Models of Streaming Applications

Filip Nálepa

Outline

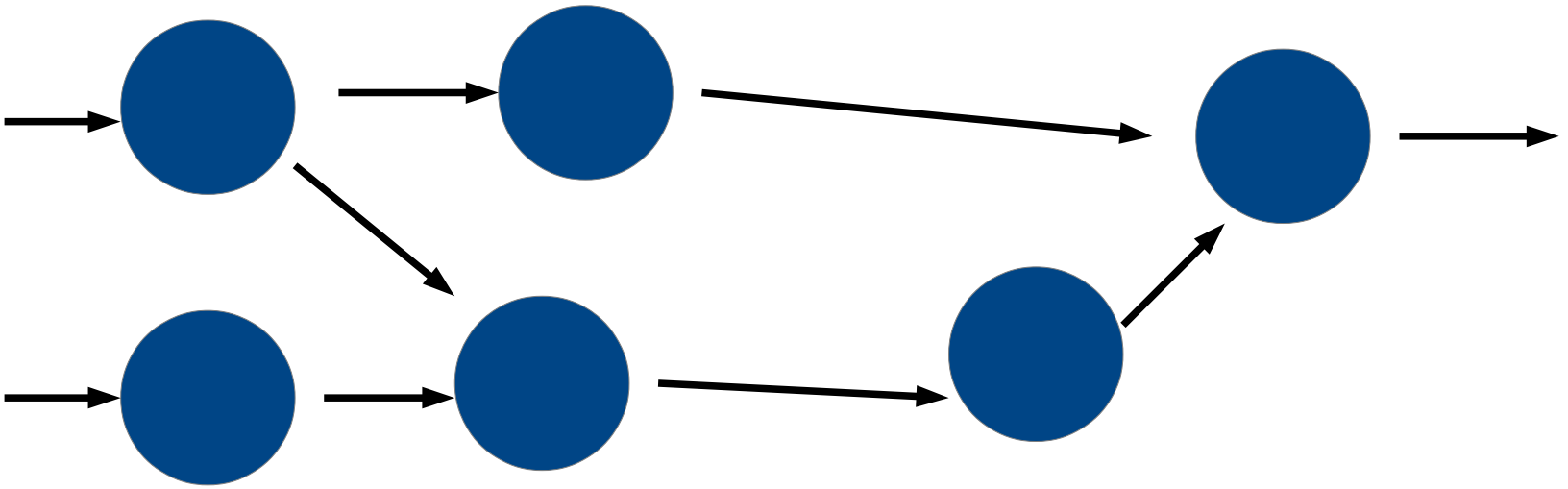
- Motivation
- Components of a streaming application
- Operator placement problem
- Performance models

Motivation

- Infinite sequence of data
- Processing data in motion
- Scenarios
  - Event detection
  - Image stream processing
  - Surveillance video analysis

