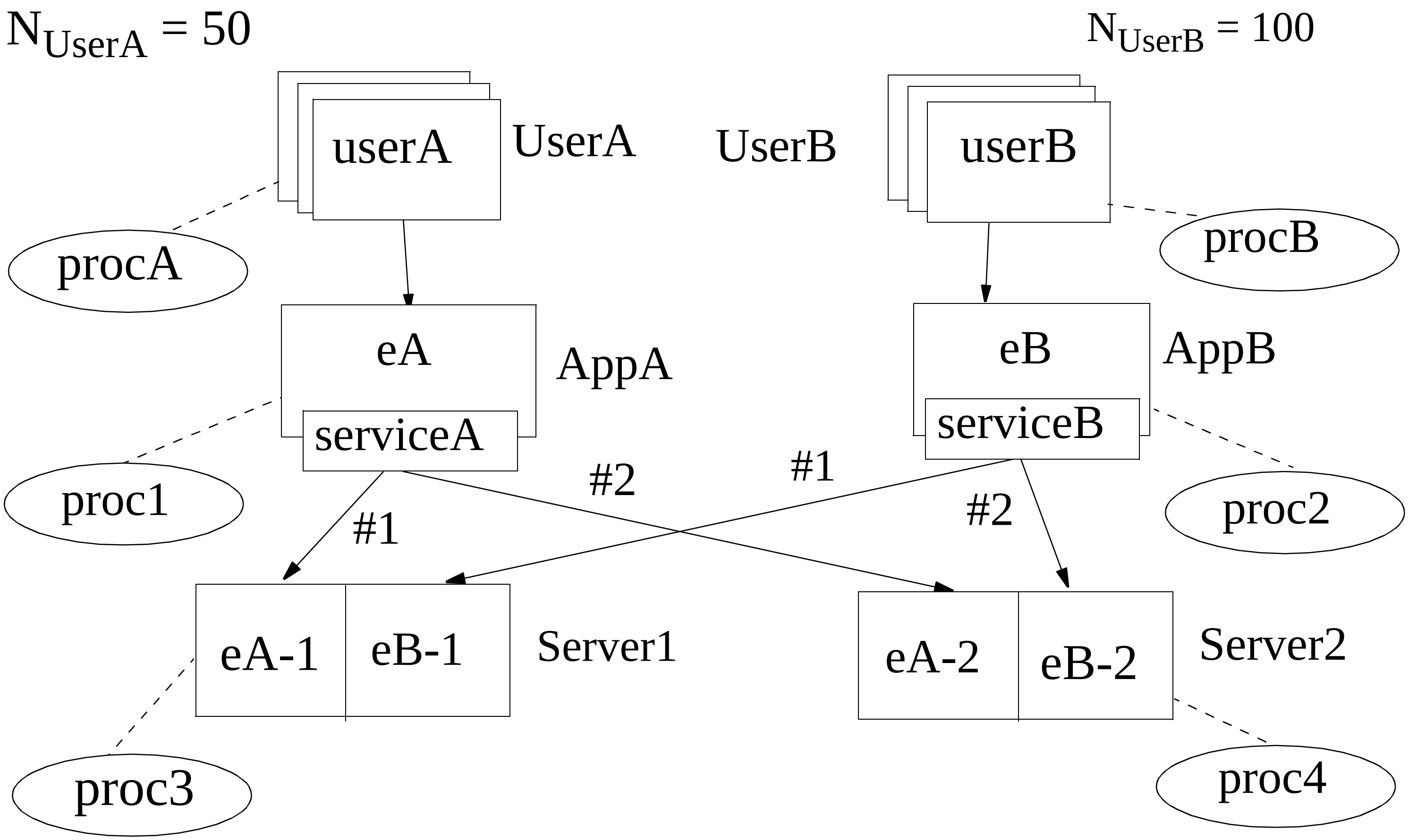
Evaluation of Dependable Layered Systems with a Fault Management Architecture

