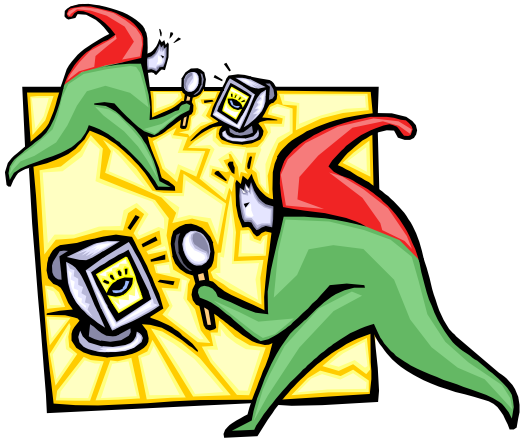
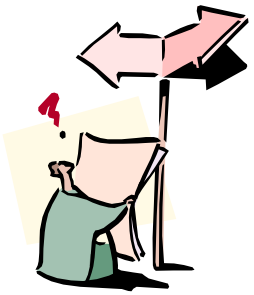
Distributed Execution with Remote Auditing



Fabian Monrose, Avi Rubin & Peter Wyckoff



Motivation

goals

Related work

Overview

System Design

Performance

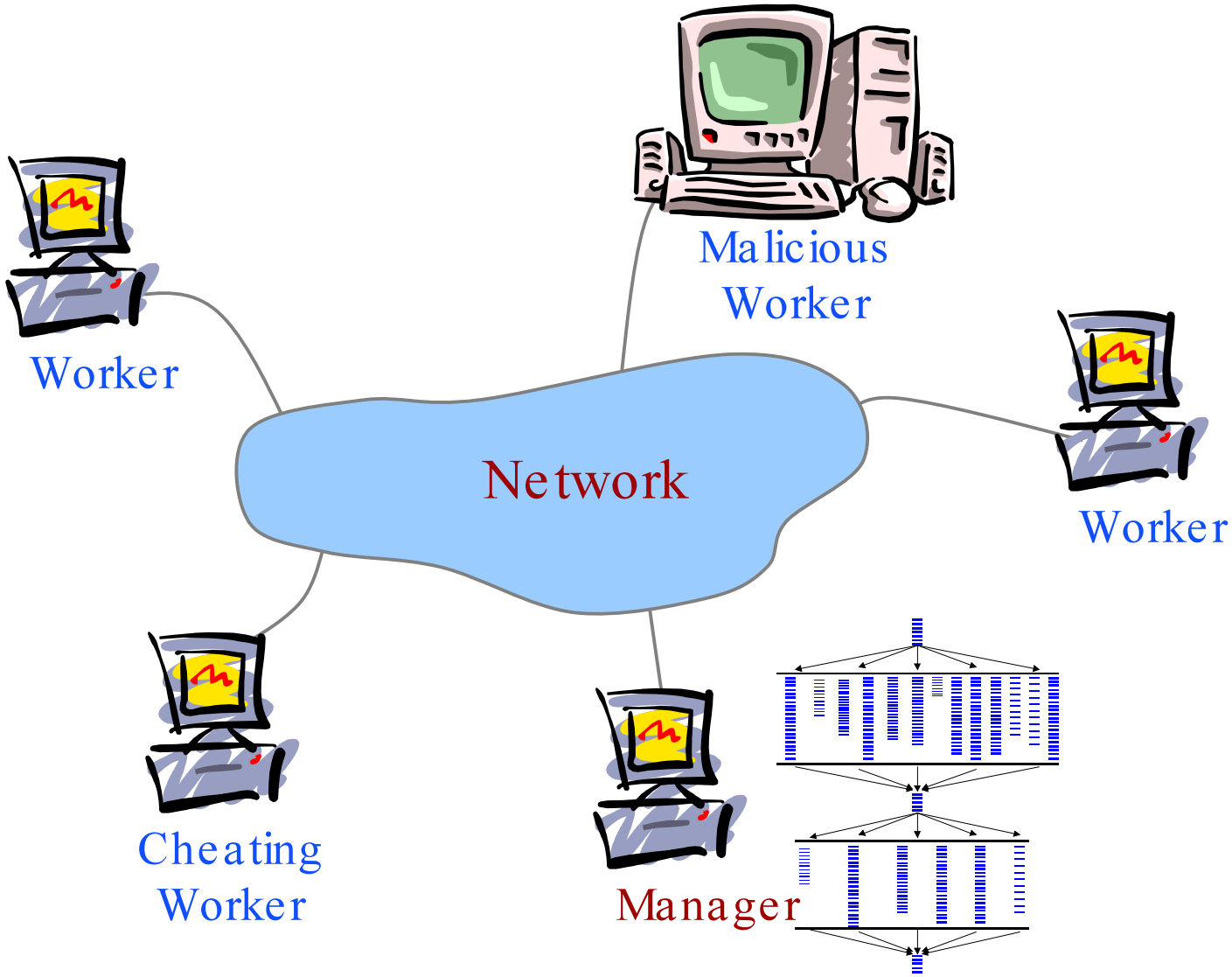
Conclusion

Motivation

- Proliferation of collaborative computing environments that support the execution of coarse-grain parallel computations on anonymous machines connected via the Internet (e.g., Charlotte, Atlas, Javelin, ParaWeb, Bayanihan, Popcorn).

