Team-based Message Logging: Preliminary Results

Esteban Meneses, Celso L. Mendes and Laxmikant V. Kalé

Parallel Programming Laboratory
University of Illinois at Urbana-Champaign

Monday, May 17, 2010
Contents

• Message Logging
• Team-based Approach
• Experimental Results
• Conclusions
1253 separate node crashes on Jaguar during 537 days (Aug-22-2008 to Feb-10-2010)
2.33 failures per day

Sequoia will have 1.6 million cores and an exascale machine around 100 million cores...

We will see failures all the time
Charm++

- Object-based over-decomposition.
- An intelligent RTS assigns objects to processors.
Charm++

- Object-based over-decomposition.
- An intelligent RTS assigns objects to processors.
Charm++

- Object-based over-decomposition.
- An intelligent RTS assigns objects to processors.
Charm++

- Object-based over-decomposition.
- An intelligent RTS assigns objects to processors.
Message Logging

- Sender-based.
- Piecewise Deterministic.
- Protocols: Pessimistic and Causal.
Message Logging

- Sender-based.
- Piecewise Deterministic.
- Protocols: Pessimistic and Causal.
Message Logging

Object α

- Sender-based.
- Piecewise Deterministic.
- Protocols: Pessimistic and Causal.

Object β

Resilience 2010, Melbourne, Australia
Message Logging

- Sender-based.
- Piecewise Deterministic.
- Protocols: Pessimistic and Causal.
Message Logging Protocols

**Pessimistic**
- Object $\alpha$
- Object $\beta$
- Object $\gamma$

**Causal**
- Object $\alpha$
- Object $\beta$
- Object $\gamma$

$m$
Message Logging Protocols

**Pessimistic**

Object $\alpha$

Object $\beta$

Object $\gamma$

**Causal**

Object $\alpha$

Object $\beta$

Object $\gamma$
Message Logging Protocols

Pessimistic

Object α

Object β

Object γ

Store m AND #m

Causal

Object α

Object β

Object γ
Message Logging Protocols

Pessimistic

Object α

Object β

Object γ

Causal

Object α

Object β

Object γ

Store $m$ AND $\#m$
Message Logging Protocols

Pessimistic

Object α ——— m ——— Object β

Object γ

Causal

Object α

Object β

Object γ

Store m AND #m

Resilience 2010, Melbourne, Australia

Monday, May 17, 2010
Message Logging Protocols

Pessimistic

Object α

Object β

Object γ

Causal

Object α

Object β

Object γ

Store \(m\) AND \(#m\)
Message Logging Protocols

Pessimistic

Store \( m \) AND \( \#m \)

Object \( \alpha \)

Object \( \beta \)

Object \( \gamma \)

Causal

Store \( m \)

Object \( \alpha \)

Object \( \beta \)

Object \( \gamma \)
Message Logging Protocols

Pessimistic

Object α

Object β

Object γ

Causal

Object α

Object β

Object γ

Store $m$ AND $\#m$

Store $m$

$m$

$m_1$

$m_2$

$m_1 \oplus \#m$
Message Logging Protocols

Pessimistic

- Object $\alpha$
- Object $\beta$
- Object $\gamma$

1. $m$ is sent from $\alpha$ to $\beta$.
2. $m$ is sent from $\beta$ to $\gamma$.
3. $m_1$ is sent from $\gamma$ to $\beta$.
4. $m_2$ is sent from $\gamma$ to $\alpha$.

Causal

- Object $\alpha$
- Object $\beta$
- Object $\gamma$

1. $m$ is sent from $\alpha$ to $\beta$.
2. $m_1 \oplus \#m$ is sent from $\beta$ to $\gamma$.

Store $m$ AND $\#m$

Store $#m$
Message Logging Protocols

Pessimistic

Object α

Object β

Object γ

Causal

Object α

Object β

Object γ

Store \( m \) AND \( \#m \)

Store \( m \)

\( m \)

\( m_1 \)

\( m_2 \)

\( m_1 \oplus \#m \)

Store \( \#m \)

Resilience 2010, Melbourne, Australia

Monday, May 17, 2010
Message Logging Protocols

Pessimistic

Object $\alpha$

$m$

Object $\beta$

$m_1$

Object $\gamma$

$m_2$

Store $m$ AND $\#m$

Causal

Object $\alpha$

$m$

Object $\beta$

$m_1 \oplus \#m$

Object $\gamma$

$m_2 \oplus \#m$

Store $\#m$
Virtualization
Virtualization

- Higher virtualization ratio:
- Hides latency overhead.
- Increases number of objects and messages.
Team-based Approach

- Goal: reduce memory overhead of message log.
- Only messages crossing team boundaries are logged.
Team-based Approach

- Goal: reduce memory overhead of message log.
- Only messages crossing team boundaries are logged.
Team-based Approach

- Goal: reduce memory overhead of message log.
- Only messages crossing team boundaries are logged.
Processor Teams

• Each team acts as a recovery unit:
  • All members checkpoint in a coordinated fashion.
  • If one member fails, the whole team rolls back.
Processor Teams

- Each team acts as a recovery unit:
  - All members checkpoint in a coordinated fashion.
  - If one member fails, the whole team rolls back.

