Brief Announcement:
The Fault-Tolerant Cluster-Sending Problem

Jelle Hellings    Mohammad Sadoghi

Exploratory Systems Lab,
Department of Computer Science,
University of California, Davis, CA, USA
Vision: resilient high-performance data processing

Requirement
Fault-tolerant communication between clusters!
The need for cluster-sending

Definition
The *cluster-sending problem* is the problem of sending a value \( v \) from \( C_1 \) to \( C_2 \) such that:

1. all non-faulty replicas in \( C_2 \) *receive* the value \( v \); 
2. only if all non-faulty replicas in \( C_1 \) *agree* upon sending the value \( v \) to \( C_2 \) will non-faulty replicas in \( C_2 \) receive \( v \); and
3. all non-faulty replicas in \( C_1 \) can *confirm* that the value \( v \) was received.

Solution (crash failures)
Pair-wise broadcasting with \((f_1 + 1)(f_2 + 1)\) messages.
Lower bounds for cluster-sending: Example

\[ n_1 = 15 \quad f_1 = 7 \]
\[ n_2 = 5 \quad f_2 = 2 \]

Claim (crash failures)
Any correct protocol needs to send at least 14 messages.
Lower bounds for cluster-sending: Example

\[ n_1 = 15 \quad f_1 = 7 \]
\[ n_2 = 5 \quad f_2 = 2 \]

Claim (crash failures)
Any correct protocol needs to send at least 14 messages.
Lower bounds for cluster-sending: Example

\[ n_1 = 15 \]
\[ n_2 = 5 \]
\[ f_1 = 7 \]
\[ f_2 = 2 \]

Claim (crash failures)
Any correct protocol needs to send at least 14 messages.
Lower bounds for cluster-sending: Example

\[ n_1 = 15 \quad f_1 = 7 \]
\[ n_2 = 5 \quad f_2 = 2 \]

Claim (crash failures)

Any correct protocol needs to send at least 14 messages.
Lower bounds for cluster-sending: Example

\[ n_1 = 15 \quad \quad \quad f_1 = 7 \]
\[ n_2 = 5 \quad \quad \quad f_2 = 2 \]

Claim (crash failures)
Any correct protocol needs to send at least 14 messages.
Lower bounds for cluster-sending: Example

\[ n_1 = 15 \quad f_1 = 7 \]
\[ n_2 = 5 \quad f_2 = 2 \]

Claim (crash failures)

Any correct protocol needs to send at least 14 messages.
Lower bounds for cluster-sending: Results

Theorem

Assume \( n_1 \geq n_2 \) and let

\[
q = (f_1 + 1) \text{ div } nf_2;
\]

\[
r = (f_1 + 1) \text{ mod } nf_2;
\]

\[
\sigma = qn_2 + r + f_2 \text{ sgn } r.
\]

We need to exchange at least \( \sigma \) messages to do cluster-sending.

▶ Similar results for \( n_1 \leq n_2 \).

▶ Byzantine failures: similar lower bounds on signatures.
An optimal algorithm

---

**Protocol for the sending cluster** $C_1, n_1 \geq n_2, n_1 \geq \sigma$:

1: Choose replicas $\mathcal{P} \subseteq C_1$ with $n_{\mathcal{P}} = \sigma$.
2: Choose a $n_2$-partition partition($\mathcal{P}$) of $\mathcal{P}$.
3: for $P \in$ partition($\mathcal{P}$) do
4: Choose replicas $Q \subseteq C_2$ with $n_Q = n_P$.
5: Choose a bijection $b : P \rightarrow Q$.
6: for $R_1 \in P$ do
7: Send $v$ from $R_1$ to $b(R_1)$.

- Crash failures: $n_1 > 3f_1$ and $n_2 > 3f_2$.
- Byzantine failures:
  - using signatures: $n_1 > 4f_1$ and $n_2 > 4f_2$;
  - using threshold signatures: $n_1 > 3f_1$ and $n_2 > 3f_2$. 
Conclusion

More information
https://jhellings.nl

Paper: https://doi.org/10.4230/LIPIcs.DISC.2019.45