Olivia Das, C. Murray Woodside
Dept. of Systems and Computer Engineering, Carleton University, Ottawa, Canada
email: odas@sce.carleton.ca, cmw@sce.carleton.ca

Layered System Model

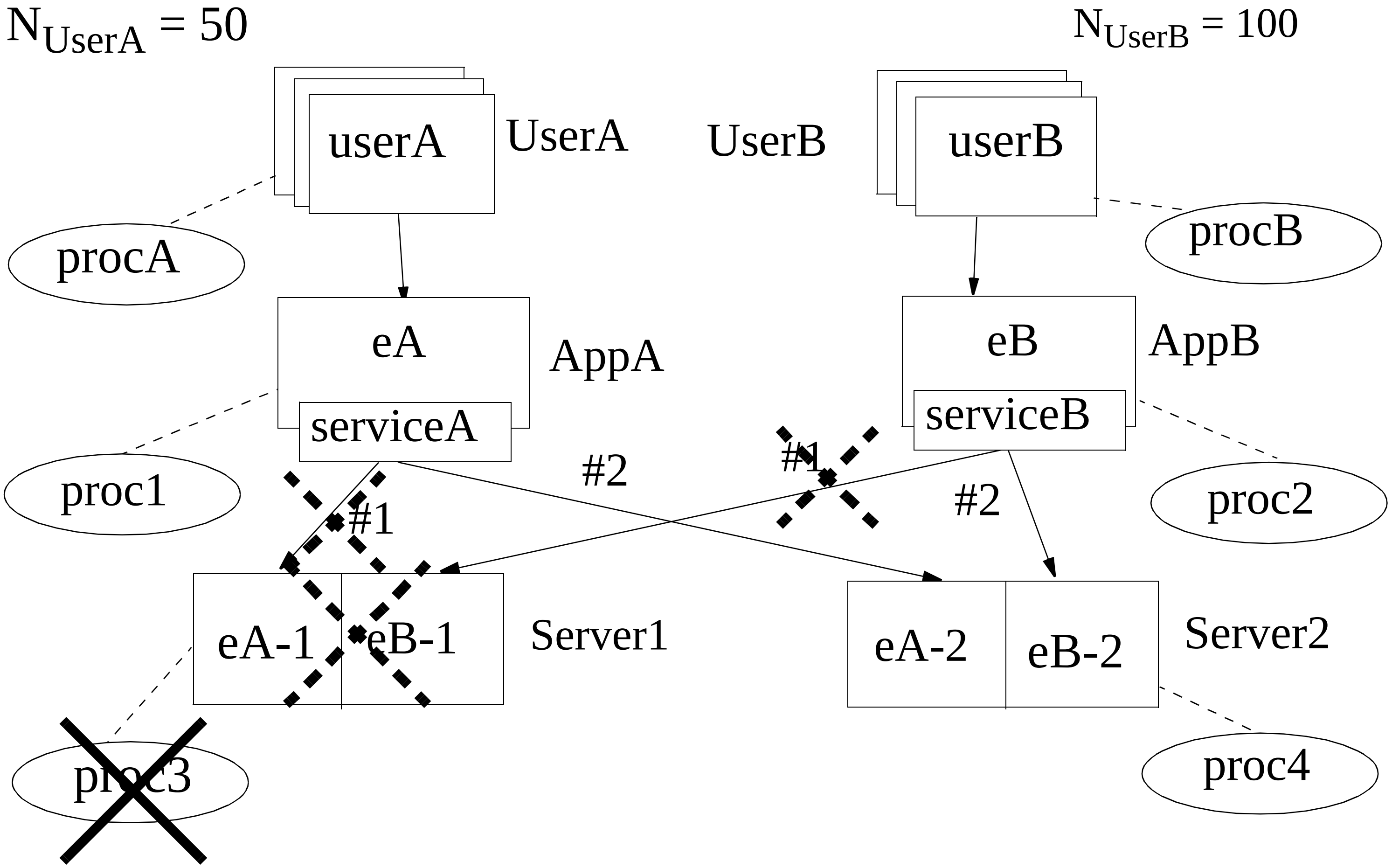
Tasks, Interactions and Dependencies, and Processors



