Execution Process Overview

- Task execution follows an ordered, dependency-aware process
- Tasks are executed in-order by default
- System handles dependencies and synchronization

Task Creation & spawn_datadeps

- Executes original caller's logic
- Creates DTaskSpec and DTask for each @spawn
- Enqueues tasks in queue.seen_tasks

Task Distribution

- distribute_tasks initiates execution
- Gathers compute resources (processors/GPUs)
- Determines execution order based on dependencies

Dependency Resolution

- Checks task arguments and access patterns
- Identifies inter-task dependencies
- Ensures completion of prerequisites

Aliasing Management

- Identifies memory location aliases
- Creates data copies to prevent conflicts
- Enqueues copy operations as new tasks

Scheduling & Execution

- Uses naive scheduler for task assignment
- Estimates execution costs
- Assigns tasks to optimal processors

Synchronization & Final Execution

- Synchronizes based on access patterns (read/write)
- Manages read and write dependencies
- Executes tasks in parallel on assigned processors