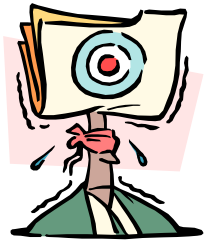
- To promote large scale participation non-altruistic market-based schemes have been proposed, but efficient mechanisms for verifying the work performed by the participants have not been addressed.

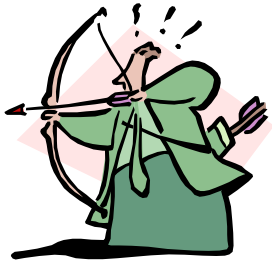
Computing Environment

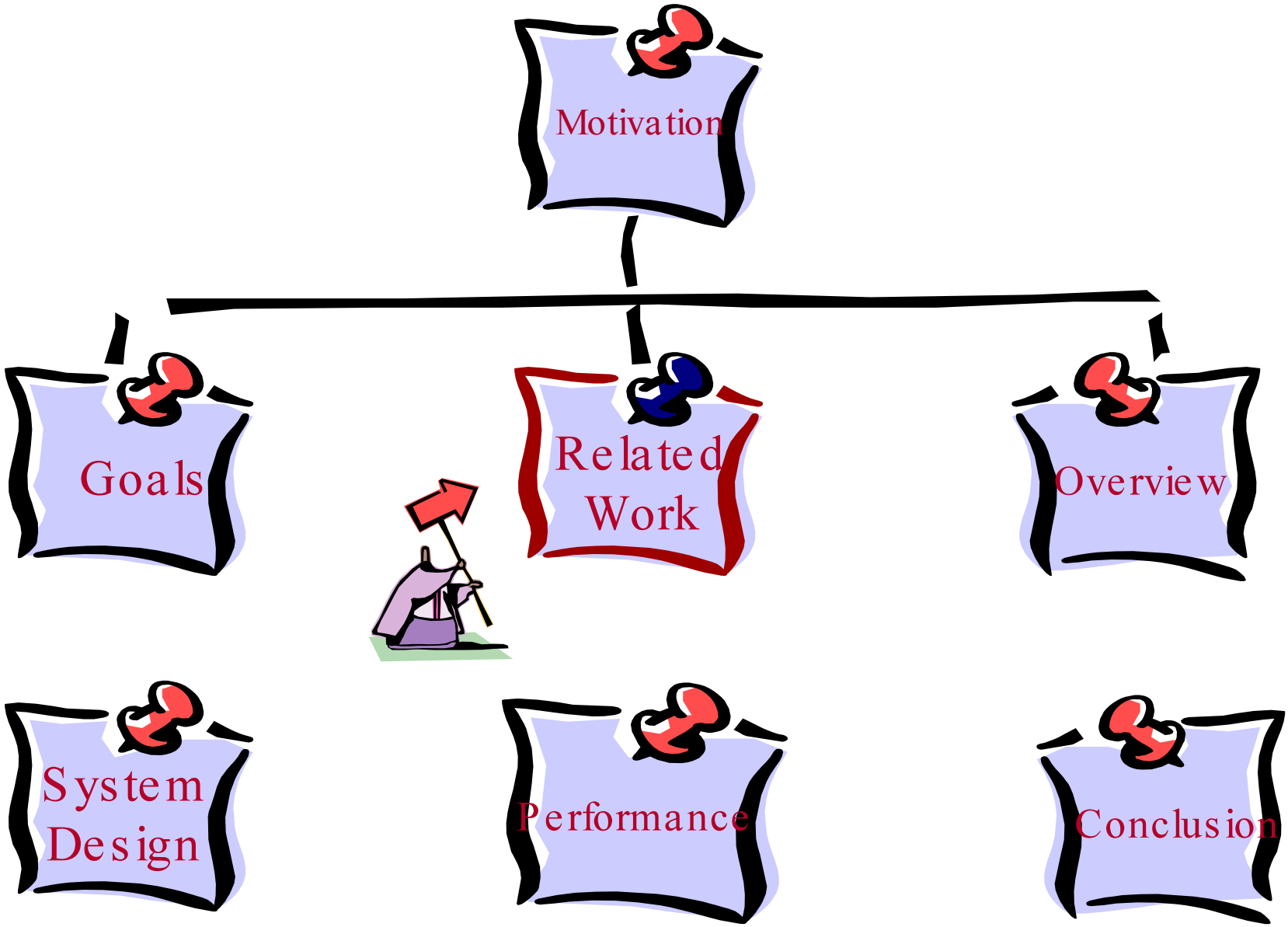


Goals

- Transparency: Integration with existing metacomputing infrastructures should not require extensive effort.
- Efficiency: Verification should require significantly less time than executing the entire task.
- Robustness: “Cheating” workers should be caught with high probability.