Staticflickr.com

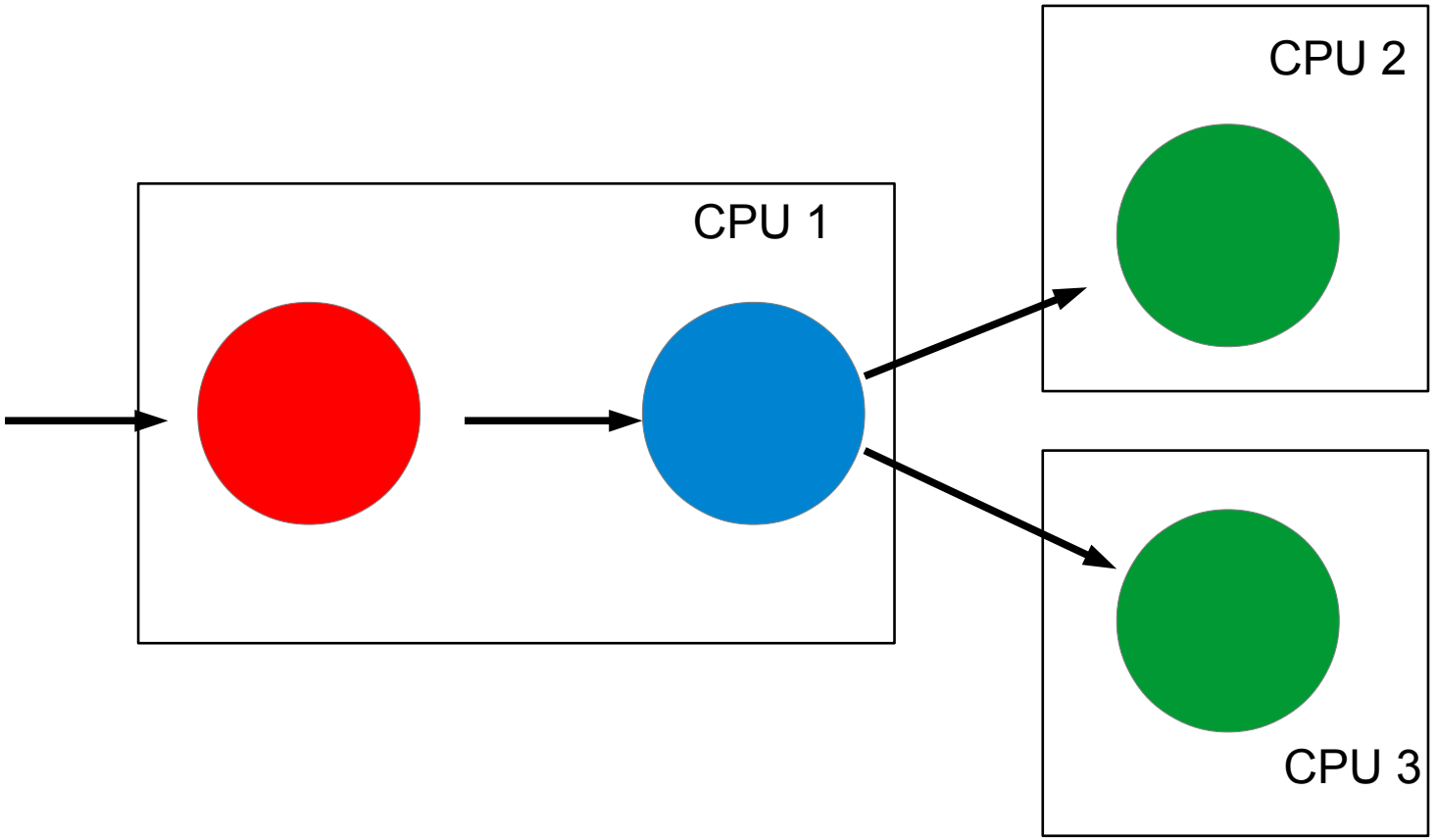
Streaming Application as a Graph



- Node – operator/task
- Edge – stream
- Stream – infinite sequence of data items/events

Operator Placement

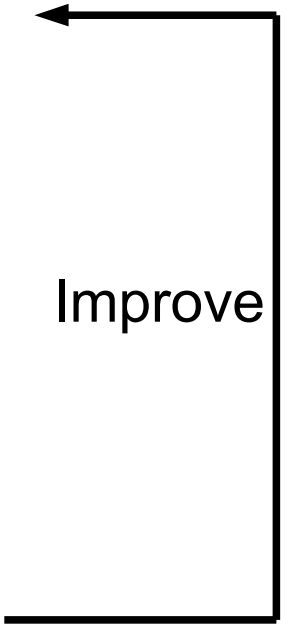
- Assignment of operators to computational resources
- Metrics: throughput, latency



Needed Models

- Model of operator placement problem
  - Purpose: place operators on resources

- Performance model
  - Purpose: retrieve metrics of the system



FINISH

Models Use Cases

- Initial operator placement
  - Placement and measurement
- Dynamic adaption to changes
  - Change/problem detection – proactive/reactive
  - New placement and verification

