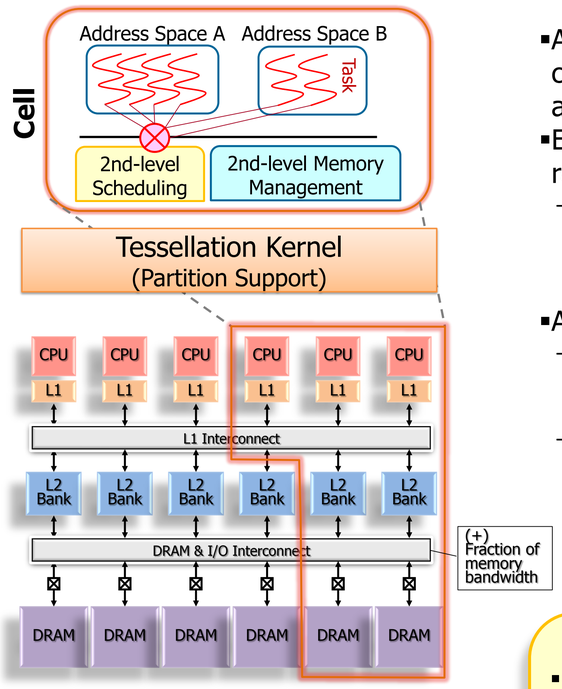
# Tessellation OS: Partition Management and Two-level Scheduling

Juan A. Colmenares, Sarah Bird, Paul Pearce, and John Kubiatowicz
Par Lab, CS Division, University of California, Berkeley

Parallel Computing Laboratory

## 1. Space-time Partitioning

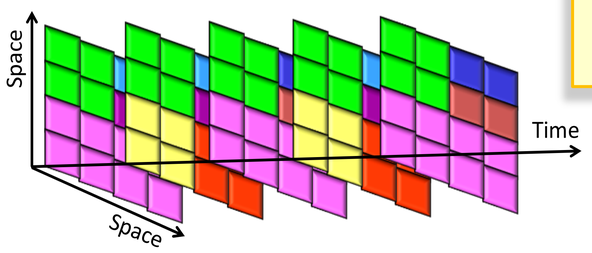



(*) The bottom part of this diagram was adapted from Liu and Asanovic, "*Mitosys: ParLab Manycore OS Architecture*," Jan. 2008.

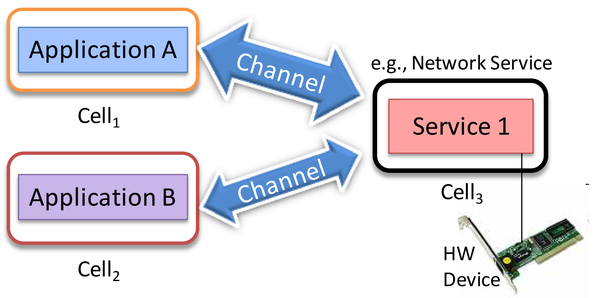
- A Spatial Partition (or Cell) comprises a group of processors acting within a hardware boundary
- Each cell receives a vector of basic resources
  - Some number of processors, a portion of physical memory, a portion of shared cache memory, and potentially a fraction of memory bandwidth
- A cell may also receive
  - Exclusive access to other resources (e.g., certain hardware devices and raw storage partition)
  - Guaranteed fractional services (i.e., QoS guarantees) from other partitions (e.g., network service and file service)

### Goals of Space-time Partitioning

- Provide responsiveness and/or QoS guarantees and more predictable real-time behavior to (parts of) applications
- Achieve better handling of power-performance tradeoffs
- Offer additional protection, fault-containment, and security capabilities

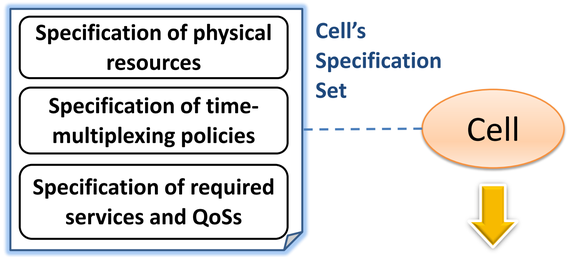


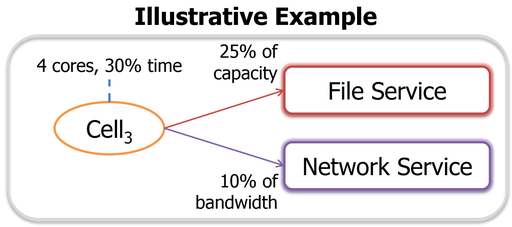

- Spatial partitioning may vary over time
  - Partitioning adapts to needs of the system
  - Some cells persist while others change with time



