Issues in Workload-Driven Evaluation

- Choosing workloads
- Scaling workload and architecture
- Picking parameters and analyzing sensitivity
- Metrics, and presenting results

© Culler 1995

Metrics for Comparing Machines

- Possibilities, given a parallel application:
  - **Execution time**
    - » **CPU Time**
    - » ***Wallclock Time***
  - **Speedup**
    - » **relative to *best sequential algorithm***
    - » **relative to *this sequential algorithm***
    - » **relative to *parallel program on one processor***
  - **MFLOPs, MIPS**
  - **Processor Utilization**

© Culler 1995

Reporting Execution Time

- Consensus is wall-clock is better than CPU user time
  - **CPU time does not record time that a process is blocked waiting**
  - **I/O operations usually not included, since they involve system activity**
- Execution time should be reported as function of number of processors and problem parameters

© Culler 1995

Speedup as a Metric

- **Speedup = T1(N)/Tp(N)**, N=problem parameter vector
- Tp(N) is performance on that problem on p procs of parallel machine
- What is T1(N)?
  - **parallel program on one processor of parallel machine?**
  - **"best" sequential implementation on one proc of parallel machine?**
  - **"best" sequential implementation on agreed-upon standard machine?**

© Culler 1995

Speedup or Time?

- Both have their place
- Execution time is appropriate for comparing machines
  - **it is what the user cares about!**
- Speedup is not, except for third T1(N) above
  - **slower the uniprocessor performance, better the speedup**
- But speedup is appropriate for evaluating how much parallelism is buying us over single proc. on a machine
  - **also for determining whether a given workload is appropriate for testing the parallel nature of the architecture**
- Processor utilization is not a useful metric for user or to compare machines
  - **may be for architect in understanding performance**

© Culler 1995

Operation Counts as Metrics

- **MFLOP/s:** Million floating point operations per second
- **MIPS:** Million instructions per second
-
- When most ops are floating point, FLOPs have advantage:
  - **instructions are much more compiler- and architecture-specific**
- Is ADD FLOP equivalent to MUL or DIV?
  - **can normalize somehow**
- Not good for comparing machines
  - **worse algorithm can yield higher FLOP rate but more time**
  - **who cares about FLOPS anyway?**
- Good for comparing measured hardware performance with peak for a given implementation of a benchmark

© Culler 1995

Beware Peak Values

- **Never obtained in practice**
  - **Peak = "guaranteed not to exceed" ...**
- Not meaningful for user with an application
- Even "sustained" performance is not really useful
  - **sustained on what benchmark?**

© Culler 1995

Other considerations

- Performance should be reported per benchmark
  - **interpreting results needs understanding of benchmark**
- Machine configuration (memory system etc.) must be specified
  - **can make dramatic difference to results; e.g. working set fitting**
- Averages over benchmarks are not very useful
  - **behavior is too dependent on benchmark, so average obscures**

© Culler 1995