......Configuration depends on Failure State

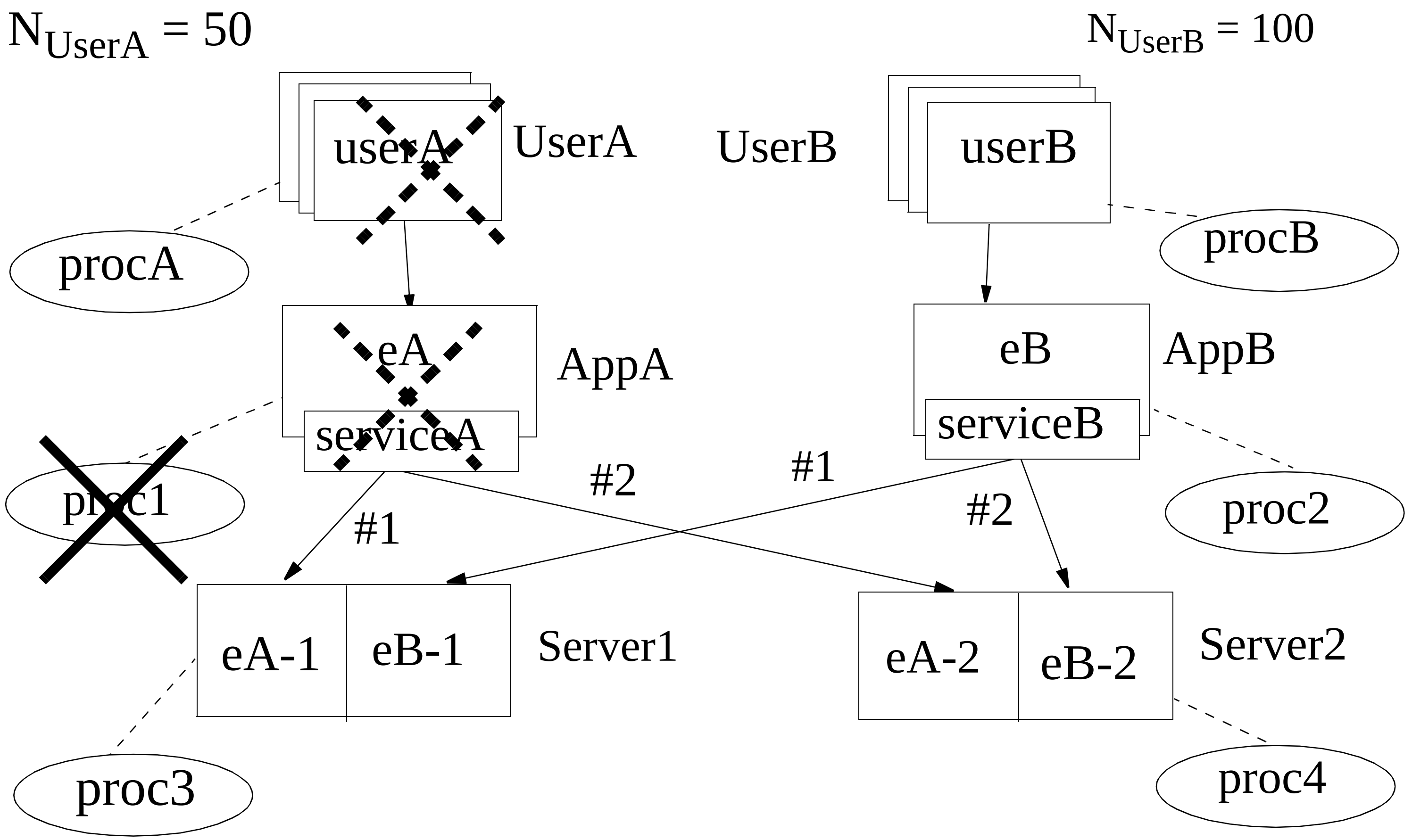
Example Configuration (1)
... failure compensated by standby servers