Motivation
Goals
Related Work
Overview
System Design
Performance
Conclusion

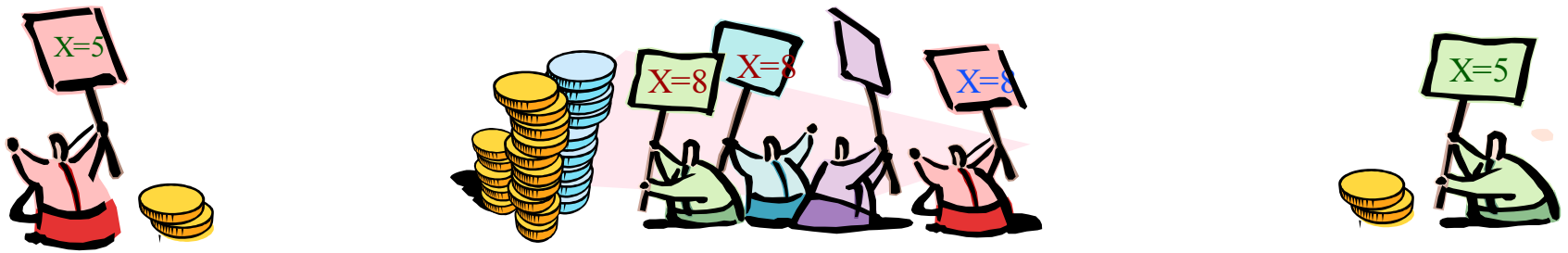
Related Work

- Efficient decompilation techniques (e.g., Mocha) exist, with typical decompilation time on the order of milli-seconds.

- "What can be done can be undone".

Redundant Computation with Online Voting

- Inefficient ---- voting schemes incur network overhead.
- Collaborators can easily sway results.
- Waste of computational resources.



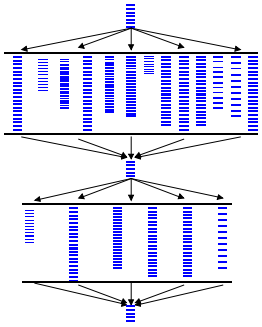
Computing with Secrets or E.F.

Detecting misbehavior through mechanisms that rely on inserting keys within active content is inadequate:

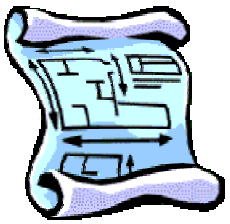
- Since the content distributed must be readable by a potentially large group of hosts, secrets are readable and thus can be easily recovered or reused.

- Encryption schemes with the necessary homomorphic properties (for CEF) do not exist.

