Stabilizing Pipelines for Streaming Applications

Presented by Andrew Berns

Co-authored with Dr. Sukumar Ghosh and Dr. Anurag Dasgupta
The University of Iowa
Outline

• Motivation
• Self-Stabilization
• Linear Pipelines
• Other Compositions
Motivation

• Streaming data in distributed systems are abundant.
• What is the guarantee that a distributed system that handles streaming data will stabilize and exhibit the correct behavior?
• We focus on modular architecture of systems handling streaming data.
Outline

- Motivation
- Self-Stabilization
- Linear Pipelines
- Other Compositions
Expected Pipeline Behavior

For each input $x$ from a constant input stream, the pipeline computes $f(x)$
Regardless of the initial state of the system, the output stream will have a suffix identical to that which will be produced by the correctly initialized system
Outline

• Motivation
• Self-Stabilization
• Linear Pipelines
• Other Compositions
A stage $i$ is composed of $k > 0$ processes, and eventually computes $f_i(x)$ for all inputs $x$. 
Stabilizing Linear Pipeline

\{\text{Program for stage } i : 1 \leq i \leq k\}
\textbf{do} \quad (v_{i-1} \neq v_i) \land (v_{i+1} = v_i) \rightarrow
\quad B_i := f_i(B_{i-1}); \quad v_i := \neg v_i;
\textbf{od}
A linear pipeline converges in at most:

\[ k(k-1)(1/2) + k(L_{\text{max}} - 1) + 1 \]

time steps.
Outline

- Motivation
- Self-Stabilization
- Linear Pipelines
- Other Compositions
Alternative Composition

Pipeline 1

Pipeline 2

selector

join

$v_0$ $v_s$ $v_s'$ $v_a$ $v_j'$ $v_j$ $v_b$ $v_k$
Stabilizing Alternative Composition

- The selector stage may “starve” one of the pipelines
- To be self-stabilizing, all executions of the selector of length $m$ must include at least one output to each pipeline
Alternative Pipeline Convergence
Time

An alternative pipeline converges in at most:
\[ t(t-1) + mtL_{\text{max}} + 1 \]
time steps.
Concurrent Composition

Pipeline 1

Pipeline 2

... 

Pipeline r

fork

join
Concurrent with Boolean Signals
Stabilizing Concurrent Composition

Pipeline 1

Pipeline 2

Pipeline r

fork

join

$w_u$

$v_u$

$w_l$

$w_2$

$w_r$

$v_j$

$w_j$
Stabilizing Concurrent Composition

$w_u = 5$

Pipeline 1

$w_j = 3$

$w_1 = 4$

Pipeline 2

$w_r = 2$

Pipeline r

fork

0

join

1

1
Repetitive Composition

$v_0 \xrightarrow{} loopback \xrightarrow{} Pipeline 1 \xrightarrow{} iterator \xrightarrow{} v_k$
Stabilizing Repetitive Composition

• Similar to the alternative composition, we have to make sure the iterator doesn’t “starve” the environment
Final Points

• Any of these stabilizing compositions can be replaced with any other stabilizing composition
• Results are possible with bounded sequence numbers
Thank You!
Stabilizing Pipelines for Streaming Applications

by Andrew Berns,
Dr. Sukumar Ghosh,
Dr. Anurag Dasgupta