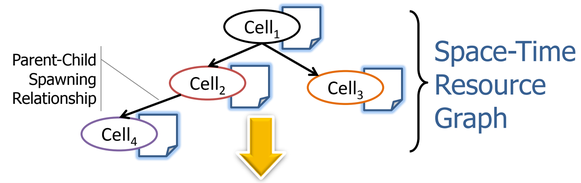

- A cell contains channel endpoints to other cells
  - Channels allow an application in a cell to access services and to interact with other applications residing in other cells
  - Message communication between cells is controlled for security and QoS enforcement
- Channels enable efficient and non-blocking message passing

## 2. Partition-management Software Layers



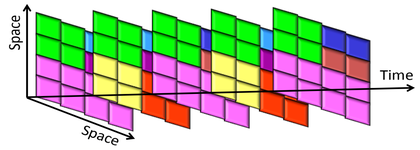

### Policy Layer




- Implements adaptation policies, which reflect global goals
- Does admission control (with help from the Mapping and Time-Mux'ing Layer)

### Mapping & Time-Multiplexing Layer

- Basic time-mux'ing policies for cells
  - Pinning (i.e., null policy)
  - Time Triggering
  - Time Fraction
  - Round Robin

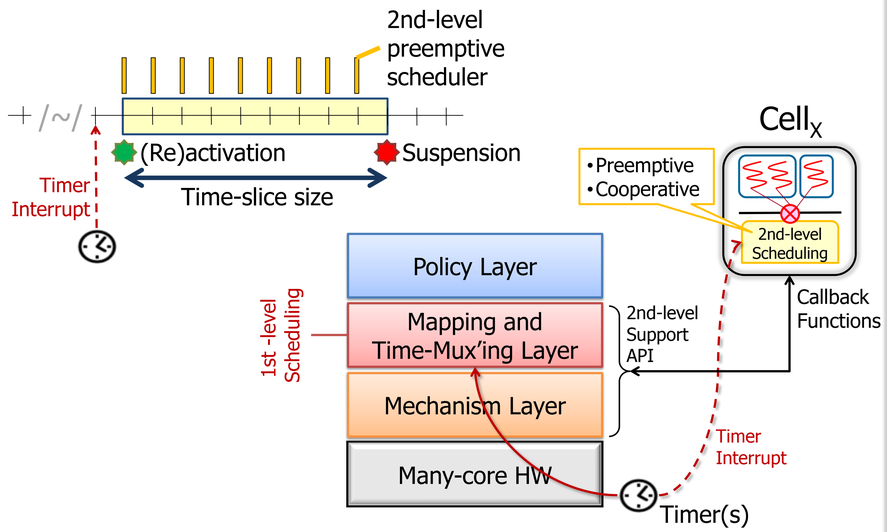



- Performs bin-packing like operation to implement the Space-Time Resource Graph
- Time-multiplexes cells at a coarse granularity

### Mechanism Layer

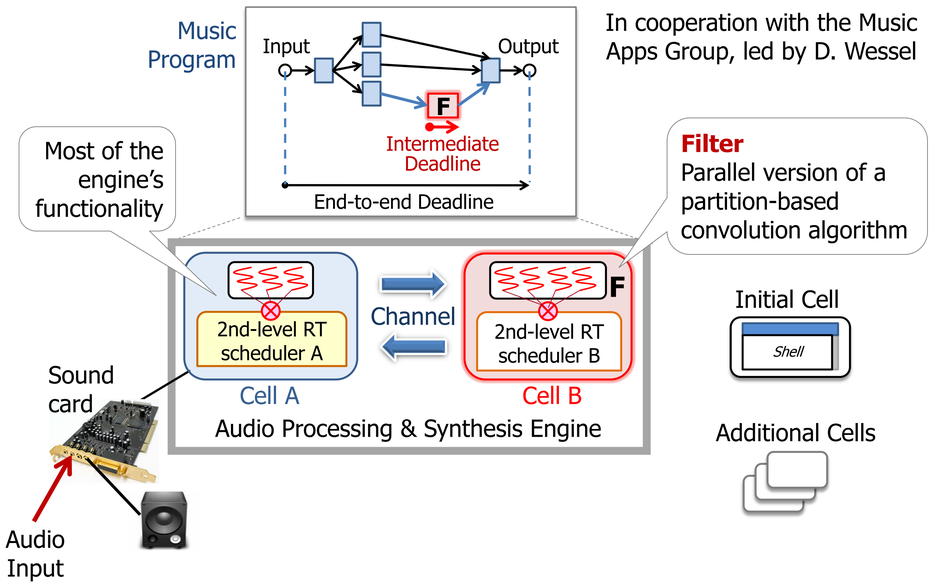
- Implements operations to establish, activate, and suspend cells, and to create inter-cell channels and send and receive messages through them
- Does performance and energy accounting for cells

## 3. Two-level Scheduling




- **Level 1**: Coarse-grained resource allocation and time-multiplexing at the cell level
- **Level 2**: Fine-grained application-specific scheduling *within* a cell

## 4. Future Demo Application




The goal is to show that Tessellation can provide acceptable performance and time predictability

Research supported by Microsoft (Award #024263) and Intel (Award #024894) funding and by matching funding by U.C. Discovery (Award #DIG0710227)