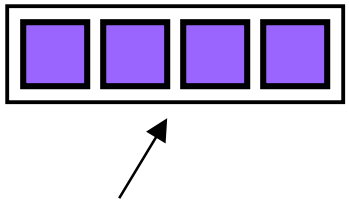
High Level Overview



SPMD
Task

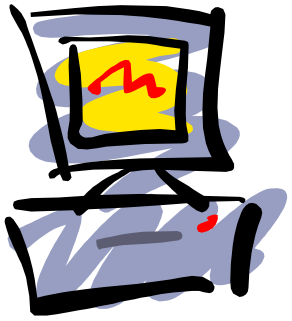


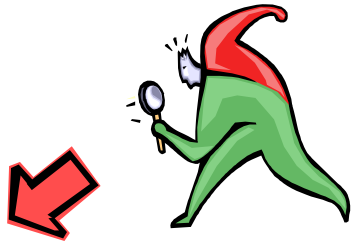
Program
Analysis



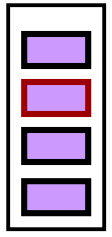
Transformed
components

Worker

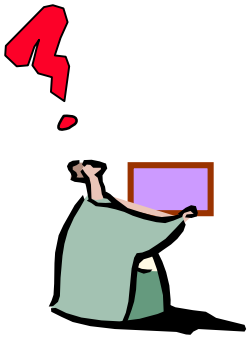




Remote Agent

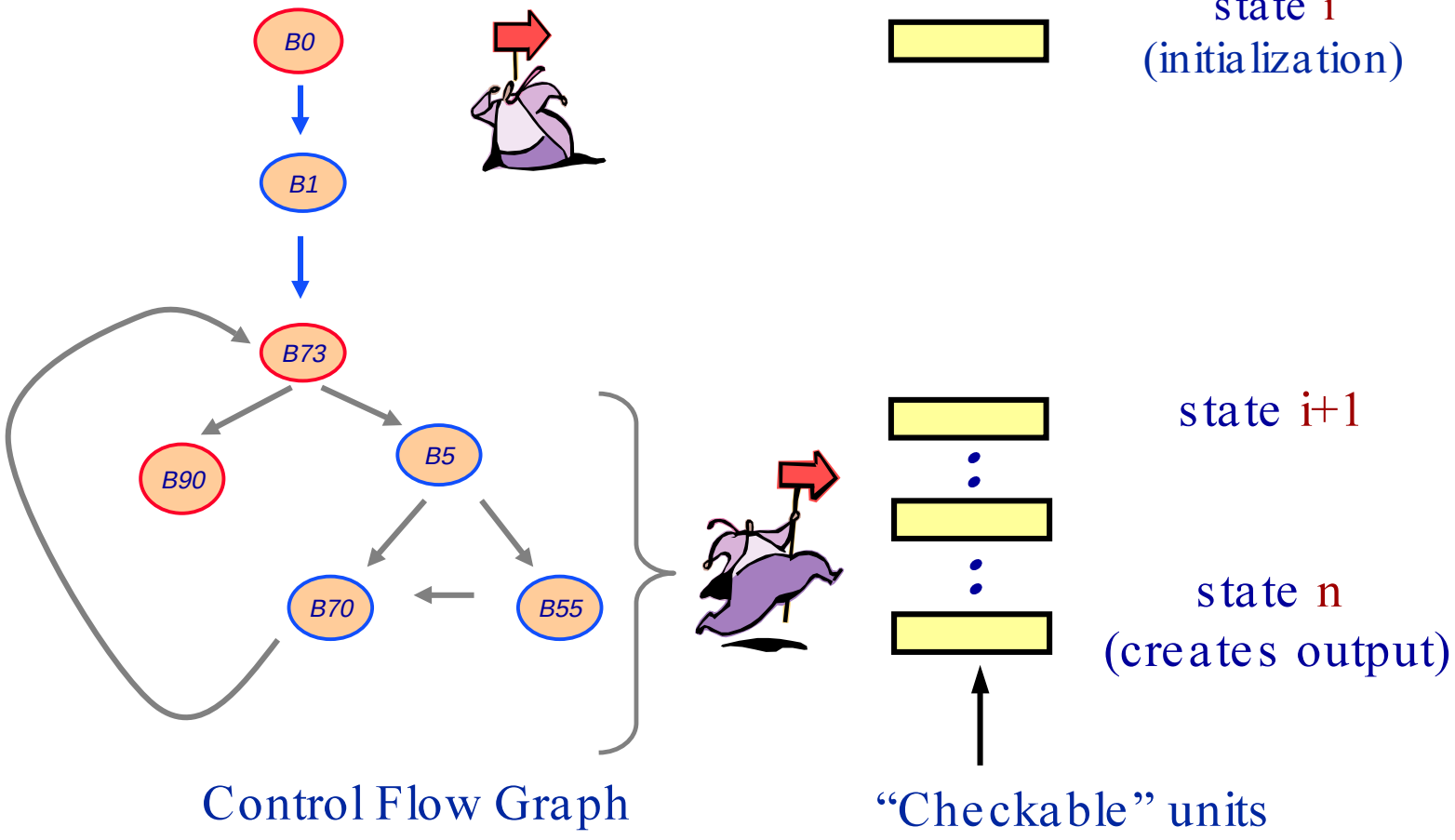


Execution
Traces



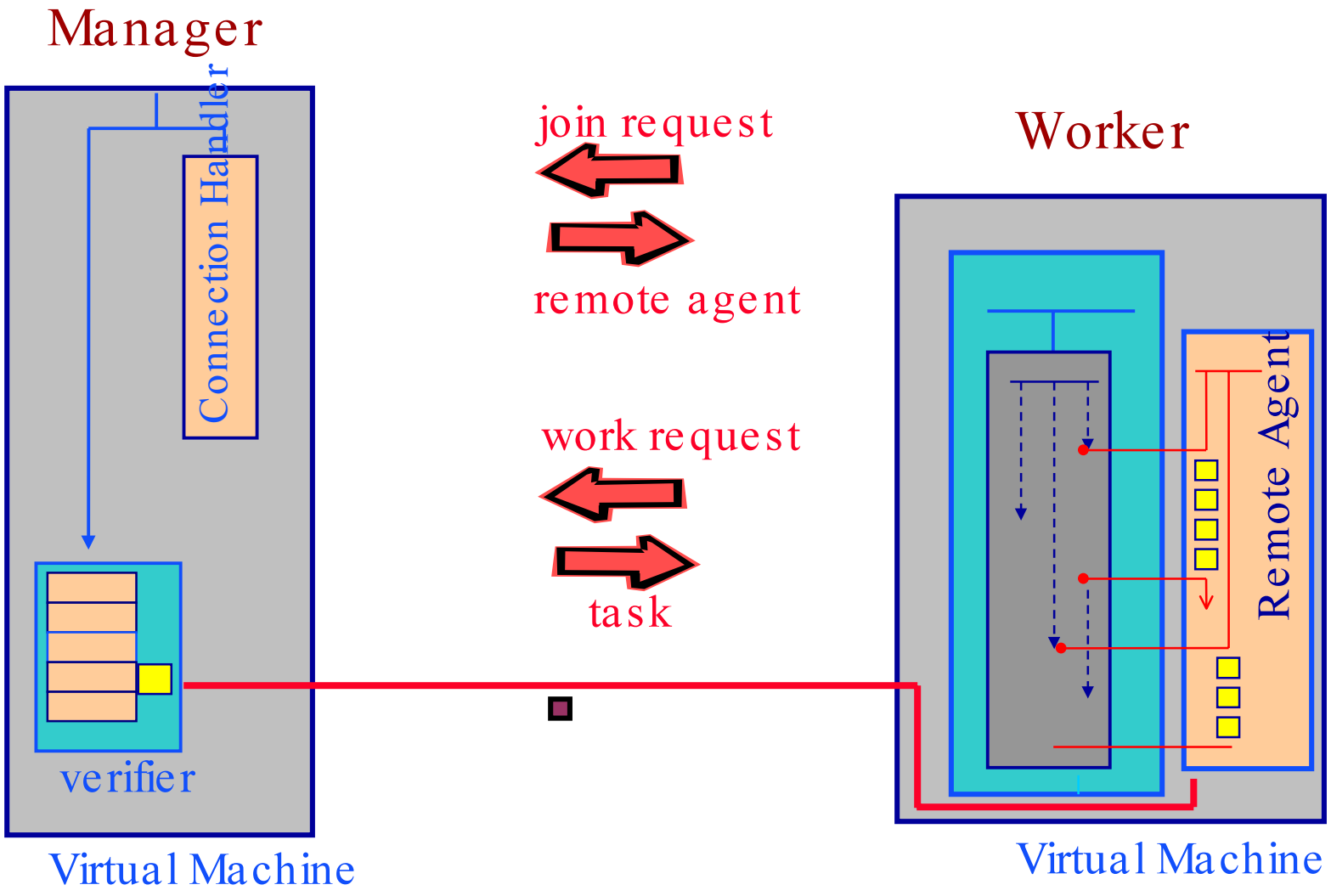
Manager/Verifier

