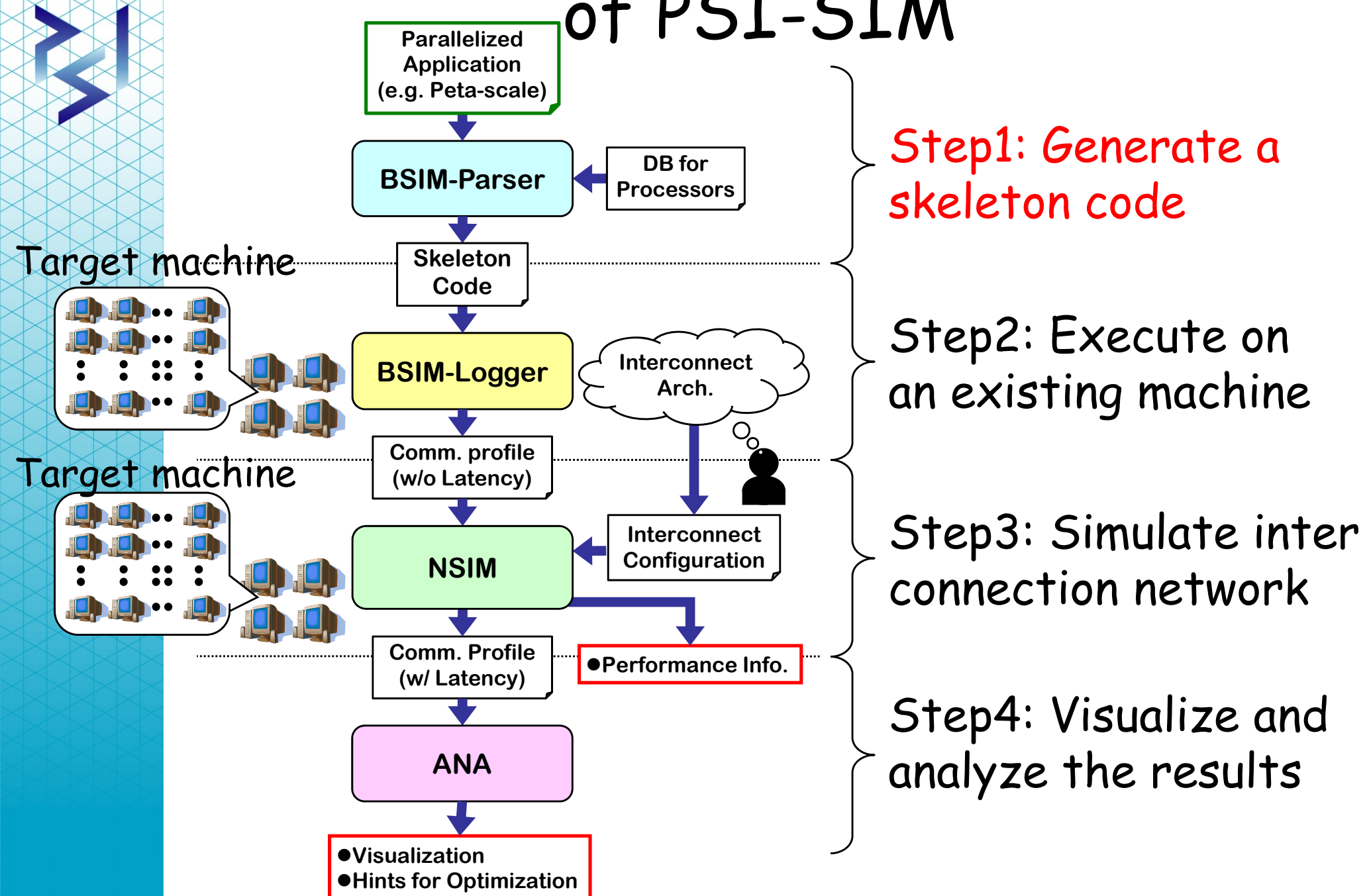
Our Goal!