Processor 3 fails and puts Server1 out... Server2 used instead



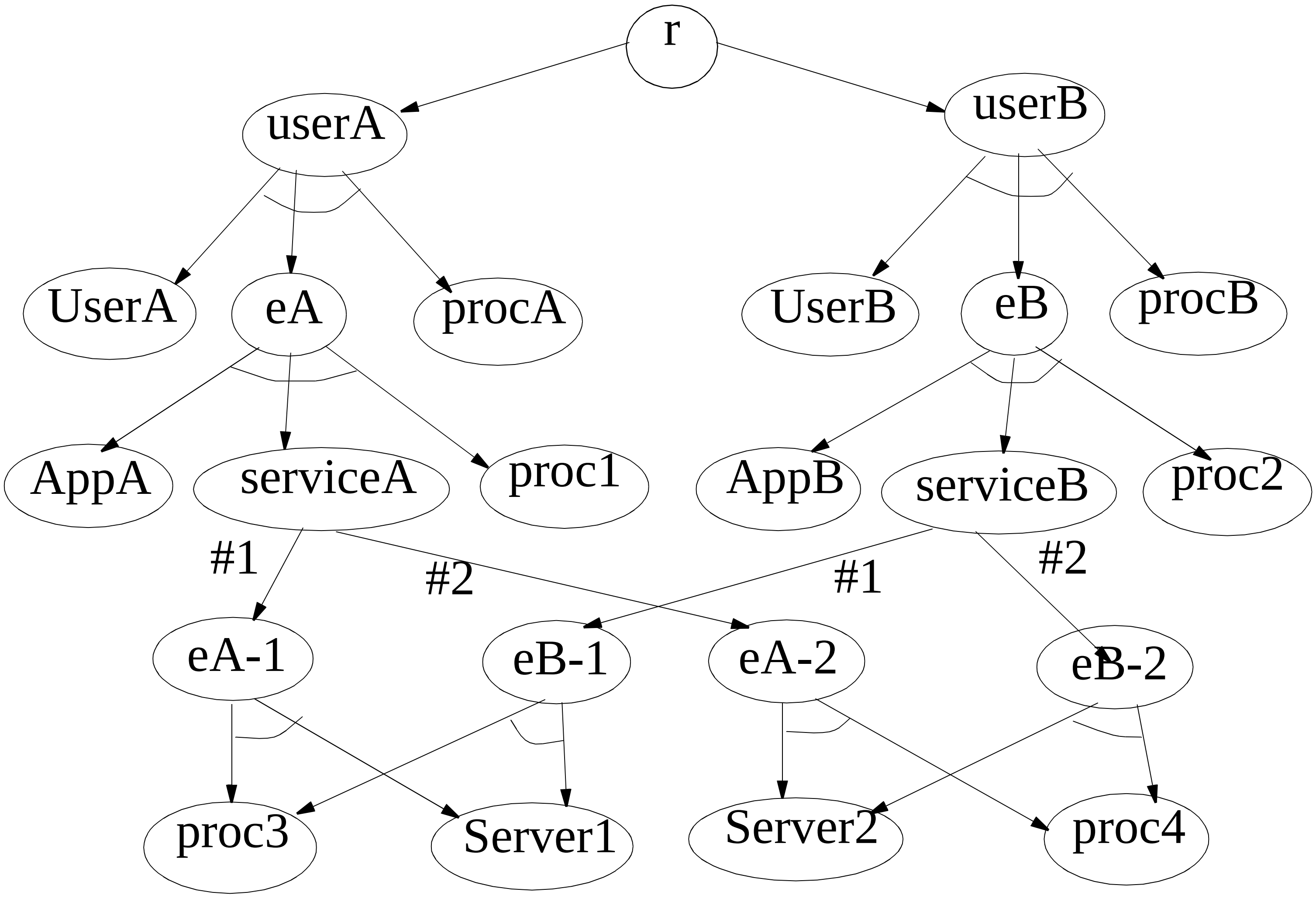
Example Configuration (2)
... failure cannot be compensated by standby servers

