Parallel-agreement is harder than set-agreement

Corentin Travers
Joint work with Zohir Bouzid

DISPLEXITY workshop 2014
Overview

- Set agreement and Parallel agreement
  Generalize the consensus problem

- **Main question**: Relative hardness of Set/Parallel agreement in *message passing, asynchronous, crash prone* system
Processes must *agree* on one of the initial values
On the Consensus Problem

- Asynchronous fault-tolerant consensus is impossible [FLP]

- Work-around
  - Safety:
    - Quorums
    - Majority of non-faulty processes
  - Liveness:
    - Partial synchrony
    - Leader
    - Failure detection
Consensus Generalisations

- **k-set agreement** [Chaudhuri 93]
  - *weak safety*:
  - up to k distinct values can be decided

- **k-parallel agreement** [Afek et al. 10]
  - *weak termination*:
  - k parallel instances of consensus,
    each proc is required to decided in *one of them*
k-set agreement

Agree on at most $k$ values

- $n$ processes $\{p_1, \ldots, p_n\}$
- Initial values $\{v_1, \ldots, v_n\}$

Three properties
- Validity
- Agreement: $\#\text{decision} \leq k$
- Termination
k-parallel agreement

K instances of consensus
Each proc. has to decide in at least one instance

\[(1, \boxed{1}) (2, \boxed{2}) (1, \boxed{1}) (1, \boxed{1}) (2, \boxed{1})\]
k-parallel agreement

- Each proc $pi$ proposes a value $vi$
- Decides a pair $(ci,ui)$ such that

Validity  
$1 \leq ci \leq K$
$ui$ is a proposed value

Agreement  For all $i,j$ : If $ci = cj$ then $ui = uj$

Termination  Every non faulty process decides
On parallel/set agreement

- **k-parallel/set agreement** solvable in asynch. system iff \#failures < \( K \)

- **k-set agreement** : computability benchmark
  - Classification of failures adversary [Gafni Kuznetsov 2011]
  - Smallest \( K \) for which K-set agreement is solvable => insights on what can be computed in a given model

- **k-parallel agreement**
  => K-parallel state machine replication [Gafni Guerraoui Generalized universality 2011]
set-agreement vs. parallel-agreement

- k-parallel agreement implements k-set agreement (at most one decision in each of the k instances)

- k-// agreement and k-set agreement are equivalent in shared memory [Afek et al. 2010]

Message passing model ?
Message passing vs. Shared memory

when \#failures < \#procs/2 [abd 95]
Computationnal model

Asynchronous message passing model
- \( n \) processes asynchronous, may crash
- \( t \): upper bound on \#crash (\( t \geq n/2 \))
- Asynchronous, but reliable communication
k-set vs. k-// agreement in message passing

- $t < \frac{n}{2}$: shared memory can be emulated in the message passing models
  k-set agreement and k-// agreement are equivalent

- $t \geq \frac{n}{2}$: ??
## Results

### Existence of leader-based protocols

<table>
<thead>
<tr>
<th>t</th>
<th>$\frac{n+k-2}{2}$</th>
<th>$\frac{kn}{k+1}$</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
</tr>
</tbody>
</table>

**k-set agreement**

**k-parallel agreement**

[This talk]

[Bouzid T. 10]

### When k-set agreement implements k-parallel agreement

<table>
<thead>
<tr>
<th>t</th>
<th>$\frac{n}{2}$</th>
<th>$\frac{n+k-2}{2}$</th>
<th>$\frac{kn}{k+1}$</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

**k-// A = k-SA**

[k-SA <= k-//A]

[k-SA < k-//A]

[This talk]

[This talk]

[Gafni et al]
Leader

- At each process $p$: $\text{leader}_p$

- *Eventual leadership*:
  Eventually,
  - there is a proc $q$ such that $\text{leader}_p = q$ for all processes
  - $q$ is a non-faulty process
Leader-based parallel agreement

- Let $f$ be a $C$ coloring of the sets of procs of size $n-t$
  s.t. $f(Q) = f(Q')$ implies $Q \cap Q' \neq \emptyset$

