The Performance Model of SilkRoad – A Multithreaded DSM System for Clusters

Presented by Weng-Fai Wong

Department of Computer Science

School of Computing

National University of Singapore

Background - Cilk

- Cilk is an efficient multithreaded runtime system for SMPs and Clusters developed at MIT.
- Cilk is good at Divide-and-Conquer paradigm.
- Cilk is using a work stealing scheduler for load balancing.
- There is no user level shared memory in (distributed) Cilk because of the nature of D-&-C.

Introduction to SilkRoad

- SilkRoad system is a variant of Cilk
- SilkRoad extends the memory consistency model of Cilk, which results in RC_dag consistency model
- SilkRoad provides user level shared virtual memory
- SilkRoad supports more paradigms and wider applications

The memory consistency model in SilkRoad - RC_dag

- RC_dag memory consistency model is extended from location consistency (LC).
- RC_dag permits the interactions via mutual exclusion (e.g. locks) between sibling threads.
- RC_dag also allows global synchronization (e.g. barriers) between threads.

RC_dag consistency in SilkRoad

- inherited work-stealing scheduler and thread migration of Cilk,
- improved the existing implementation of LC by eliminating the “home” and trigger the diff transferring with thread stealing and return.
- included the semantics of Lazy Release Consistency (LRC), and
- implemented barrier and lock.

Performance Analysis (1)

$T_p = T_c + T_s + T_{\text{sync}}$

- T_c is the computation time
- T_s is the scheduling overhead
- T_{sync} is the overhead caused by global synchronization

Performance Analysis (2)

- In Cilk, the execution time on P processor
  $$T_p = O(T_1 / P + T_\infty)$$
  where T_1 is the execution time on one processor and T_∞ is the execution time on infinite processors.
- In SilkRoad,
  $$T_p \approx c_1(T_1 / P) + c_\infty T_\infty + P\ T_s(N)$$
  where c_1 and c_∞ are constants.

Experimental Framework

- Cluster of eight 500MHz Pentium III PCs
- Connected by 100 Mbps Ethernet
- Six applications
  - Matrix multiplication
  - N-queen problem
  - Barnes-Hut
  - LU Decomposition
  - TSP
  - Embarassing parallel application

Experimental Results

Applications		sequential execution	2 processors	4 processors	8 processors
matmul	512×512	9.81s	5.79s	5.03s	4.73s
	1024×1024	84.66s	38.41s	28.08s	24.75s
nqueen	12	14.64s	6.99s	3.61s	2.15s
	13	76.61s	39.94s	19.75s	10.82s
	14	528.34s	310.31s	155.03s	81.98
lu	512×512	18.16s	5.23s	4.82s	4.77s
	1024×1024	83.56s	28.28s	21.74s	16.27s
barnes-hut	16384	144.2s	112.4s	96.68s	81.53s
tsp	19b	11.58s	6.85s	5.49s	4.75s
ep	2^{24}	23.02s	11.66s	6.01s	3.15s

Table 1. Timing of the SilkRoad applications.

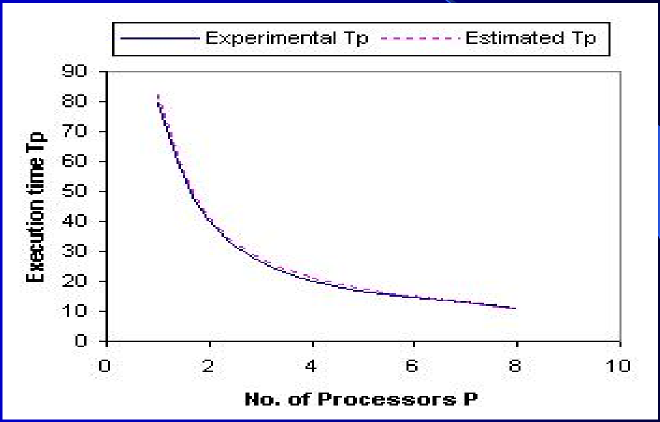
Performance of 13 Queens

- T_1 and T_∞ are 79.64s and 0.03s respectively (calculated by the runtime system)
- The constants c_1 and c_∞ are 1.03 and 10 respectively (adjusted according to experimental results)
- $T_s(N)$ is 0 since there is no lock operations

So, $T_p \approx 1.03\ (T_1 / P) + 10\ T_\infty$,

And the *coefficient of determination* is 0.996.

Modeling Performance (1)



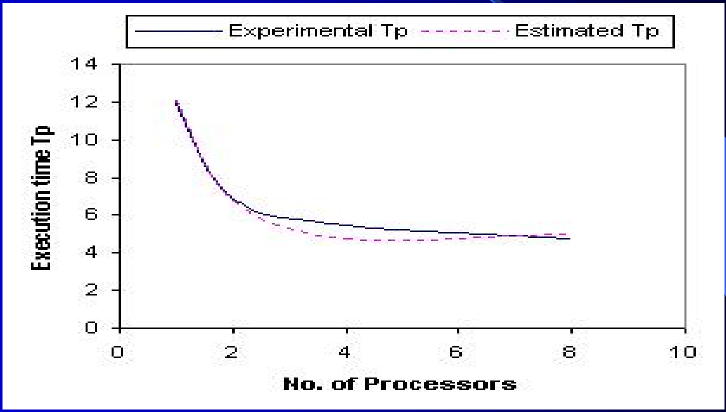
The performance model of Nqueen problem

Performance of TSP

- T_1 and T_∞ are 11.94s and 2.35s respectively (calculated by the runtime system)
- The constants c_1 and c_∞ are 0.962 and 0.085 respectively, $T_s(N)$ is 0.421 (adjusted according to experimental results)

So, $T_p \approx 0.962\,(T_1 / P) + 0.085\,T_\infty + 0.421P$,
And the coefficient of determination is 0.98.

Modeling Performance (2)



The performance model of TSP

Conclusion

- SilkRoad's performance analysis model is based on Cilk's theoretical performance model plus the consideration of global synchronization overhead.
- The performance model of SilkRoad is close to experimental results