Processor 2 fails and puts Application1 out... Group Users1 is off the air.... performability measure is reduced

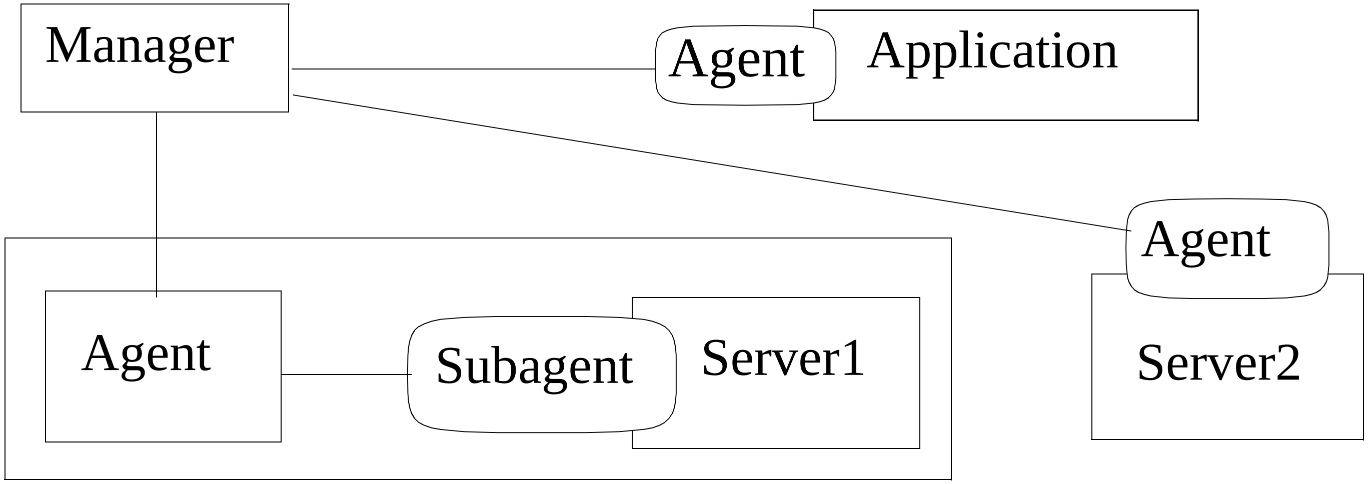


Fault Propagation Graph....

