Heterogeneous Chip Multiprocessor Design

O. Ozturk,
Bilkent University

Architecture Session

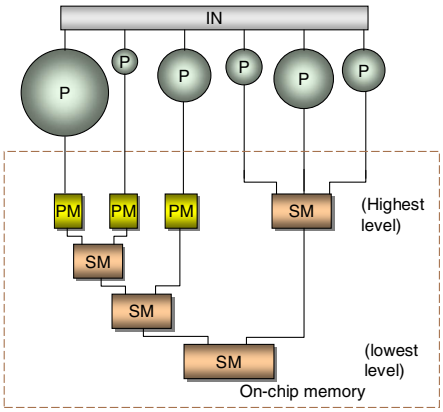
Motivation

- Homogeneity inherent to Chip Multiprocessors (CMPs) is a source of limitation in extracting the best utilization from these architectures.
- To overcome the limitations due to the homogeneous behavior of CMPs, Heterogeneous (asymmetric) CMPs have been proposed.
- Every application has a different processing need and a memory requirement. Even one single application has different requirements throughout its execution.
- An application may exploit a high level of instruction-level parallelism where a powerful core will be a better match, whereas a simpler core will suffice for a different application with lower instruction-level parallelism.

Motivation

- To choose the best match for an application will reduce the power consumption.
- However, homogeneous CMPs provide only one type of core to match all these various requirements.
- We propose a heuristic solution to the problem of memory hierarchy management, i.e., the data management across the memory hierarchy.
- The goal is to determine the optimal data transfers across the different levels of a software-managed multi-level memory hierarchy so that the overall energy consumption in memory is minimized.

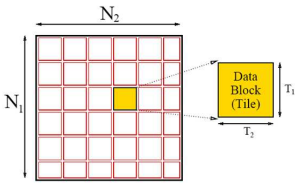
Our Approach



Our Approach

– Early design decision: perform data transfers at well-defined program points.

– The data access pattern of multiple nests can be obtained as a concatenation of the individual access patterns of the nests involved.




```
for j1=1, N1, 1
  for j2=1, N2, 1
    ...X[j1,j2]...
    ...X[j1-1,j2+1]...
```

$\Longrightarrow$

```
for ii1=1, N1, T1
  for ii2=1, N2, T2
    for i1=ii1, max(N1,ii1+T1-1), 1
      for i2=ii2, max(N2,ii2+T2-1), 1
        ...X[i1,i2]...
        ...X[i1-1,i2+1]...
```

Memory Distribution

- Every application has a different processing need and a memory requirement. Even one single application has different requirements throughout its execution.
- We call the former as inter-application variation, whereas the latter as intra-application variation. We collect the application behavior using a profile tool and generate an affinity table among the processors to identify the shared and private data elements.
- If certain processors have data sharing above a given threshold we create a shared memory component for these two processors accordingly.
- Similarly, a processor exhibiting a primary data access beyond a certain level we assign a private memory.

Conclusion

- Heterogeneous chip multiprocessors are becoming more and more attractive from both area and power angles. Therefore, it is more important than ever to explore options for coping with memory access problems, instead of using the previously proposed methods on homogeneous chip multiprocessors.
- This paper focuses exclusively on software-managed on-chip memory components and tries to employ an efficient memory hierarchy according to the application needs.
- Our preliminary results indicate that the proposed approach generates promising results and the solution times it takes are not excessive.