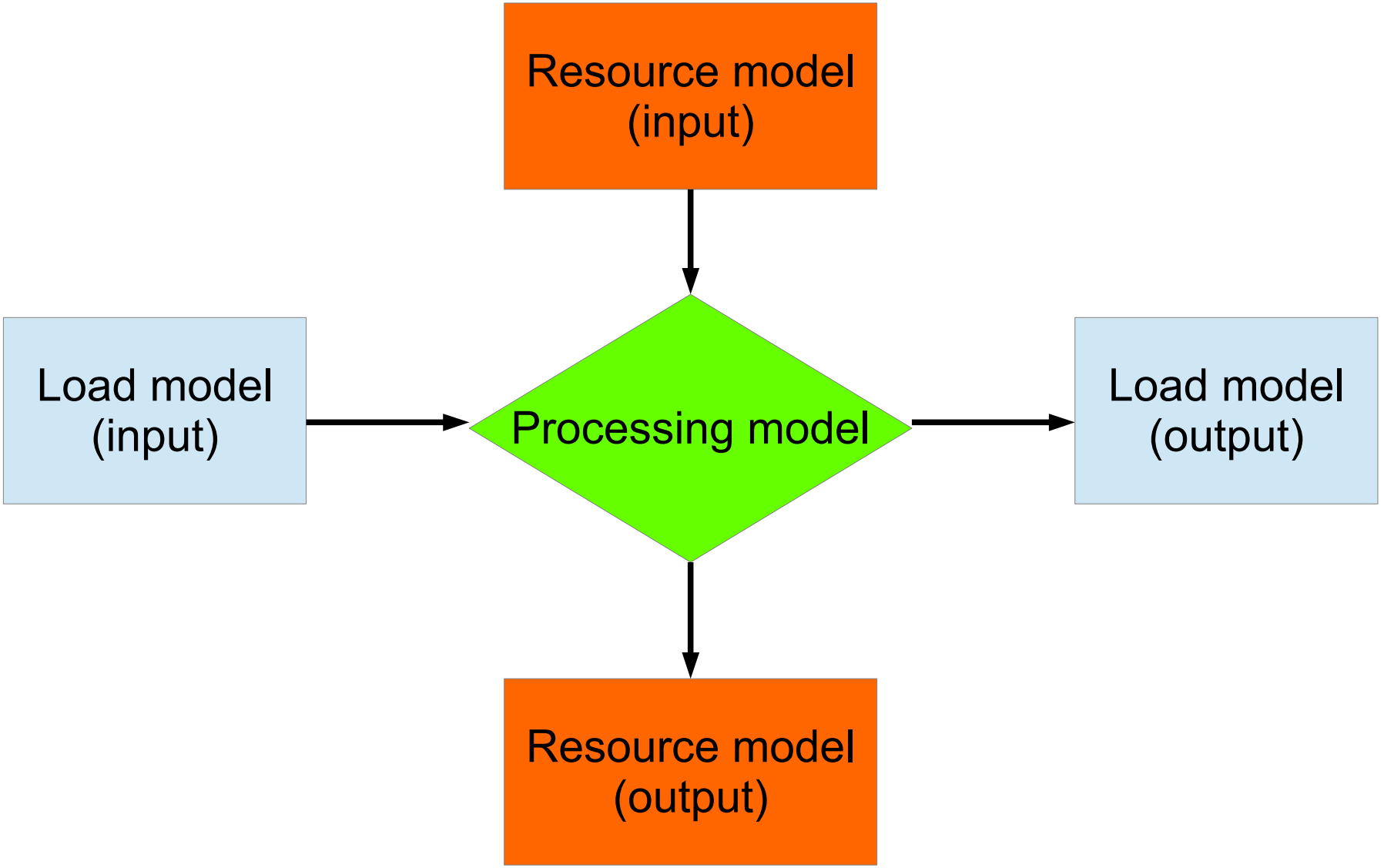
Model of Operator Placement Problem

- Computational resources, underlying network
- Streaming graph, operators, streams
- Optimization criteria
- Purpose: place operators on resources

