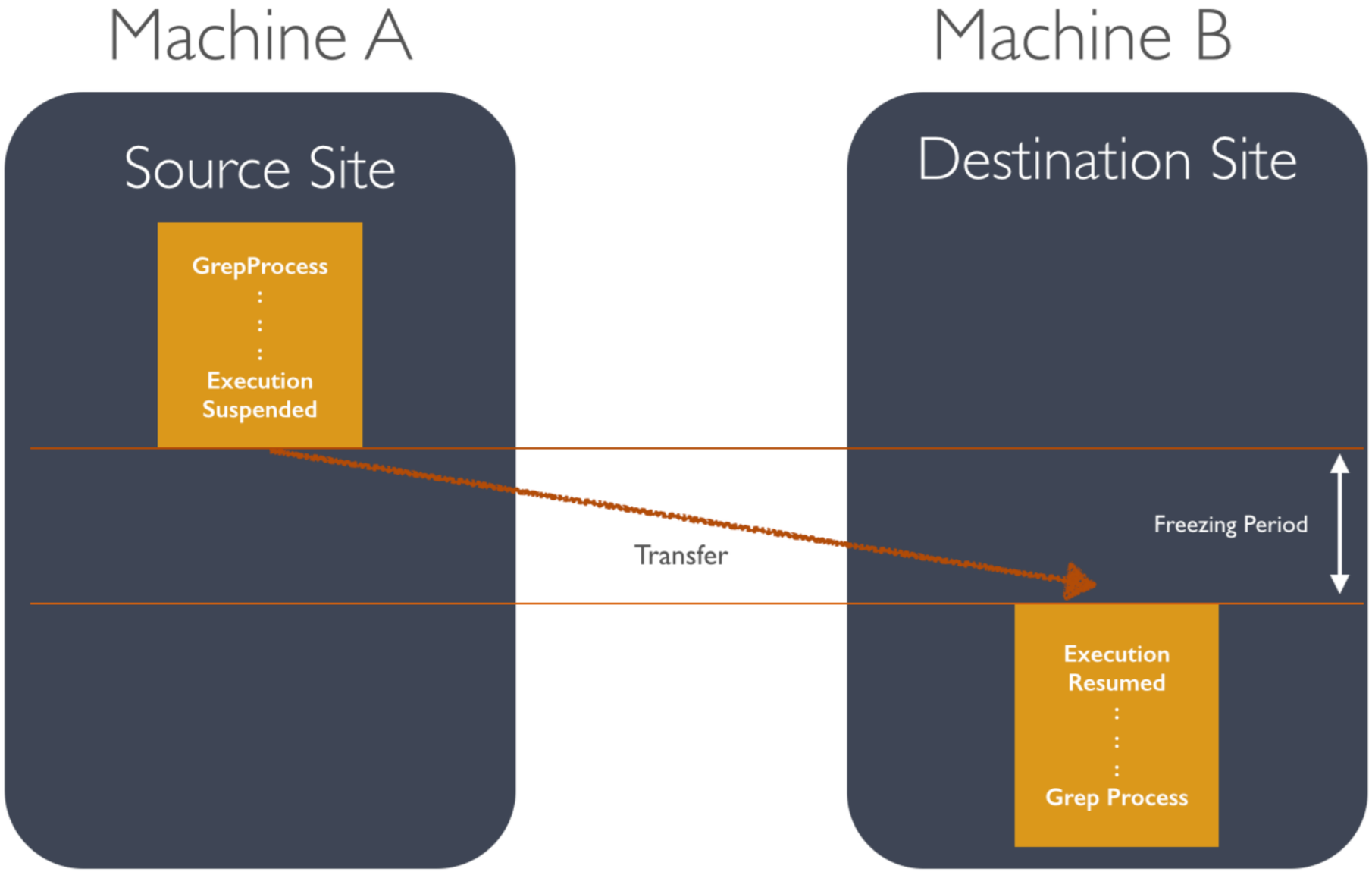

Concept borrowed from: Prof. Munehiro Fukuda (University of Washington)

15-440 / 640

Project 1 Recitation

GOALS

- Give you some food for thought
- Encourage good design and implementation
- Clarify some concepts and principles
- Answer some questions, Make life easier (or NOT)
- Sabotage: Your Mission of Trying to Land on Moon

OUTLINE

1. Requirements and Architecture Overview
2. AFS is your friend - Really.
3. Serialization
4. Transactional File I/O and the friendship with AFS
5. Just Enough: Network I/O [Send/Reply]
6. Reminder of what is NOT expected
7. Report Clarifications and Use of Third-Party Libraries
8. What is EXPECTED

REQUIREMENTS

- “Work” Migration (State Preservation)
- Work should be agnostic (Pick up right where you left off)
- Semantics of Migratable-Process and Transactional I/O
- System Orchestration - (Launch, Remove, Migrate)
- Cluster Reporting and Health Management
- Two Examples: Designed to demonstrate the nature of your Framework

ARCHITECTURAL OVERVIEW

ELEMENTS OF CONCERN

Communication	Information	Interaction (User Experience)	Processing	Failure Management
- Listeners on each machine - Send and Receive - Status/Health Monitoring - Concurrency	- "State" Preservation - Metadata Info - Semantics of Messages - Global & Local State	- User Input - Cluster Information - Cluster Status Report - Start/Stop	- Examples Programs - File Access - On-User-Request Demand Processing	- No special failure mgmt. - We only need Failure Reporting

http://daretodv8.wordpress.com/2014/08/26/distributed-systems-product-design-framework/

AFS - ANDREW FILE SYSTEM

- Your best-friend for Project 1. [Quick Background]
- What is a Distributed File System anyway ?
- Why should I care about it for Lab 1 ?
  - Ubiquitous Access to your files
  - No need to worry about file transfers (woo-hoo)
  - Mechanism to test your project
  - Just enough I/O that you need
- You will be submitting your code using AFS

JAVA SERIALIZATION

- What ? + Motivations for Serialization
- Examples (Discussed with some code snippets)
- AFS => Your Friend
- Personalize Advice - Taken time out to see the api descriptions and look at examples or tutorials. Guess what - they actually help!!

TRANSACTION FILE I/O

- Remember AFS your friend again - so your files are magically everywhere.

- What's "Transactional" about this ?
  [Remember where you left-off . Maintain State]

- Things to think about
  - Cache the file descriptors for performance
  - Separation of responsibilities, API Design

NETWORK: JUST ENOUGH

- Requests and Replies (Main Purpose)
  - Simple Semantics
  - Consistent Infrastructure
- No Need for Complex Network Connection Interactions and Network State saving for your examples
- Using Sockets to transfer your process information

WHAT IS **NOT** EXPECTED

- Load Balancer :(
  Don’t make it harder than it is.
  We will do that for you in the upcoming assignments :)

- Synchronization amongst Multiple Processes for File I/O

- Preserving Network State of Processes

- User Interaction + Health Monitoring Capabilities at Every Node

- Extra Cool “Stuff” will earn you:
  OUR RESPECT + BRAGGING RIGHTS.

REPORT CLARIFICATIONS & THIRD PARTY LIBS

- Report is a VERY IMPORTANT component in your grade.

- Use of Third-Party Libraries is OK as long as:
  - You package it properly
  - Runs on GHC machines
  - We do not have to install something extra
  - Doesn’t make the problem trivial

WHAT IS EXPECTED

- Detailed Instructions about Deployment
- Design Decisions and Explanations in Report
  - (not equal to description of code)
- Well commented code that helps us walk-through your code

Q & A