Benchmarking and Tuning

Charlie Peck, Samuel Leeman-Munk
Intermediate Parallel Programming and Cluster Computing
@ University of Oklahoma OSCER
August, 2010

Please copy the accompanying materials from ~leemasa/tuning
cp -r ~leemasa/tuning ~

Efficiency

- Time

- Resources
  - CPU
  - Memory
  - Disk
  - Network

Benchmarking – Measuring
Tuning – Optimizing

First-level Benchmarking

- time (/usr/bin/time -p)
- vmstat
- iostat
- top

Detailed Benchmarking

- printf() (Fortran: PRINT)
- gprof
- getrusage() (Fortran: CPU_TIME)
- Performance counters (C: PAPI)

Fortran CPU_TIME information
http://gcc.gnu.org/onlinedocs/gcc-4.0.4/gfortran/CPU_005fTIME.html

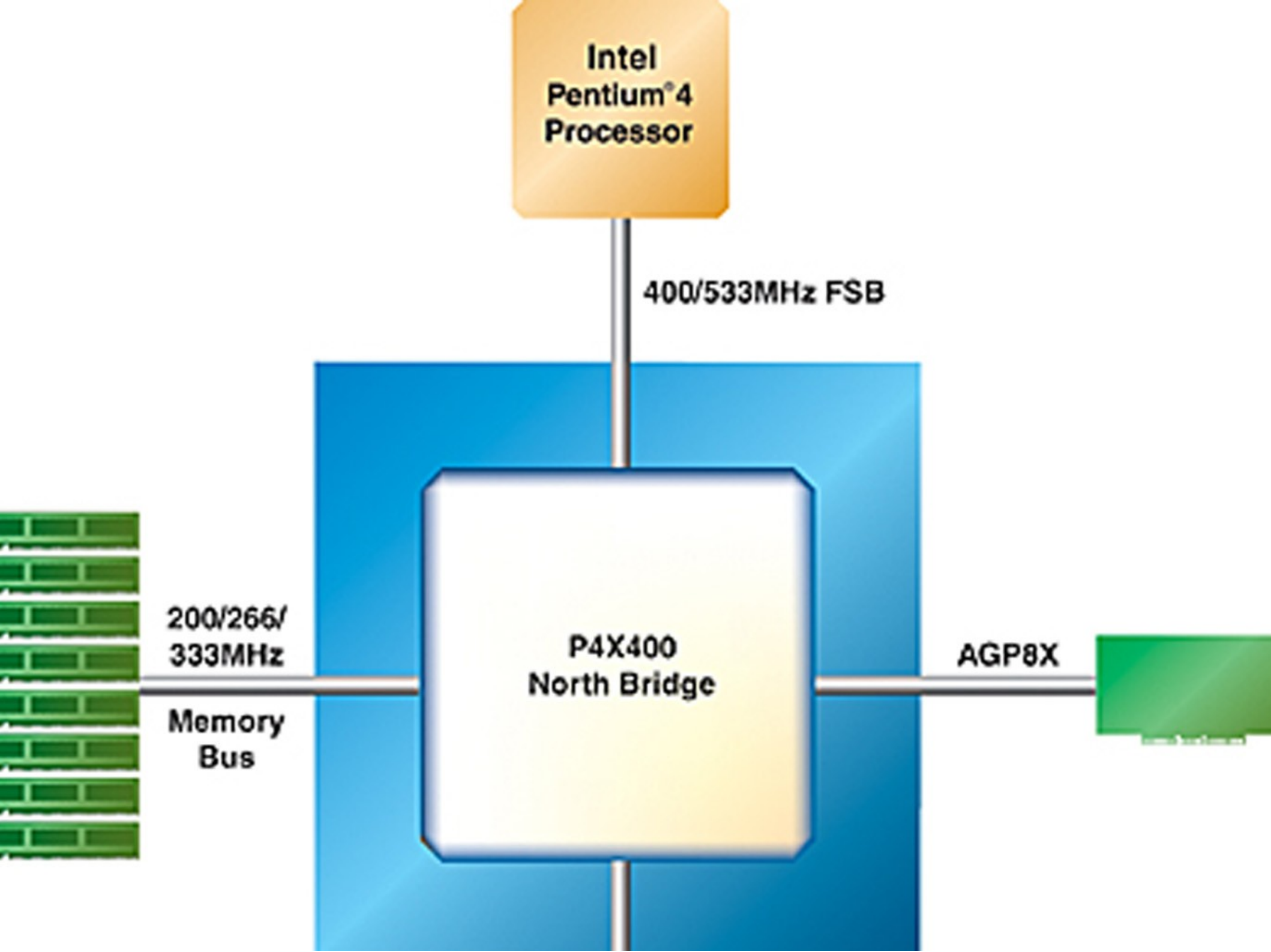
Tuning

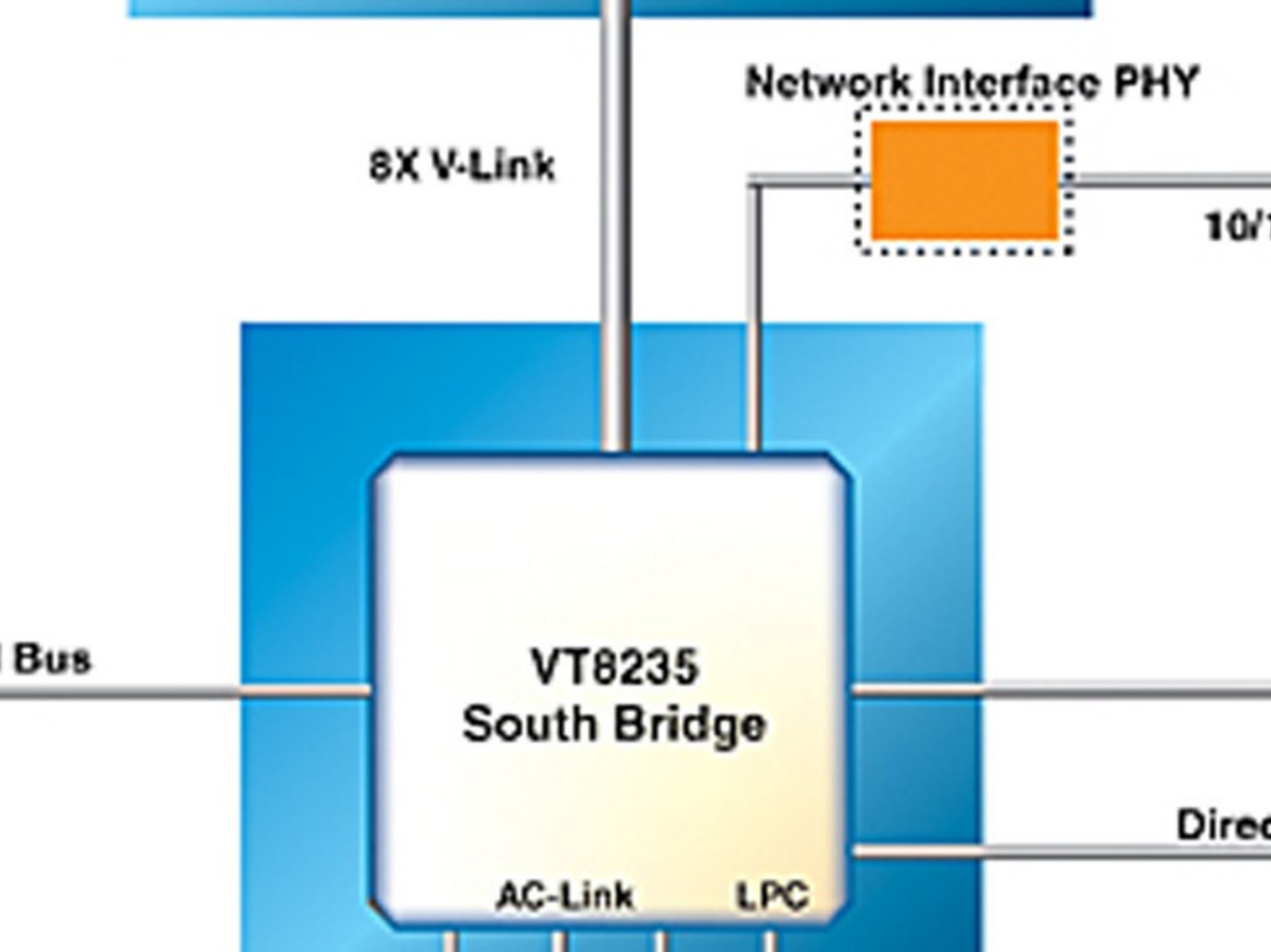
- Resource limits
- Compiler choice (gcc, icc)
- Compiler optimizations (-O,-funroll-loops)

RECONSIDER THE ALGORITHM

MPI Tuning

- Network port contention
- Communication vs. Computation
- Load Balancing


Intel
Pentium®4
Processor
400/533MHz FSB
P4X400
North Bridge
200/266/
333MHz
Memory
Bus
AGP8X


Network Interface PHY
8X V-Link
VT8235
South Bridge
AC-Link
LPC

MPI Tuning

- Network port contention
- Communication vs. Computation
- Load Balancing

Parallel Benchmarking with PetaKit

- Equip C source with pkit.h
  - See instructions in stats/README
  - Fortran users must code output manually

Running PetaKit

- perl stat.pl
- --cl 'mpirun.lsf /home/<username>/tuning/area-mpi \
- -s $problem_size'
- --problem_size 10000000000 --cores 1-8
- --ppn 8
- --scheduler lsf --database text --queue normal