Example: $n = 5, t = 3$

$\{p1,p2\} \{p1,p3\} \{p1,p4\} \{p1,p5\}$
$\{p2,p3\} \{p2,p4\} \{p2,p5\}$
$\{p3,p4\} \{p3,p5\} \{p4,p5\}$

Sets with the same color = A quorum system
**C-parallel agreement**

- Let $A_1, \ldots, A_C$ be $C$ instances of a quorum-based, leader-based asynchronous consensus algorithm (i.e. [mostefaoui raynal])

- Instance $A_i$ is associated with quorum system **colored $i$**

  \[
  n = 5, \quad t = 3 \quad A_1 \quad \{p_1,p_2\} \quad \{p_1,p_3\} \quad \{p_1,p_4\} \quad \{p_1,p_5\} \\
  \quad A_2 \quad \{p_2,p_3\} \quad \{p_2,p_4\} \quad \{p_2,p_5\} \\
  \quad A_3 \quad \{p_3,p_4\} \quad \{p_3,p_5\} \quad \{p_4,p_5\}
  \]

- Each proc $p$ participates simultaneously in $A_1, \ldots, A_C$
  - $p$ decides $v$ in $A_i$ => decide $(i,v)$ in parallel agreement
C-parallel agreement (cont'd)

- **A1, ... AC**: C instances of a *quorum-based, leader-based asynchronous* consensus algorithm
- Instance **Ai** is associated with quorum system *colored i*

\[
\begin{align*}
A1 & \quad \{p1,p2\} \quad \{p1,p3\} \quad \{p1,p4\} \quad \{p1,p5\} \\
A2 & \quad \{p2,p3\} \quad \{p2,p4\} \quad \{p2,p5\} \\
A3 & \quad \{p3,p4\} \quad \{p3,p5\} \quad \{p4,p5\}
\end{align*}
\]

**n = 5, t = 3**

**Correctness**
- **Agreement**: At most one value decided in each instance
- **Termination**: One set of (n-t) non-faulty procs colored i, for some 1 <= i <= C, the corresponding Ai terminates
- **Value of C?**
Kneser Graphs KG(n,x)

- **Vertex**: subset $Q$ of $\{1,\ldots,n\}$ of size $x$
- **Edge**: $(Q,Q')$ is an edge iff $Q \cap Q' = \emptyset$

**Chromatic number**

$X(KG(n,x)) = n - 2x + 2$ [Lovasz 78]

$X(KG(5,2)) = 3$
C-parallel agreement

\[ C = \min \ \#\text{colors to color } KG(n,n-t) \]
\[ = X(KG(n,n-t)) = 2t-n+2 \]

**Lemma:** There is a leader-based \( k \)-parallel agreement protocol if \( k \geq 2t-n+2 \), i.e., \( t \leq (n+k-2)/2 \)
**Lemma:** There is no leader-based $k$-parallel agreement protocol for $k < 2t-n+2$, i.e. if $t > (n+k-2)/2$
Lower bound

no leader-based $k$-parallel agreement if $k < 2t-n+2$

**Proof** protocol implies coloring of $KG(n,n-t)$

- $A$: $t$-resilient $k$-parallel agreement protocol
- $Q, Q'$: subset of procs. of size $n-t$
- $e_Q, e_Q'$: execution of $A$ in which only processes in $Q$ (resp. $Q'$) participate

\[ \begin{align*}
Q & \quad p \\
p & \text{decides } (c,v) \\
\text{color}(Q) & = c
\end{align*} \]

\[ \begin{align*}
Q' & \quad p' \\
p' & \text{decides } (c',v') \\
\text{color}(Q') & = c'
\end{align*} \]

If $Q \cap Q' = \emptyset$, $c \neq c'$
\[ 1 \leq c, c' \leq k \]

coloring of $KG(n,n-t)$ hence $k \geq X(KG_n,n-t) = 2t-n+2$
Set-agreement vs. Parallel-agreement

- k-∥ agreement implements k-set agreement (at most one decision in each of the k instances)

- k-∥ agreement and k-set agreement are equivalent in shared memory [Afek et al. 2010]