Program Analysis + Code Transformation

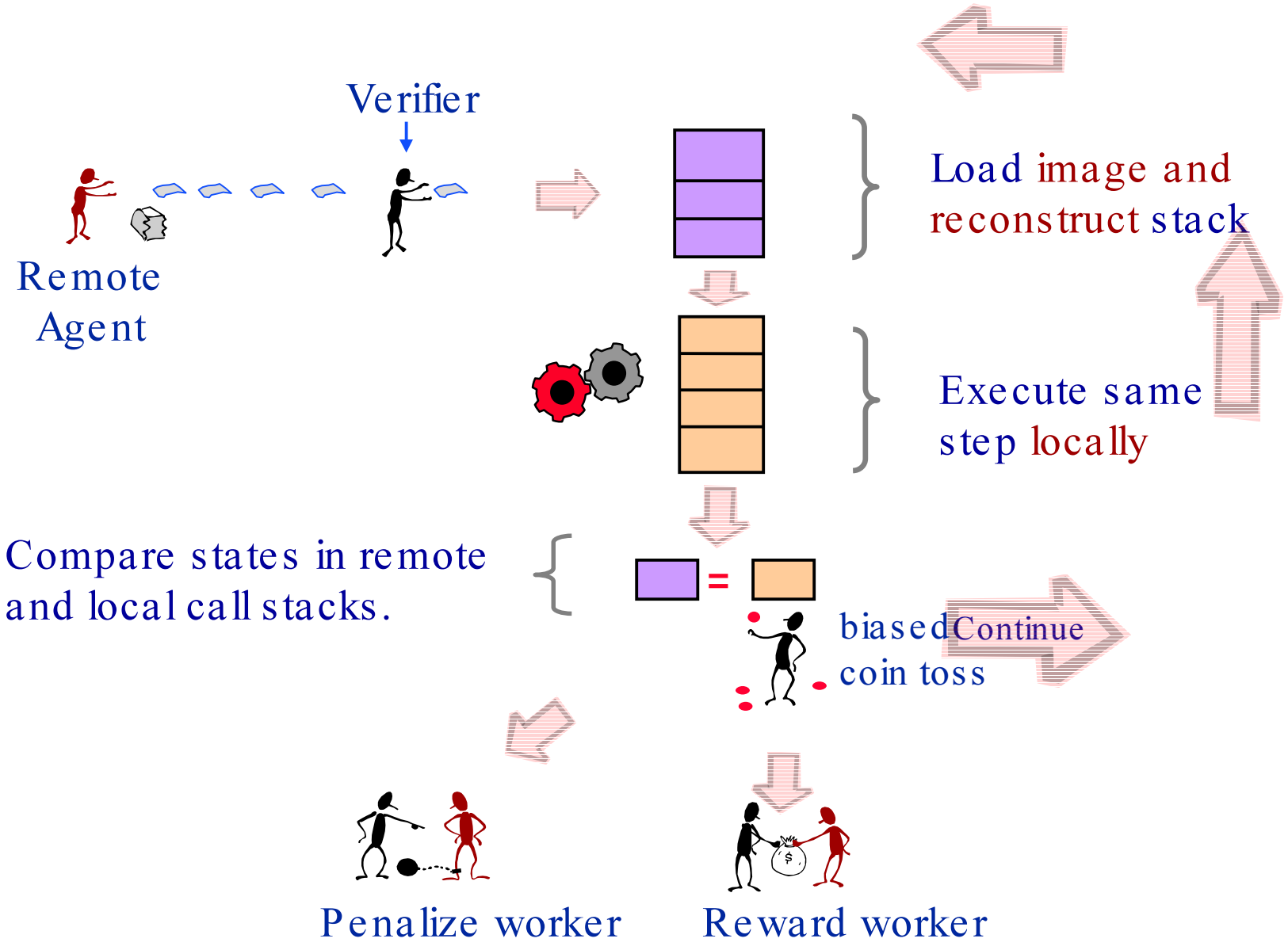


Key to instrumentation: (1) Execution of these units will be captured in traces
(2) Each trace will correspond to the output of exactly one unit.