- Problem...
 - Simulations are 3-orders of magnitude slower than real machines!
 - "Peta-scale" is 3-orders of magnitude larger than "Tera-scale" (i.e. available machines) !
 - How can we bridge the gap?
- Develop an efficient performance evaluation environment: PSI-SIM
 - **Divide** compute-node simulations and network simulations!
 - **Abstract** the target application program to accelerate simulation speed!

Performance-Evaluation Flow of PSI-SIM



Performance-Evaluation Flow of PSI-SIM



What is the Skeleton Code?

Original code

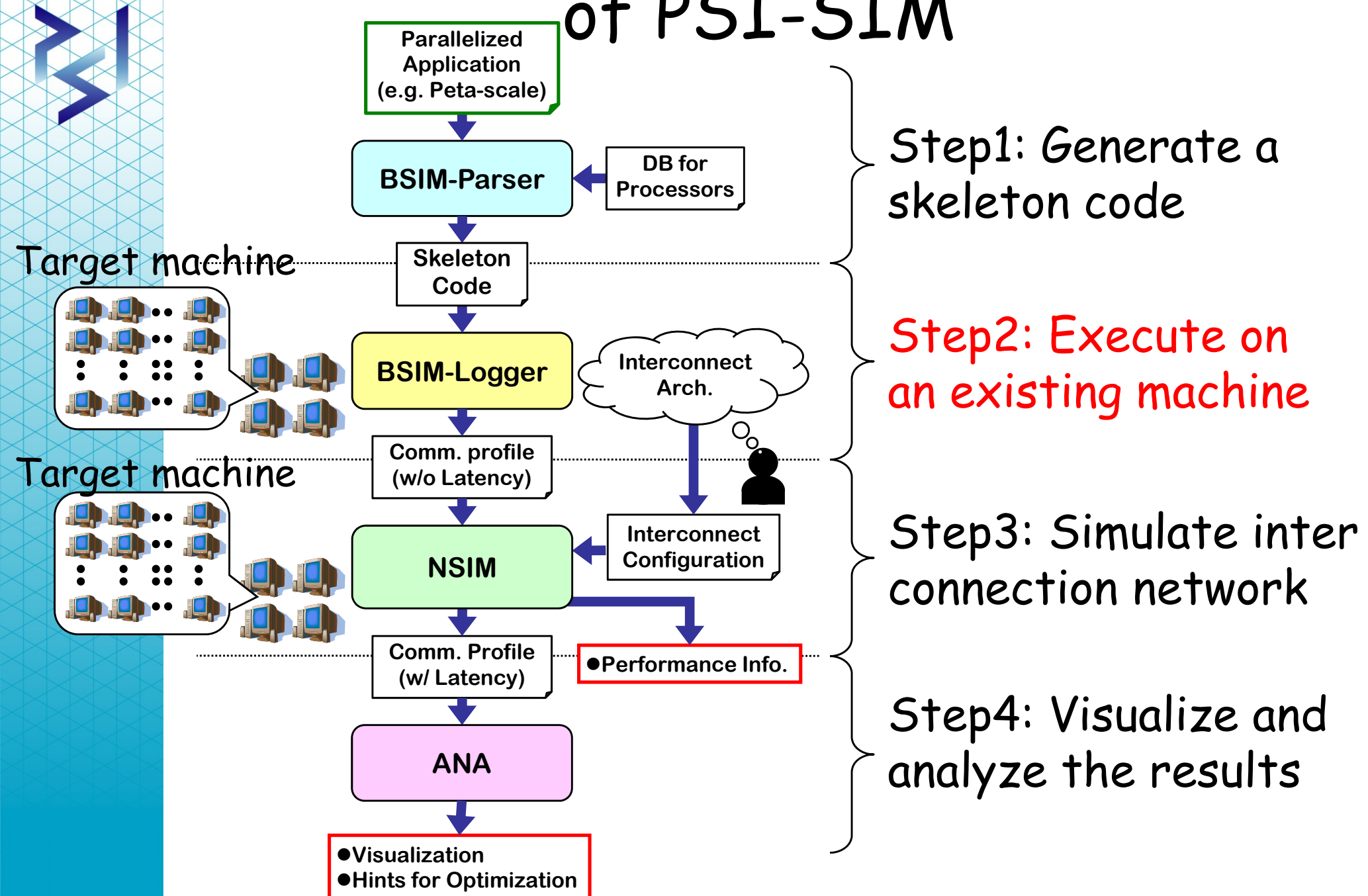
```
foo( ) {  
  Inst. Block A  
  for (i=0;i<n;i++) {  
    Inst. Block B  
    if (hoge) {  
      Inst. Block C  
    } else {  
      Inst. Block D  
    }  
    Inst. Block E  
  }  
  MPI_Comm.  
  Inst. Block F  
  for (j=0; j<n; j++)  
    for (k=0; k<n; k++)  
      Func( );  
}
```

