The Synchronization Power of Coalesced Memory Accesses

Phuong H. Ha (Univ. of Tromsø, Norway)
Philippas Tsigas (Chalmers Univ. of Tech., Sweden)
Otto J. Anshus (Univ. of Tromsø, Norway)
Problem

- Memory access mechanisms influence the system synchronization capability.

- Conventional wisdom: single-word assignment has consensus number 1
  ⇒ stronger synch. primitives (e.g. TAS, FAA, CAS) added.

- Can we make single-word assignment stronger?
  ⇒ transistors saved from strong synch. primitives can be used to enhance other functionality.

Transistor distribution

[These figures are from NVIDIA CUDA Programming Guide, version 2.0]
What is a memory word?

- A group of \( n \) bytes that can be stored or retrieved in a single, basic operation.
  - \( n \) is called word size
  (in byte-addressable memory)

- Words of size \( n \) must always start at addresses that are multiples of \( n \).
  (Alignment restriction)

[Hamacher et al. 2002, Hennessy et al. 2003]
Key idea 1

- **Word size** $n$ can be any integer
  - instead of powers of 2 as in conventional architectures
  - Ex: solving 2-process consensus using 2-byte write and 3-byte write.

- **Feasibility**: NVIDIA CUDA
  - *int1, int2, int3, int4*

**Size-varying word model (svword)**

<table>
<thead>
<tr>
<th>bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

- p's 2-byte write: `1 2 3 4 5 6 ...`
- q's 3-byte write: `1 2 3 4 5 6 ...`

Ex: solving 2-process consensus using 2-byte write and 3-byte write.

- [2,3,4] ⇒ p wrote first ⇒ agree on red

**Conventional architectures**

<table>
<thead>
<tr>
<th>bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

- p's 2-byte write: `3 4 5 6 7 8 ...`
- q's 4-byte write: `3 4 5 6 7 8 ...`

- [4,5,6,7] ⇒ q cannot determine if p has written!
Key idea 2

- Some of the $n$ bytes of a word may be left untouched in a single-word assignment.
  - Ex: solving 2-process consensus using 4-byte writes
  - Feasibility: NVIDIA CUDA
    - Coalesced memory accesses

Aligned-inconsecutive word model (aiword)

<table>
<thead>
<tr>
<th>bytes</th>
<th>...</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>p's 4-byte write</td>
<td>...</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>...</td>
</tr>
<tr>
<td>q's 4-byte write</td>
<td>...</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>...</td>
</tr>
</tbody>
</table>

$[4, 5, 6] \Rightarrow p$ wrote first $\Rightarrow$ agree on red

- Feasibility: NVIDIA CUDA
Our main technical contributions

- Develop **general models** for coalesced memory accesses.
- Prove the **exact consensus numbers** of these models:
  - size-varying word model (svword)
  - aligned-inconsecutive word model (aiword)
  - the **combination** of these two models (asvword)
Road-map

- Size-varying word model (svword)
- Aligned-inconsecutive word model (aiword)
- The combination of these two models (asvword)
Size-varying word model (svword)

- A svword consists of $b$ consecutive memory units, $b \in [1,B]$, $B$ is a constant.
  - $b$-svword for short
  - $b$-svwrite = $b$-svword assignment

- Alignment restriction:
  - Svwords of size $b$ must start at addresses that are multiples of $b$.

- Ex: 2-svwrite, 3-svwrite and 5-svwrite
Svword’s consensus no. ≥ 3

- **Idea:**
  - 5-svwrite can *partly overlap* both 2-svwrite and 3-svwrite
  - ⇒ can construct (binary) consensus objects for 3 processes

- **Ex:**
  - *Binary consensus (BC)* for 3 processes
  - *Consensus* for 3 processes

- **Diagram:**
  - p₁’s 2-svwrite...
  - p₂’s 3-svwrite...
  - p₃’s 5-svwrite...

- **Example:**
  - [17,18,20] ⇒ p₃’s write → p₂’s write
  - [14,15,16] ⇒ p₁’s write → p₃’s write ⇒ *red* wrote first ⇒ agree on red

DISC ’08
Svword’s consensus no. ≤ 3

- Idea
  - p’s critical assignment must
    - write to p’s private unit
    - partly overlap q’s critical assignment if p’s critical value ≠ q’s critical value (Bivalency argument)
  - b-svwrite accesses consecutive units ⇒ each b-svwrite can partly overlap at most 2 other b-svwrites.

Svword’s consensus number is exactly 3
Road-map

- Size-varying word model (svword)
- Aligned-inconsecutive word model (aiword)
- The combination of these two models (asvword)
Aligned-inconsecutive word (aiword)

- Memory is aligned to $m$-unit words, $m$ is a constant.
  - $m$-aiword for short

- A read/write operation accesses an arbitrary non-empty subset of the $m$ units of an aiword.
  - $m$-aiwrite = $m$-aiword assignment.

- Alignment restriction
  - $m$-aiwords must start at addresses that are multiples of $m$.