Run-time system

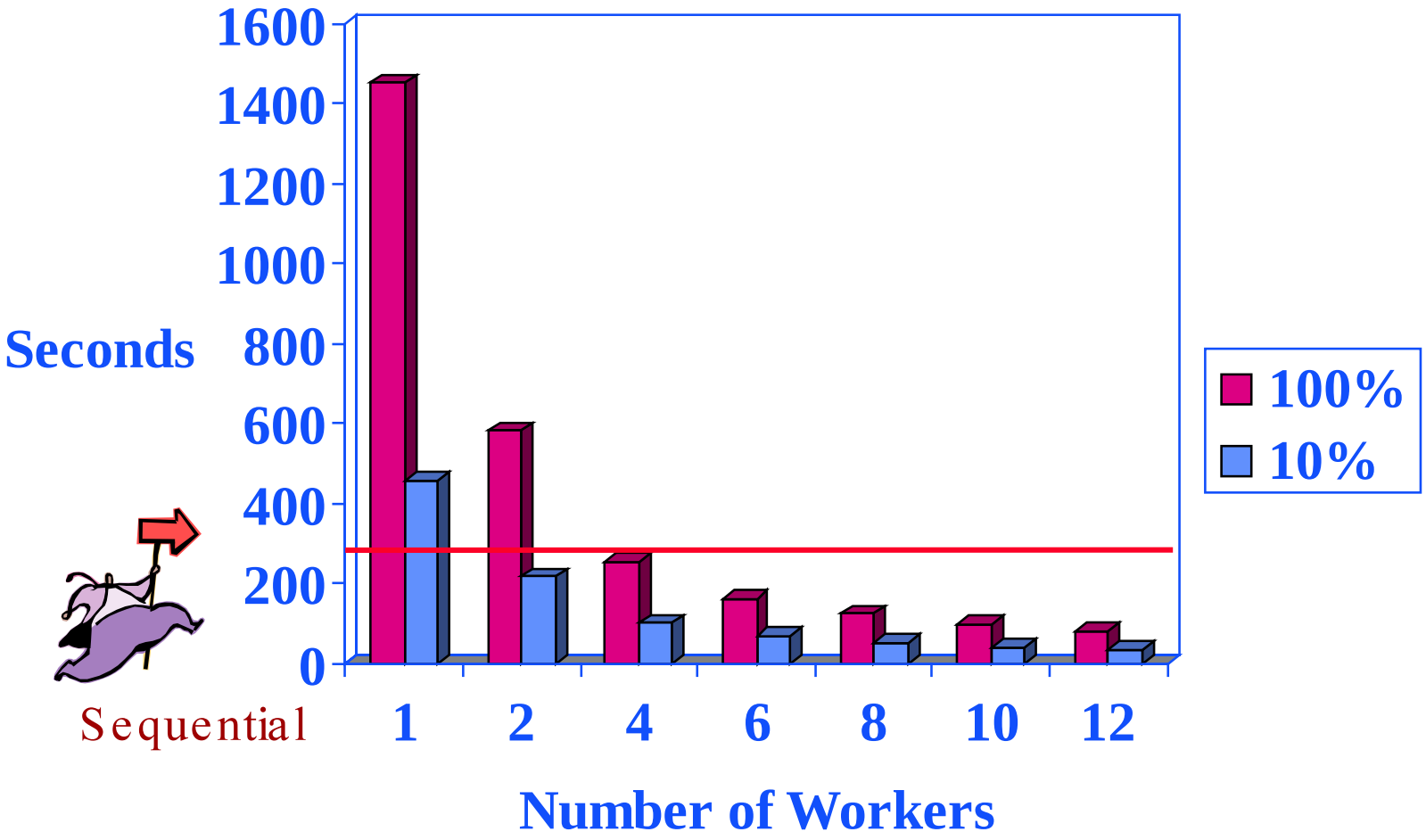


Verification



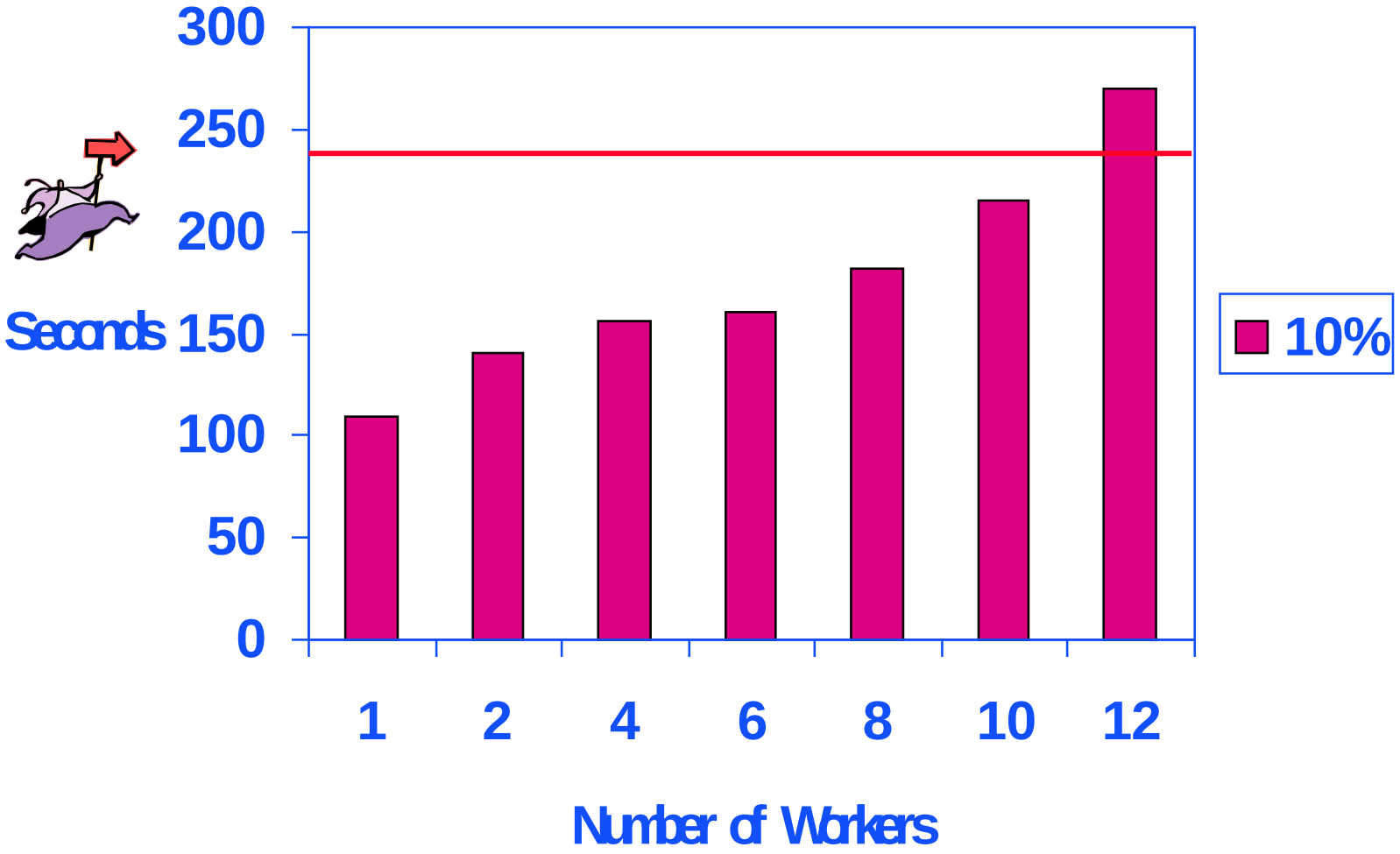
Discrete Log Performance Results

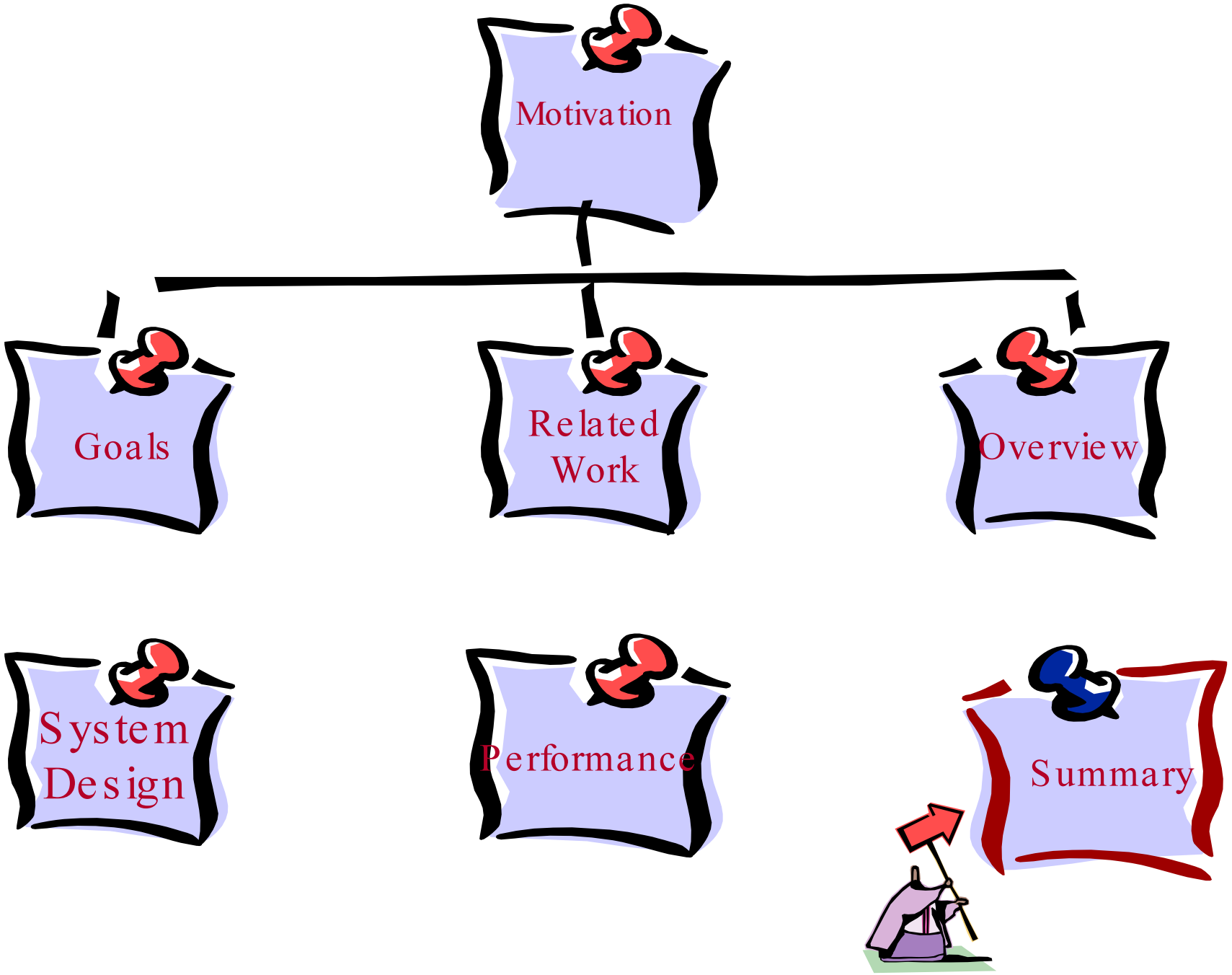
runtime system (i.e., remote monitoring)



Discrete Log Performance Results

(verification)




Motivation
Goals
Related Work
Overview
System Design
Performance
Summary

Limitations

- More than 35% of execution time is spent resuming suspended threads within the remote JVM --- however, since the Remote Debugging interface has been unsupported since JDK 1.02, performance enhancements are unlikely.

- Restricted to a set of programs that are “execution capturable” and have uniquely identifiable units.