Performance Analysis

- Accuracy vs efficiency
- Simulation and experiments
- Formal methods

Performance model



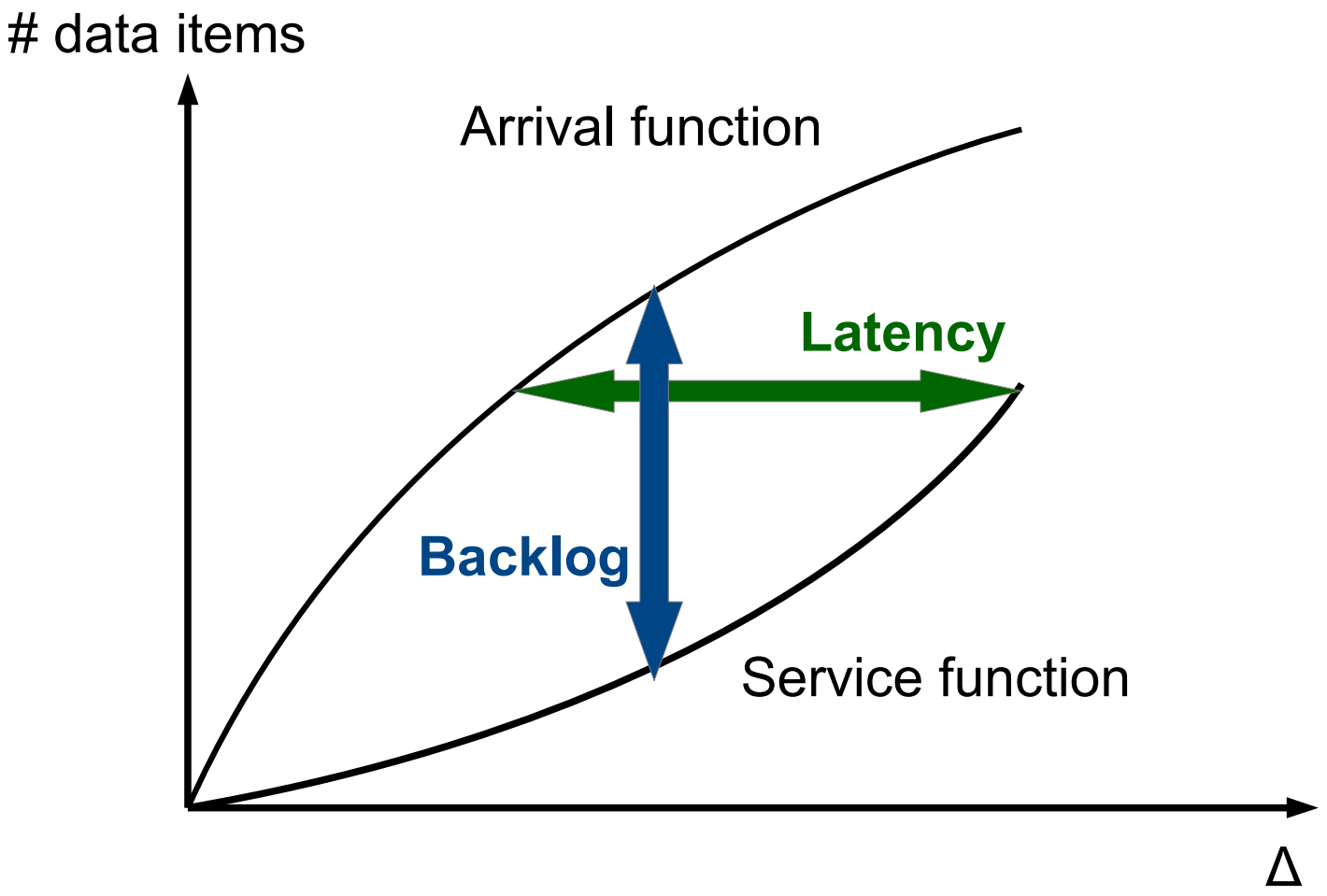
Standard Event Models

- Periodic
- Periodic with jitter
- Burst – period, maximal number of items, minimal distance between items
- Sporadic – minimal distance between items
- Advantages: simple, easy to analyze
- Disadvantages: too restrictive, unrealistic assumptions

Real-Time Calculus

- Arrival function $\alpha(\Delta)$ – maximal number of data items that can arrive in any time interval of length Δ
- Service function $\beta(\Delta)$ – minimal number of data items that can be processed in any time interval of length Δ