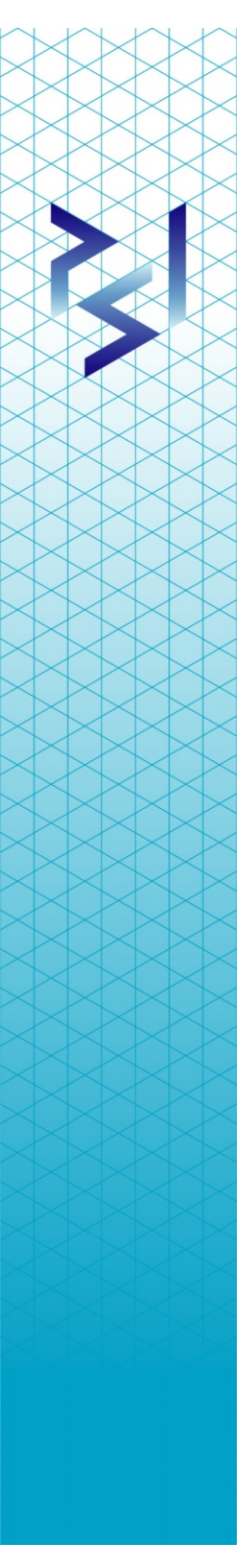
Skeleton code

```
foo( ) {  
  BSIM_ADD_TIME(10ms)  
  MPI_Comm.  
  BSIM_ADD_TIME(1ms)  
  BSIM_ADD_TIME(15s)  
}
```

- Computation blocks are replaced by "Estimated" execution times!
- Other modifications (e.g. reducing required memory size)

Performance-Evaluation Flow of PSI-SIM



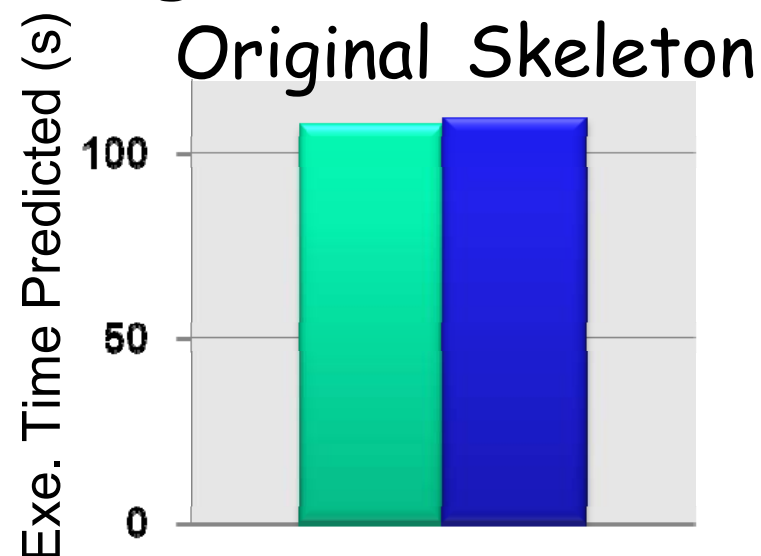
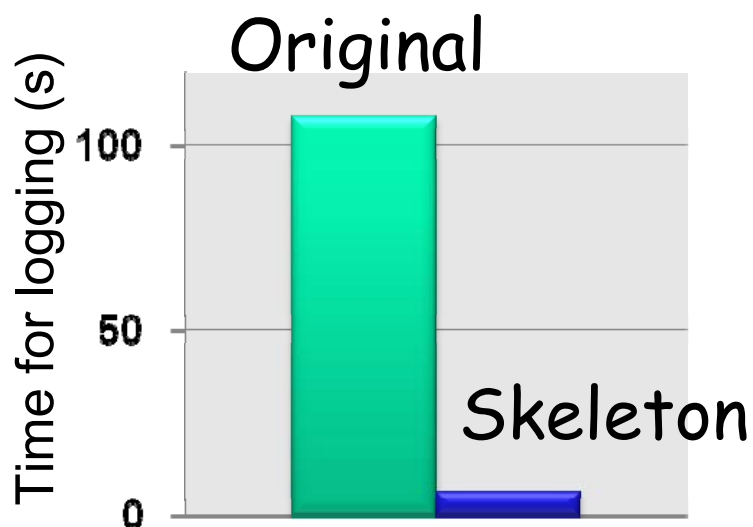


