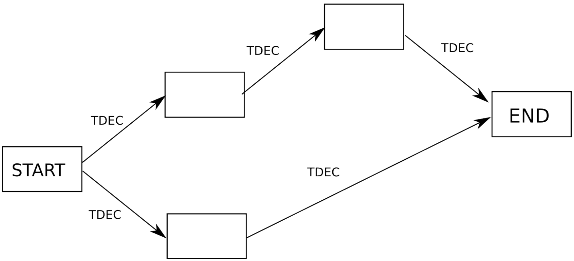
TStar : Extending Dataflow

Boris Arnoux, Software Engineer at PARKAS Team, INRIA ENS-Ulm, Paris

November 27, 2011

Dataflow computation is data driven, it is built upon a DAG (Dataflow Graph)



We extend Dataflow : A purely functionnal paradigm.

Figure: Example of Dataflow graph.

The advantages of Dataflow are numerous :

- Hiding latencies.
- Deterministic execution.
- High parallelism.
- It composes well.

This model could be an excellent alternative to threads !!
However, it remains restricted to purely-functional programs.

How can we complete this model to make it more powerful ?

- We need some persistent construction.
- We need to preserve determinism.

But before, a few words about the system model...

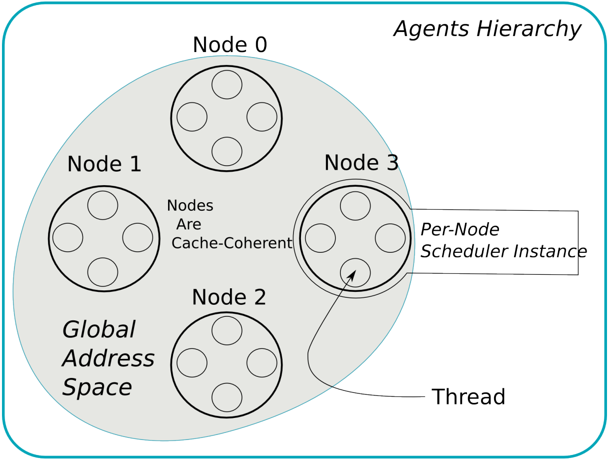


Figure: Hierarchy of agents

We introduce OWM as allocatable memory :

- It is shared at an intermediate level in the hierarchy.
- It is created in a reserved area for every node.
- The scheduler prepares resources asynchronously.

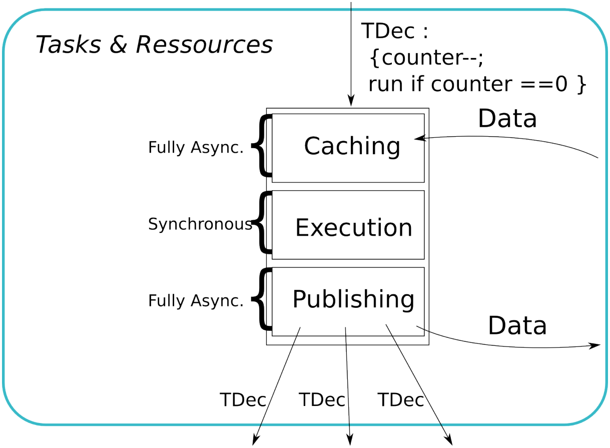


Figure: Interaction between tasks and ressources.

What semantics to give to Cache and Publish Operations ?
We have some things to consider :

- Dependencies.
- Task activation.
- Distributed copies.

We introduce a new multi-purpose, cache coherence protocol :

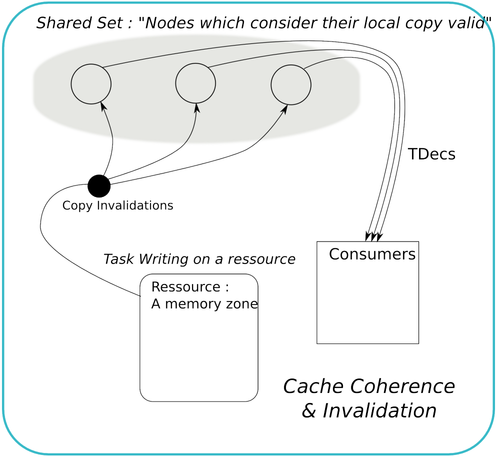

Shared Set : "Nodes which consider their local copy valid"
TDecs
Copy Invalidations
Task Writing on a ressource
Consumers
Ressource :
A memory zone
Cache Coherence
& Invalidation

Figure: A multi-purpose protocol.

What does this protocol allow ?

- It links "write" causality with dataflow causality.
- It allows specifying causality directions in the graph.
- It limits the complexity with conservative assumptions.

Future perspective :

- Elaborate more than examples.
- Provide high level OpenMP pragmas.
- Provide Object-Based C++ interface.

Questions ?