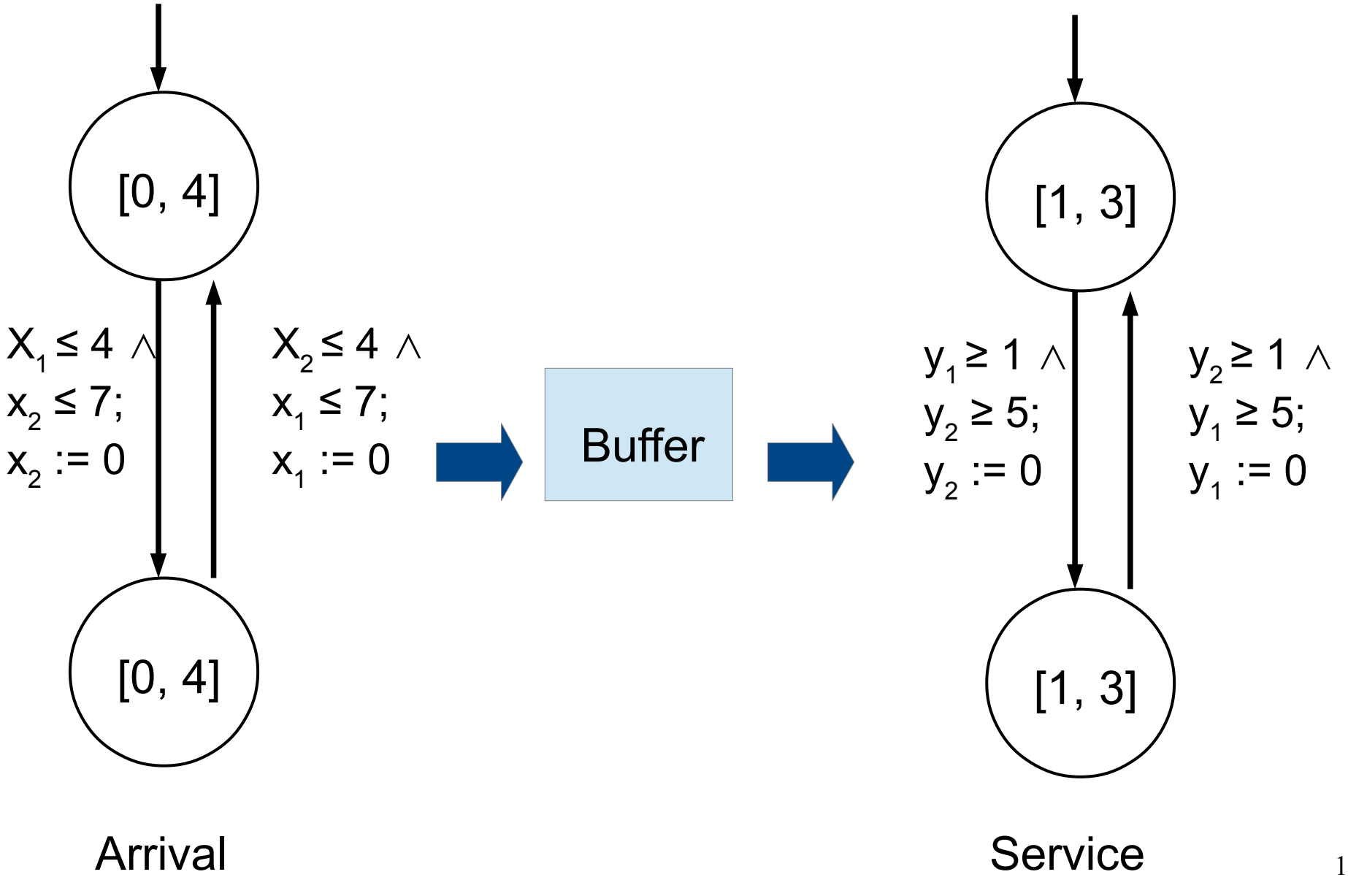
Arrival and Service Function



Real-Time Calculus

- Arrival function $\alpha(\Delta)$ – maximal number of data items arrived in any time interval of length Δ
- Service function $\beta(\Delta)$ – minimal number of data items that can be processed in any time interval of length Δ
- Analysis based on algebraic computations
- Advantage: efficient
- Disadvantage: no state dependencies

Event Count Automata



Event Count Automata

- Arrival and service function represented as automata
- Automata connected by buffers
- Network of automata described as a Colored Petri Net for analysis
- Advantage: very accurate
- Disadvantage: state-space explosion → inneficient

Performance Analysis Summary

- Simulation – easy to use, no guarantees
- Standard event models – simple, not accurate
- Real-time calculus – efficient, captures burstiness, no state dependencies
- Event count automata – accurate, not efficient
- Combinations – tradeoffs

Summary

- Streaming application – directed graph of operators and streams
- Operator placement problem
- Performance models

Thank you for your attention.