Generating Communication Profile

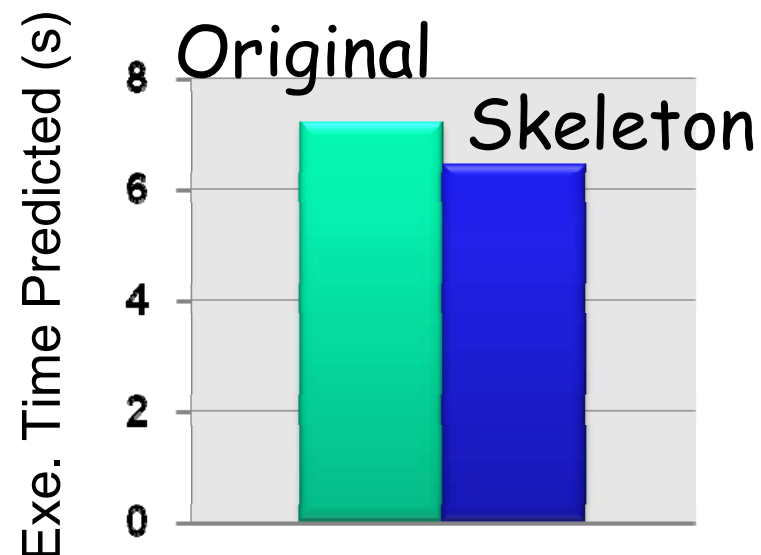
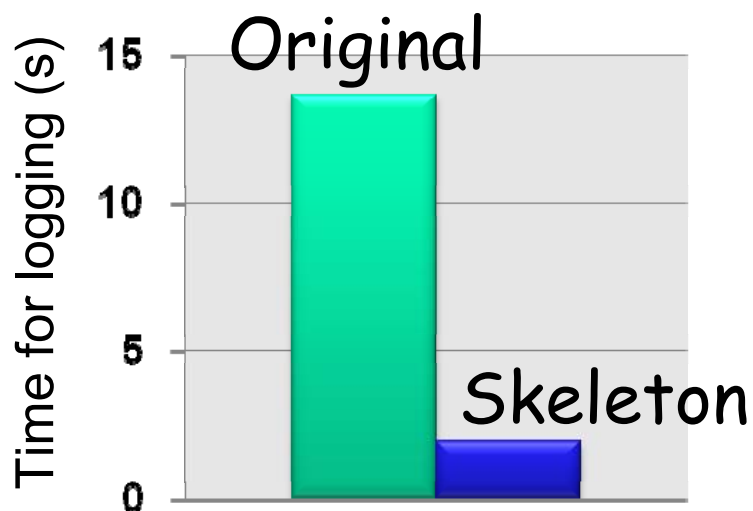
- BSIM-Logger
 - Executes the skeleton code on an existing machine
 - Emulates the behavior of target machine
 - Generates a communication profile under the assumption of a ZERO-latency ideal network
- Why Fast?
 - Abstracted computation blocks are NOT executed (just update virtual timers)
 - Mask real communications, but generate accurate logs

How Fast? How Accurate?