$k \leq N$

Team Size
Processor Teams

- Each team acts as a recovery unit:
  - All members checkpoint in a coordinated fashion.
  - If one member fails, the whole team rolls back.
Processor Teams

• Each team acts as a recovery unit:
  • All members checkpoint in a coordinated fashion.
  • If one member fails, the whole team rolls back.
Processor Teams

- Each team acts as a recovery unit:
  - All members checkpoint in a coordinated fashion.
  - If one member fails, the whole team rolls back.

Message Logging

Checkpoint/Restart

Team Size

1 \leq k \leq N

Resilience 2010, Melbourne, Australia
Recovery
Recovery

Object $\alpha$

Object $\beta$

Team X

Object $\gamma$

Team Y

$C$

$C$

$m$

$m_1$

$m_2$

$m_3$
NPB-CG

NPB-CG (Abe, p=512, class=D)

73% reduction in message log size

Memory Overhead (MB)

Iteration

Team size = 1
Team size = 16

Resilience 2010, Melbourne, Australia
NPB-MG (Abe, p=512, class=D)

Memory Overhead (MB)

Iteration

Team size = 1
Team size = 16

Resilience 2010, Melbourne, Australia
NPB-MG (Abe, p=512, class=D)

51% reduction in message log size

Team size = 1
Team size = 16

Resilience 2010, Melbourne, Australia
Communication Pattern

Reproduced with permission from http://proactive.inria.fr

Resilience 2010, Melbourne, Australia
Communication Pattern

CG Communication pattern C 64
Message density distribution (CG version: 1.1)

MG Communication pattern C 64
Message density distribution (MG version: 1.0)

Reproduced with permission from http://proactive.inria.fr

Resilience 2010, Melbourne, Australia
Communication Pattern

CG Communication pattern C 64
Message density distribution (CG version: 1.1)

MG Communication pattern C 64
Message density distribution (MG version: 1.0)

Reproduced with permission from http://proactive.inria.fr

Resilience 2010, Melbourne, Australia
Communication Pattern

CG Communication pattern C 64
Message density distribution (CG version:1.1)

MG Communication pattern C 64
Message density distribution (MG version:1.0)

Reproduced with permission from http://proactive.inria.fr

Resilience 2010, Melbourne, Australia
Communication Pattern

CG Communication pattern C 64
Message density distribution (CG version: 1.1)

MG Communication pattern C 64
Message density distribution (MG version: 1.0)

Reproduced with permission from http://proactive.inria.fr

Resilience 2010, Melbourne, Australia

Monday, May 17, 2010
Communication Pattern

Reproduced with permission from http://proactive.inria.fr

Resilience 2010, Melbourne, Australia
Communication Pattern

Reproduced with permission from http://proactive.inria.fr

Resilience 2010, Melbourne, Australia
Message Log Reduction

NPB-CG (CLASS=C, p=64)

Average Logged Messages

Team Size

NPB-MG (CLASS=C, p=64)

Average Logged Messages

Team Size

Resilience 2010, Melbourne, Australia

Monday, May 17, 2010
Stencil 3D

Jacobi (Abe, p=256, n=1536, b=64)

Memory Overhead (MB)

Iteration

Team size = 1
Team size = 4
Team size = 16

Resilience 2010, Melbourne, Australia
Stencil 3D

Jacobi (Abe, p=256, n=1536, b=64)

51% reduction in message log size

Team size = 1
Team size = 4
Team size = 16

Iteration

Memory Overhead (MB)
Recovery Time

![Bar chart showing recovery time by team size]

- **Recovery Time (seconds)**
  - 1 team: 3 seconds
  - 4 teams: 4 seconds
  - 16 teams: 5 seconds

*Resilience 2010, Melbourne, Australia*
Recovery Time

30% increase in recovery time

![Bar chart showing recovery time vs team size]
Research Questions

• Highly connected objects should belong to the same team.

• Exploit communication graph, dynamic groups, team-aware load balancer.

• Processor teams vs object teams?

• Overlapping teams?
Conclusions

• Team-based approach can substantially reduce the memory overhead.

• **Contribution**: team size acts as a middle point between two traditional techniques.

• It can be used in conjunction with major message logging protocols.
Future Work

- Enrich Team-based Approach.
- Smarter team formation.
- Coupling with load balancer.
- Dealing with correlated failures.
- SMP-aware fault tolerance.
- Larger Charm++ applications.
Acknowledgments

- **US Department of Energy, FastOS Program** (Colony-1 and Colony-2 projects).

- **NSF/NCSA**: Deployment efforts specific for Blue Waters.

- **Machine allocation**: TeraGrid MRAC: NCSA, TACC, ORNL.

- **Greg Bronevetsky** from LLNL.

- **Franck Cappello** from INRIA/Illinois.
Q&A