The Most Needed Feature(s) for OpenMP 3.0

Bronis R. de Supinski

Center for Applied Scientific Computing

Lawrence Livermore National Laboratory

June 2, 2005

What makes new features worthwhile?

- **Support for underlying OpenMP philosophy**
  - High-level constructs
  - Easy-to-understand semantics
  - Low cost extension/modification of serial code

- **Programmer needs**
  - Variety of parallelism models
  - Control when appropriate

- **Clarity of specification (incontrovertible definitions)**
  - Portability
  - Not just "quality of implementation"

Features for Clarity of Specification

- **Little details that 2.5 deferred to 3.0**
  - **Over a dozen "small" outstanding issues**
  - **Many concern clarity of specification (e.g., directive grammar)**
  - **Orthogonality of constructs and base language**
    - **Reduction operators: min and max**
    - **Array reductions**
    - **Allow unsigned integers as LCVs**

- **(More) formalized memory model**
  - **Avoids natural language interpretations**
  - **Stated strictly in terms of operation orderings**
  - **Won't happen in 3.0…**

Several Worthy Candidates

- **Data distribution**
  - **Associate data to threads**
  - **Is this implementation- or architecture-specific?**

- **Task queues**
  - **Supports a very common form of parallelism**
  - **Long-standing, well-understood proposals**

- **Informational interface for tool support**
  - **Variable name mangling**
  - **Outline routines (or indicate that they aren't used)**
  - **Run-time library names**

- **Others…**

Personal Favorites

- **Contexts or subteams**
  - Allow (subsets of) team to be reordered and named
  - Provides greater user control
    - Synchronization
    - Sections with varying parallelism
  - Supports portable libraries

- **Work distribution**
  - User knows which thread should execute which work
  - Let them specify it!
  - Schedule rules help but not always natural
  - Often what users mean by "data distribution"

UCRL-PRES-212612

CASC