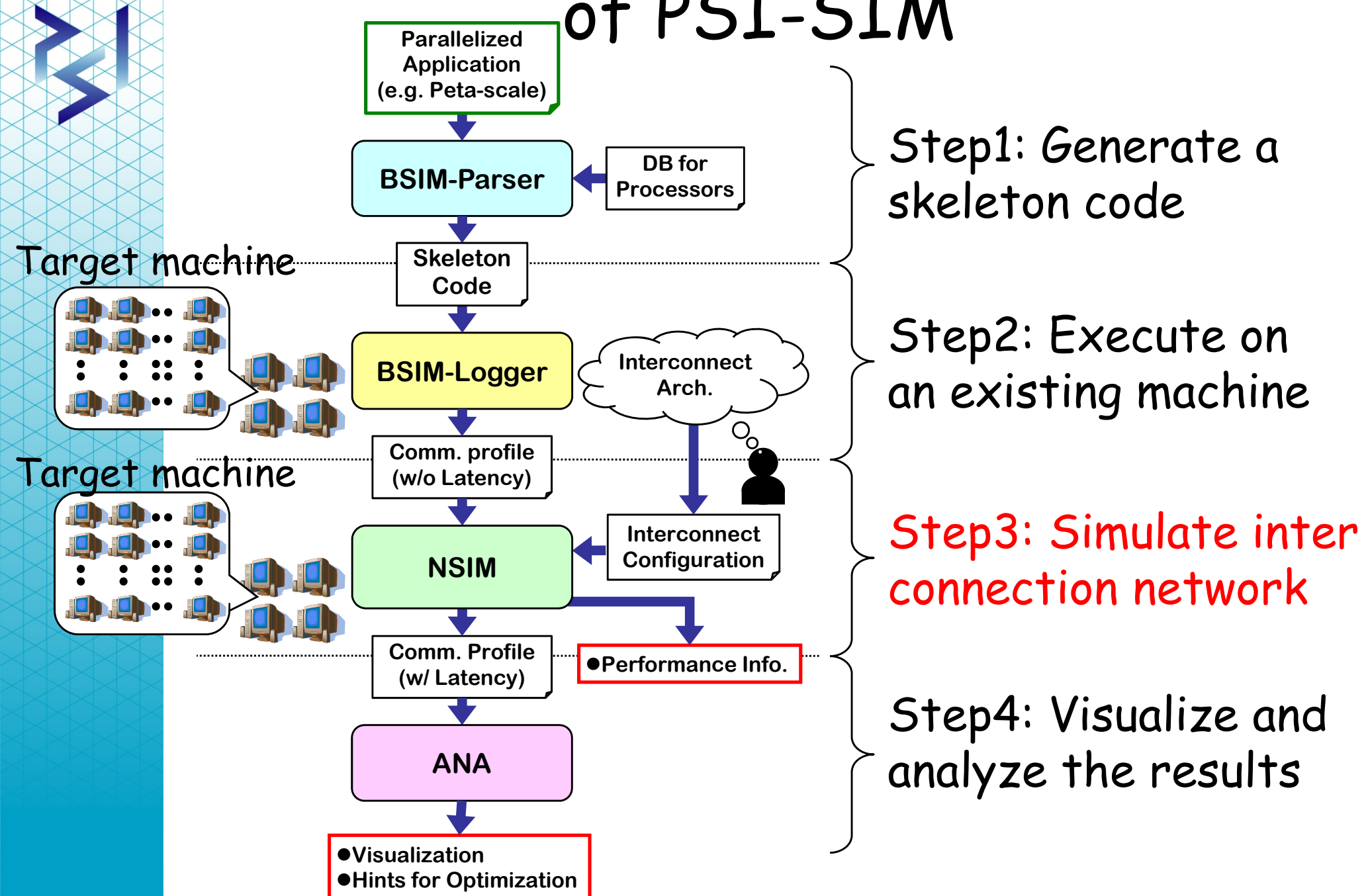
ERI (Electron Repulsion Integral)



