What data to cache?
Where to store the cached data?
What data to evict when the cache fills up?
When to store data back to system DRAM? (coherence)
What address to use to access cached data?
How to optimize programs to increase locality?
CPU CACHE

What data?
Where to store?
What to evict?
Coherence?
What address to use?
How to optimize?
SCRATCHPAD MEMORY
Software managed

What data to cache?
Where to store the cached data?
What data to evict when the cache fills up?
When to store data back to system DRAM? (if ever)
What address to use to access cached data?
How to increase use of scratchpad?
SCRATCHPAD MEMORY

- What data?
- Where to store?
- What to evict?
- Coherence?
- What address to use?
- How to optimize?
GPU DEVICE MEMORY

What data?
Where to store?
What to evict?
Coherence?
What address to use?
How to optimize?
GPU DEVICE MEMORY
Software managed, for correctness as well as performance

What data to cache?
Where to store the cached data?
What data to evict when the cache fills up?
When to store data back to system DRAM?
What address to use to access cached data?
How to increase use of scratchpad?
What: Data directives
Where: Anywhere
Evict: Error
Coherence: Update
Address: Device address but same name
Optimize: Correctness
What data to cache? Data directives and clauses.

Where to store the cached data? Anywhere in device memory.

What data to evict when the cache fills up? Program error.

When to store data back to system DRAM? Update directives, end of data region.

What address to use to access cached data? Device address != host address.

How to increase use of scratchpad? Must use device memory.
OPENACC USING MANAGED MEMORY

- `ta=tesla:managed` on K80

**What**: Allocated

**Where**: Driver managed

**Evict**: Error

**Coherence**: Driver

**Address**: Host address

**Optimize**: Locality works

malloc cost synchronization
CUDA UNIFIED (MANAGED) MEMORY

Driver managed on K80

What data to cache? All dynamically allocated.
Where to store the cached data? Driver managed.
What data to evict when the cache fills up? Runtime error.
When to store data back to system DRAM? Driver managed.
What address to use to access cached data? Host address.
How to increase use of device memory? Locality works.
CUDA UNIFIED (MANAGED) MEMORY

OpenACC -ta=tesla on K80

What data to cache?  All dynamically allocated data (malloc, new, allocate).

Where to store the cached data?  Driver managed.

What data to evict when the cache fills up?  Runtime error.

When to store data back to system DRAM?  Driver managed.

What address to use to access cached data?  Host address.

How to increase use of device memory?  Locality works.
CUDA UNIFIED (MANAGED) MEMORY

Driver managed on K80

Allocation is expensive, allocates both device and host pinned memory
Can only allocate up to size of device memory
All data migrated to device for each kernel launch (and paged back)
Host must not access managed data while GPU is active
Data movement is fast (host pinned memory)
OPENACC USING MANAGED MEMORY
- \texttt{ta=tesla:managed} on P100

- **What**: Allocated
- **Where**: Driver managed
- **Evict**: Driver managed
- **Coherence**: Driver
- **Address**: Host address
- **Optimize**: Locality works
  
  *malloc cost*
CUDA UNIFIED (MANAGED) MEMORY

Driver managed on P100

What data to cache? All dynamically allocated.
Where to store the cached data? Driver managed.
What data to evict when the cache fills up? Driver managed.
When to store data back to system DRAM? Driver managed.
What address to use to access cached data? Host address.
How to increase use of device memory? Locality works.
CUDA UNIFIED (MANAGED) MEMORY

Driver managed on P100

Allocation is expensive, allocates host pinned memory
Can allocate more memory than GPU (oversubscription)
Data is paged to GPU (and paged back)
Host may access managed data while GPU is active
User can set preferences and prefetching policies
SPEC ACCEL™ opens the path to optimizing OpenACC performance with managed memory vs. directives.

POWER8+NVLink and P100 GPU, PGI 17.1 compilers

PGI 17.1 Compilers OpenACC SPEC ACCEL™ 1.1 performance measured March, 2017. SPEC® and the benchmark name SPEC ACCEL™ are registered trademarks of the Standard Performance Evaluation Corporation.
356.SP PERFORMANCE ANOMALY
356.SP PERFORMANCE ANOMALY
356.SP PERFORMANCE W/ POOL ALLOCATOR
356.SP PERFORMANCE W/ POOL ALLOCATOR

POWER8

Haswell
CUDA UNIFIED MEMORY ON PASCAL & BEYOND

Future HMM OS Option

Works for static data, global data, dynamically allocated data

All memory treated as managed

Data is paged to GPU (and paged back)

Host may access managed data while GPU is active

User can set preferences and prefetching policies
SUFFICIENT?
Why use OpenACC data directives at all?

- Portability to environments where Unified Memory isn’t supported
- Useful to runtime as hints to set preferences or to prefetch data
- Moving large blocks is much more efficient than demand paging
THE FUTURE
GPU Device Memory as a Cache

Compile for multicore (let hardware cache manage data)
Compile for GPU (let driver manage data)
Add data directives as needed (for correctness and performance)