- Ex: 8-aiwrite
**m-aiword's consensus no. ≥ ⌈(m+1)/2⌉**

- **Idea:**
  - Construct a *binary* consensus object for $N=⌈(m+1)/2⌉$ processes in which $(N-1)$ processes propose the same value.
  - Construct a *multivalued* consensus object for $N$ processes using the binary consensus object.

- **Ex: 9-aiword**

  **Binary consensus (BC) for 4+1 processes**

  **Consensus for 5 processes**

  - Writing schema:
    - $p_0, p_1, p_2, p_3$
    - $[0, 4, 8] \Rightarrow p_4 \rightarrow p_0$
    - $[1, 5, 8] \Rightarrow p_1 \rightarrow p_4$
    - $[2, 6, 8] \Rightarrow p_4 \rightarrow p_2$  \(\Rightarrow \text{red wrote first}\)
    - $[3, 7, 8] \Rightarrow p_4 \rightarrow p_3$

  - Time:
    - $p_0, p_1, p_2, p_3, p_4$
**m-aiword's consensus no. \( \leq \lceil (m+1)/2 \rceil \)**

**Idea:**
- Lemma: \( p_i \)'s critical assignment must atomically write to
  - \( p_i \)'s own unit \( u_i \)
  - shared units \( u_{i,j} \) written only by \( p_i \) and \( p_j \) where \( p_i \)'s critical value \( cv_i \neq p_j \)'s critical value \( cv_j \).
  (Bivalency argument)

\[ \Rightarrow \text{solving consensus for 2 subsets } S_1 \text{ and } S_2, \text{ where } cv_1 \neq cv_2 \text{ and } n_1 + n_2 = N, \text{ needs to write atomically to } m \text{ units, where } m = N + n_1n_2 \geq 2N - 1 \Rightarrow N \leq (m+1)/2 \]

**m-aiword's consensus number is exactly \( \lceil (m+1)/2 \rceil \)**
Road-map

- Size-varying word model (svword)
- Aligned-inconsecutive word model (aiword)

The combination of these two models (asvword)
Asvword = aiword + svword

- An extension of aiword:
  - aiword’s m units are replaced by m svwords of the same size b, b ∈ \{1,B\}.
  - m.b-asvword for short
  - m.b-asvwrite = m.b-asvword assignment
  - m=t.B or B=t.m, t∈N*.

- Alignment restriction
  - m.b-asvwords must start at addresses that are multiples of (m.b).

- Ex: m=8, B=2:
  - 8.2-asvword vs. 8.1-asvword

DISC ’08
Asvword’s consensus no. when \( m \leq B \)

- Asvword’s consensus number is \( \lvert (m+1)/2 \rvert \), like aiword’s.
- Idea:
  - When \( B=t.m, t \in \mathbb{N}^* \), the combination of \( m.1\)-asvwrite and \( m.B\)-asvwrite does not provide any additional strength compared to \( m\)-aiwrite.
- Ex: \( B=m=4 \)
  - \( p \) and \( q \) write to \( u_p, u_q, u_{p,q} \) using 4.1-asvwrite and 4.4-asvwrite.

\[ \text{4.1-asvword} \]
\[ \text{p's 4.1-asvwrite} \]
\[ \text{b=1} \]
\[ \text{4.4-asvword} \]
\[ \text{q's 4.4-asvwrite must} \]
\[ \text{overwrites } u_p! \]
\[ \text{DISC '08} \]
Asvword's consensus no. when \( m > B \)

- **Asvword's consensus number** \( N \)
  - \( \frac{mB}{2} \) if \( m = 2tB, t \in \mathbb{N}^* \)
  - \( \frac{(m-B)B}{2} \) if \( m = (2t+1)B \)

- **Idea**
  - Processes can atomically modify \( m.B \) units using \( m.B \)-asvwrite vs. \( m \) units using \( m \)-aiwrite.
  - Avoid overwriting unintended units:
    - Each B-svword contains either private units or shared units, but not both.

- **Ex**: \( m = 8, B = 2 \) \( \Rightarrow \) \( N = 8 \)

---

**Binary consensus (BC) for 7+1 processes**

**8.1-asvw**

- \( p_0, p_1, \ldots, p_6 \)
- \( p_0 \), \( p_3 \)
- \( p_6 \)'s 8.1-asvwrite

**8.2-asvw**

- \( p_7 \)
- \( p_7 \)'s 8.2-asvwrite

**2-svword**

DISC '08
Conclusions

- Develop new memory access models for coalesced memory accesses and prove their exact consensus numbers \(N\).
  - size-varying word model, \(b\)-svword, \(b \in [1,B]\).
    - \(N = 3, \forall B \geq 5\)
  - aligned-inconsecutive word model, \(m\)-aiword
    - \(N = |(m+1)/2|\)
  - the combination of these two models, \(m.b\)-asvword, \(b \in [1,B]\).

\[
N = \begin{cases} 
\left\lfloor \frac{m+1}{2} \right\rfloor & \text{if } B = tm, t \in N^* \\
\frac{mB}{2} & \text{if } m = 2tB \\
\frac{(m-B)B}{2} & \text{if } m = (2t+1)B
\end{cases}
\]
Thanks for your attention!