NAS PARALLEL FT



Performance-Evaluation Flow of PSI-SIM

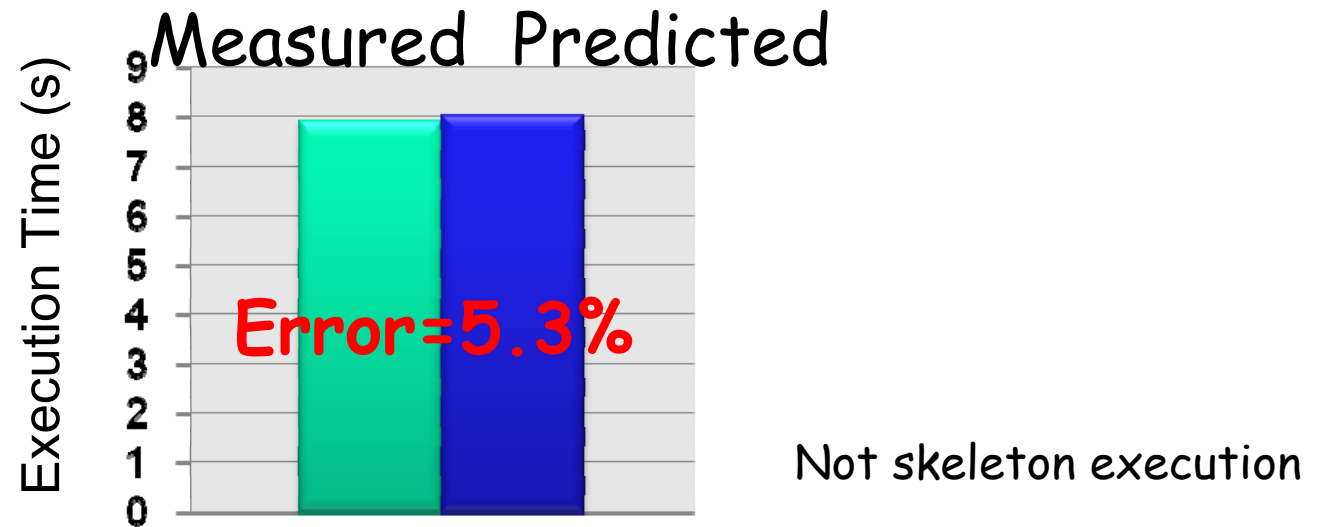


Fast, Flexible Interconnection Network Simulator



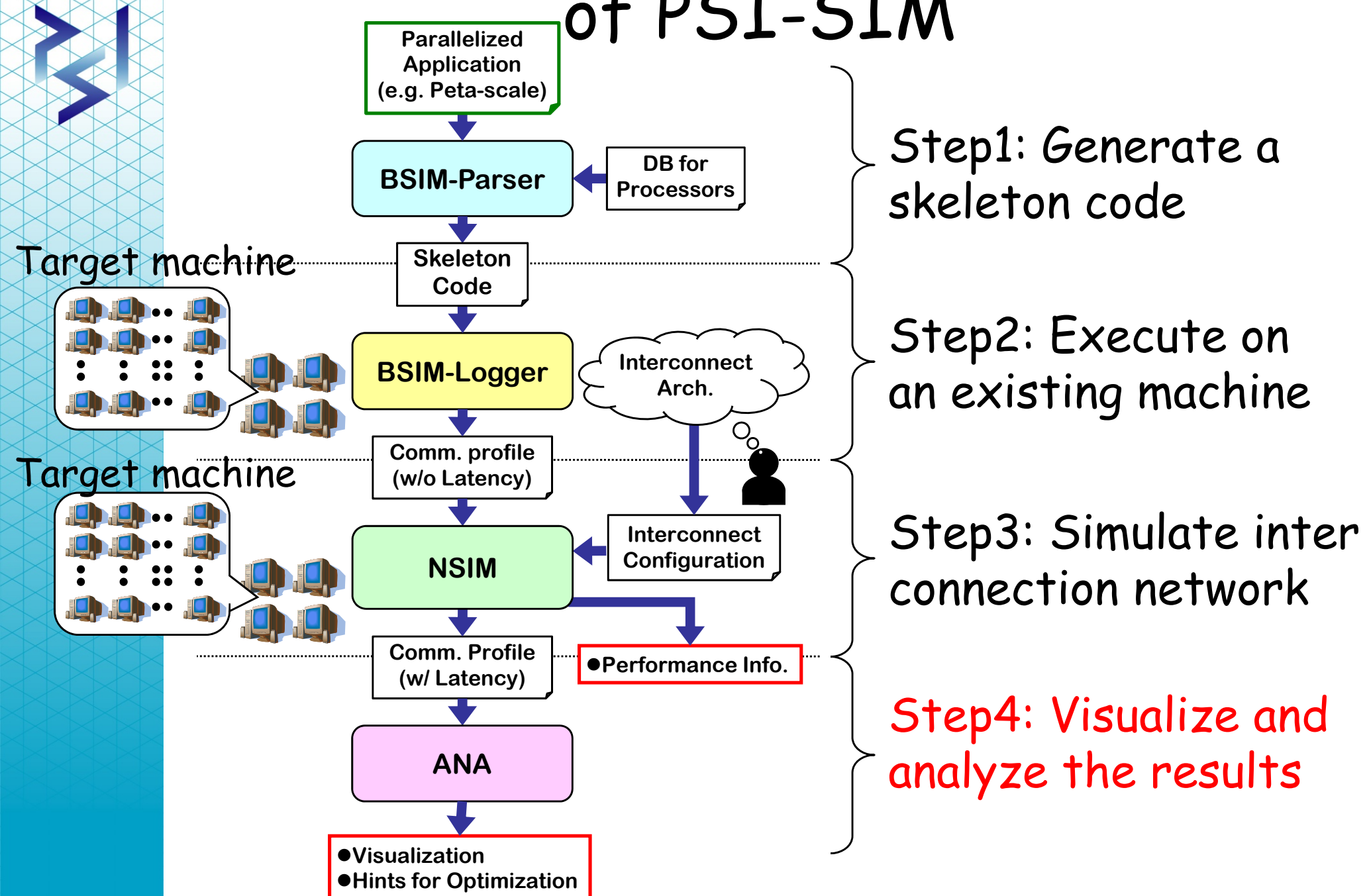
- NSIM
 - Inputs the communication profile and a network configuration file
 - Generates a communication profile with estimated interconnect latency
- Why Fast? Why Flexible?
 - Parallelized implementation
 - Support a number of parameters
 - Topology , Spec. of routers/switches, buffer size, and so on

Performance of BSIM + NSIM



- Performance prediction for HPL execution @16nodes PC cluster
- <120s (problem size = 5,000) @8CPU
- About 9,000 MPI-Comm./s@8CPU

Performance-Evaluation Flow of PSI-SIM



ANA GroupWork Viewer



Performance Indicator

- Execution time after load-balance optimization



Group Work

- Indicate load balance

Communication Indicator

- Amount of communications per second

Conclusions

- PSI-SIM
 - Performance evaluation environment for supercomputers
 - BSIM+NSIM+ANA
- On Going Work: Performance Prediction for
 - "Tera-Scale" machine (1K CPU Cores) by using a "Giga-scale" machine (e.g. 32 CPU Cores)
 - "Peta-Scale" machine (4K PSI-SIMD CPUs) by using a "Giga-scale" machine