used to find the configuration states, add up their probabilities



Management Subsystem



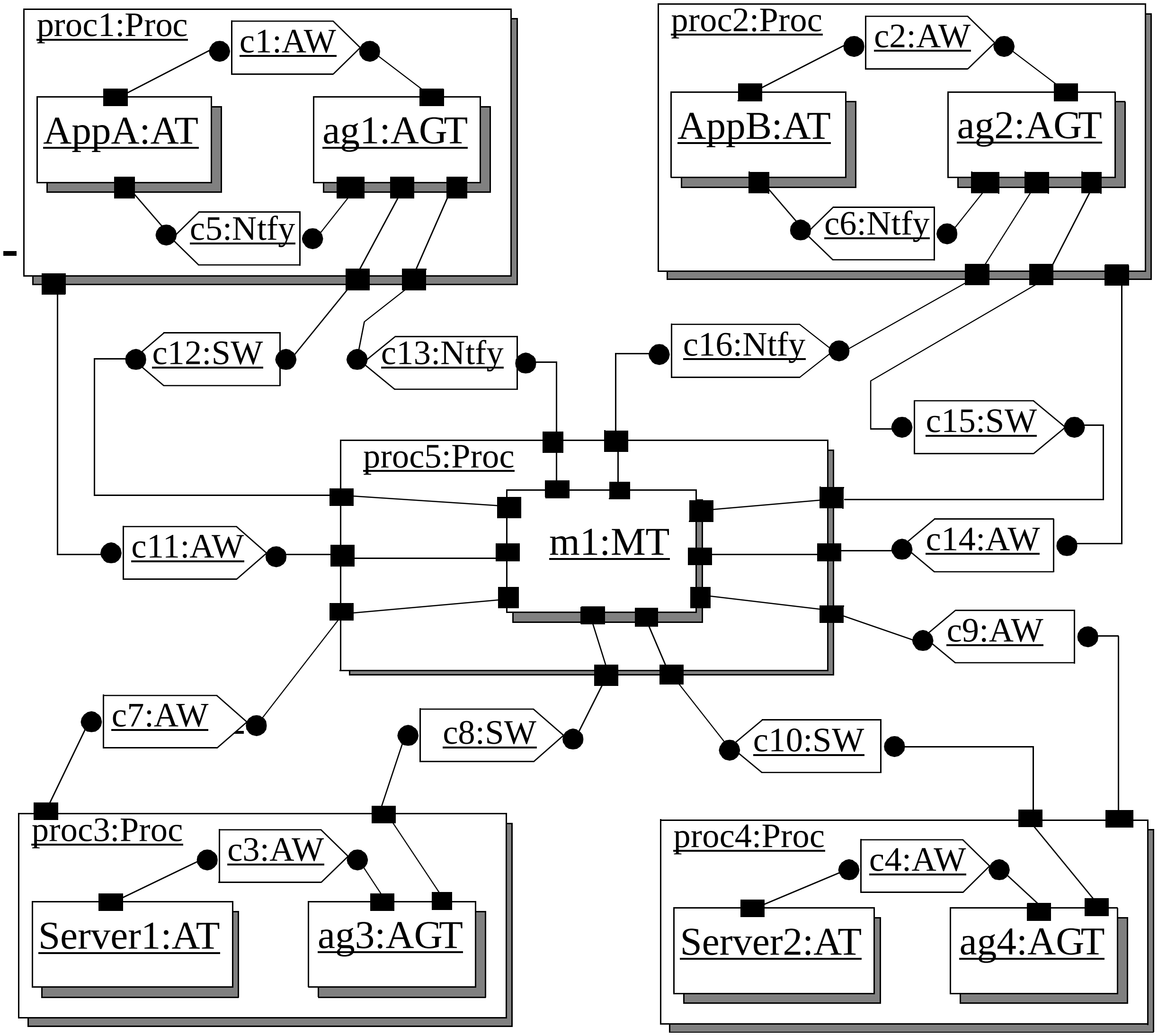
- Reaction delays
- Management subsystem failures and repairs

Specifying a Management Architecture

Elements

Components
- **Application processes**
- **Management Agents**
- **Managers**

Connectors
- **Alive-watch**
- **Status-watch**
- **Notifier**



Functionality

Application process status is monitored by its local agent (Alive-watch connection)

Processor status is monitored by a Manager on another node,

... e.g. by pinging

System wide status is gathered by Managers (Status connections)

.... and distributed back to Agents (Notify connections)

Application process reconfiguration is triggered by the agent on its node (Notification connection)

.... e.g. to switch to a standby server, or to restart a process

Capability to reconfigure is conditioned by “Knowledge” of the status of the system

.... that is, by the Management Architecture and its failures

Analysis.... currently....

* Markov model for component failures and repairs

.... (e.g., independent failure of processors and processes)

* Derive configurations and their probabilities

....Additional configurations that include Management Subsystem failure

* Reconfiguration capability is limited by “Knowledge” of the status, and thus by the Management Subsystem state

.... thus, additional delays to repair

* Analyse the performance of each configuration

.... assemble measures based on configuration probabilities

.... related to work by Haverkort with queueing models and server failures

.... here, extended with *layered dependencies* for failure, and *layered queuing models* for performance

* Consider bounds and approximations

Conclusions

Scalable technique
... separation of performance-level analysis from failure repair
... analysis of effective configurations gives a MUCH smaller set of configurations, than of failure states.

Even so, explosion of configurations is a limitation....

Publications..... www.sce.carleton.ca/faculty/woodside