Improving the Performance Competitive Ratios of Transactional Memory Contention Managers

Gokarna Sharma
Costas Busch

Louisiana State University, USA
STM Systems

- **Progress** is ensured through contention management (CM) policy

- **Performance** is generally evaluated by competitive ratio
  
  \[
  \text{Competitive ratio} = \frac{\text{makespan of my CM policy}}{\text{makespan of optimal CM policy}}
  \]

- **Makespan** primarily depends on the TM workload
  
  - arrival times, execution time durations, release times, read/write sets

- **Challenge**
  
  \textit{How to schedule transactions such that it reduces the makespan?}
Related Work

• Mostly empirical evaluation

• Theoretical Analysis
  – [Guerraoui et al., PODC’05]
    • **Greedy** Contention Manager, Competitive Ratio = $O(s^2)$ ($s$ is the number of shared resources)
  – [Attiya et al., PODC’06]
    • Improved to $O(s)$
  – [Schneider & Wattenhofer, ISAAC’09]
    • **RandomizedRounds** Contention Manager, Competitive Ratio = $O(C \log n)$ ($C$ is the maximum number of conflicting transactions and $n$ is the number of transactions)
  – [Attiya & Milani, OPODIS’09]
    • **Bimodal** scheduler, Competitive Ratio = $O(s)$ (for bimodal workload with equi-length transactions)
Our Contributions

• Balanced TM workloads

• Two polynomial time contention management algorithms that achieve competitive ratio very close to $O(\sqrt{s})$ in balanced workloads
  – Clairvoyant – Competitive ratio = $O(\sqrt{s})$
  – Non-Clairvoyant – Competitive ratio = $O(\sqrt{s} \cdot \log n)$ w.h.p.

• Lower bound for transaction scheduling problem
Roadmap

• Balanced workloads

• CM algorithms and proof intuitions

• Lower bound and proof intuition
Balanced Workloads

- A transaction is balanced if:
  \[
  \frac{|R_w(T_i)|}{|R(T_i)|} \leq \beta, \quad \frac{1}{s} \leq \beta \leq 1
  \]
  is some constant called balancing ratio

- \(|R(T_i)| = |R_w(T_i)| + |R_r(T_i)|\), where \(|R_w(T_i)|\) and \(|R_r(T_i)|\) are number of writes and reads to shared resources by \(T_i\), respectively

- For read-only transaction, \(|R_w(T_i)| = 0\), and for write-only transaction, \(|R_r(T_i)| = 0\)

- A workload is balanced if:
  - It contains only balanced transactions
Algorithms

• **Clairvoyant**
  – Clairvoyant in the sense it requires knowledge of dynamic conflict graph to resolve conflicts
  – Competitive ratio = $O(\sqrt{\log n})$

• **Non-Clairvoyant**
  – Non-Clairvoyant in the sense it does not require knowledge of conflict graph to resolve conflicts
  – Competitive ratio = $O(\sqrt{v_s \cdot \log n})$ with high probability
  – Competitive ratio $O(\log n)$ factor worse than *Clairvoyant*
Proof Intuition

- Knowing ahead the execution times ($t$) and total number of shared resource accesses ($|R(T_i)|$) of transactions

- **Intuition**
  - Divide transactions into $l+1$ groups according to execution time, where $l = \left\lfloor \log\left(\frac{t_{\text{max}}}{t_{\text{min}}}\right) \right\rfloor$
  - Again divide each group into $\kappa + 1$ subgroups according to shared resource accesses needed, where $\kappa = \left\lfloor \log(s) \right\rfloor$
  - Assign a total order among the groups and subgroups
Proof Intuition (Contd...)

- **Analysis**
  - For a subgroup $A_{ij}$, competitive ratio $= O(\min(\lambda^j, \frac{s}{\lambda^j}))$, where $\lambda^j = 2^{j+1} - 1$
  - For a group $A_i$, competitive ratio $= O(\sqrt[3]{\frac{s}{\beta}})$ after combining competitive ratios of all the subgroups
  - After combining competitive ratios of all groups, competitive ratio $= O(l \cdot \sqrt[3]{\frac{s}{\beta}})$
  - For $l = O(1)$ and $\beta = O(1)$, competitive ratio $= O(\sqrt{s})$

- $O(\log n)$ factor in Non-Clairvoyant due to the use of random priorities to resolve conflicts
Lower Bound

• **We prove the following theorem**
  
  – Unless NP $\subseteq$ ZPP, we **cannot** obtain a polynomial time transaction scheduling algorithm such that for every input instance with $\beta = 1$ and $l = 1$ of the **TRANSACTION SCHEDULING** problem the algorithm achieves competitive ratio smaller than $O((Vs)^{1-\varepsilon})$ for any constant $\varepsilon > 0$.

• **Proof Intuition**
  
  – Reduce the NP-Complete graph coloring problem, **VERTEX COLORING**, to the transaction scheduling problem, **TRANSACTION SCHEDULING**

  – Use following result [Feige & Kilian, CCC’96]
    
    • No better than $O(n^{(1-\varepsilon)})$ approximation exists for **VERTEX COLORING**, for any constant $\varepsilon > 0$, unless NP $\subseteq$ ZPP
Lower Bound (Contd...)

• Reduction: Consider input graph $G = (V, E)$ of VERTEX COLORING, where $|V| = n$ and $|E| = s$

• Construct a set of transactions $T$ such that
  – For each $v \in V$, there is a respective transaction $T_v \in T$
  – For each $e \in E$, there is a respective resource $R_e \in R$

• Let $G'$ be the conflict graph for the set of transactions $T$
  – $G'$ is isomorphic to $G$, for $\beta = 1$, $t_{min} = t_{max} = 1$, and $l = 1$
  – Valid $k$-coloring in $G$ implies makespan of step $k$ in $G'$ for $T$

• Algorithm Clairvoyant is tight for $\beta = O(1)$ and $l = O(1)$
Conclusions

• Balanced TM workloads

• Two new randomized CM algorithms that exhibit competitive ratio very close to $O(\sqrt{s})$ in balanced workloads

• Lower bound of $O(\sqrt{s})$ for transaction scheduling problem
[Full paper to appear in OPODIS 2010]


Thank You!!!