Conditions on t,k,n for which k-∥ agr. can be implemented from k-set agr. in message passing, t >= n/2 ?
k-parallel agreement is harder than k-set agreement

**Thm:** If \( t > (n+k-2)/2 \), there is **no protocol** that implements \( k-\parallel \) agreement from k-set agreement

**Proof:** Reduction from (impossibility of) failures detectors emulation
Failure detectors [Chandra Toueg 96]

- Distributed oracles that give information on failures
  - Example leader f.d: output at each process a proc. Id s.t. eventually same Id of a correct process is output

- Failure detector D is *sufficient* to solve task T if there is a protocol that uses D for T
  - Leader f.d. is sufficient to solve consensus if \( t < n/2 \)

- Failure detector D is *necessary* to solve task T if for from f.d. D' sufficient for T, one can emulate D
  - Leader f.d. necessary for consensus [Chandra Hadzilacos Toueg 96]
k-parallel agreement is harder than k-set agreement

**Thm:** If $t > (n+k-2)/2$, there is no protocol that implements k-// agreement from k-set agreement.

**Proof:** Reduction to failure detectors emulation

\[
\begin{align*}
&\text{k-Set Agr} & \quad & \quad & \text{k-// Agr} & \quad & \quad & \text{Tasks} \\
&\quad & \quad & A & \quad & \quad & \quad \\
&\quad & \quad & \uparrow \text{sufficient} & \quad & \quad & \quad \\
&\quad & \quad & \text{[Bouzid T. 10]} & \quad & \quad & \quad \\
&\left(\Sigma_k, \Omega\right) & \quad & \quad & \quad & \quad & \quad & \quad & \left(V \Sigma_k\right) \\
&\quad & \quad & B & \quad & \quad & \quad & \quad & \quad & \text{Failure detectors}
\end{align*}
\]
Failure detector definitions

- $\Sigma_k$ at each proc output $Q$ : subsets of $\{p1,\ldots,pn\}$
  - **Liveness**: eventually $Q$ contains only correct proc.
  - **Intersection**: $\forall Q_1,\ldots,Q_{k+1}, \exists i \neq j, Q_i \cap Q_j \neq \emptyset$

- $V \Sigma_k$ at each proc output $V$: vector $[Q_1,\ldots,Q_k]$ of $k$ subsets of $\{p1,\ldots,pn\}$
  - **Liveness**: $\exists c, 1 \leq c \leq k$ eventually $V[c]$ contains only correct proc.
  - **Intersection**: $\forall c, 1 \leq c \leq k, \forall i, j V_i[c] \cap V_j[c] \neq \emptyset$
$\Sigma_k$ cannot emulate $V \Sigma_k$

$\left( \Sigma_k , \Omega \right)$ \quad \xrightarrow{B} \quad V \Sigma_k \quad t > \frac{n+k-2}{2}$

Reduction to a $k$-coloring of the kneser graph $KG(n,n-t)$

$\exists c : V[c] \subseteq Q$
- eventually

Valid coloring:
- If $Q \cap Q' = \emptyset$ "merge" $e_Q$ and $e_Q'$
- $c \neq c'$ (intersection property)
Open question

\[ 0 \leq t < \frac{n}{2} \]

k-set agr. and k-// agr.
are equivalent

\[ \frac{n+k-2}{2} < t \leq n \]

k-// Agr. strictly harder
than k-set agr.

\[ \frac{n}{2} \leq t \leq \frac{n+k-2}{2} \]

????
upper bound: k-set agr. implements (2k-1)-// agr.
Future work

- Computability: what can be computed when a majority of the processes may fail?

- Partition-tolerant algorithms
Thanks !
From k-SC to (2k-1)-PC

Asumption: \( t \leq \frac{n+k-2}{2} \)

\( p_i \in A \) decides \((i, v_i)\) \hspace{1cm} \( p_i \in B \) implements k-PC from k-SC in SM

Liveness: if \( A \cap Correct = \emptyset \), at most \( t - (k-1) < \frac{n-(k-1)}{2} \) failures in B
if \( A \cap Correct \neq \emptyset \), dec. of \( p \in A \) can be adopted by any proc