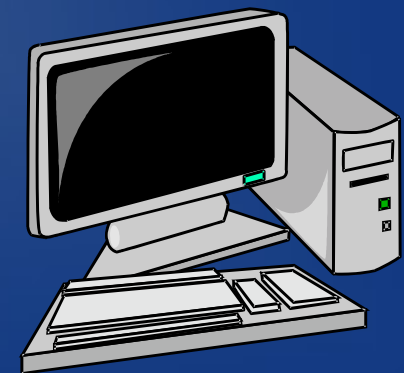
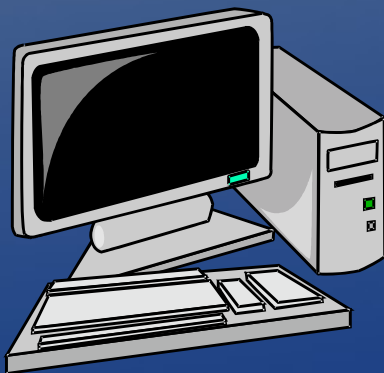
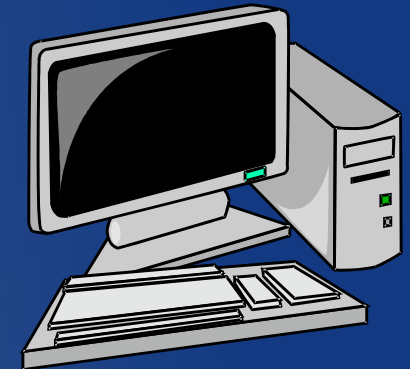
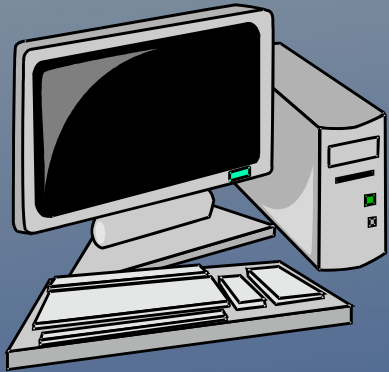




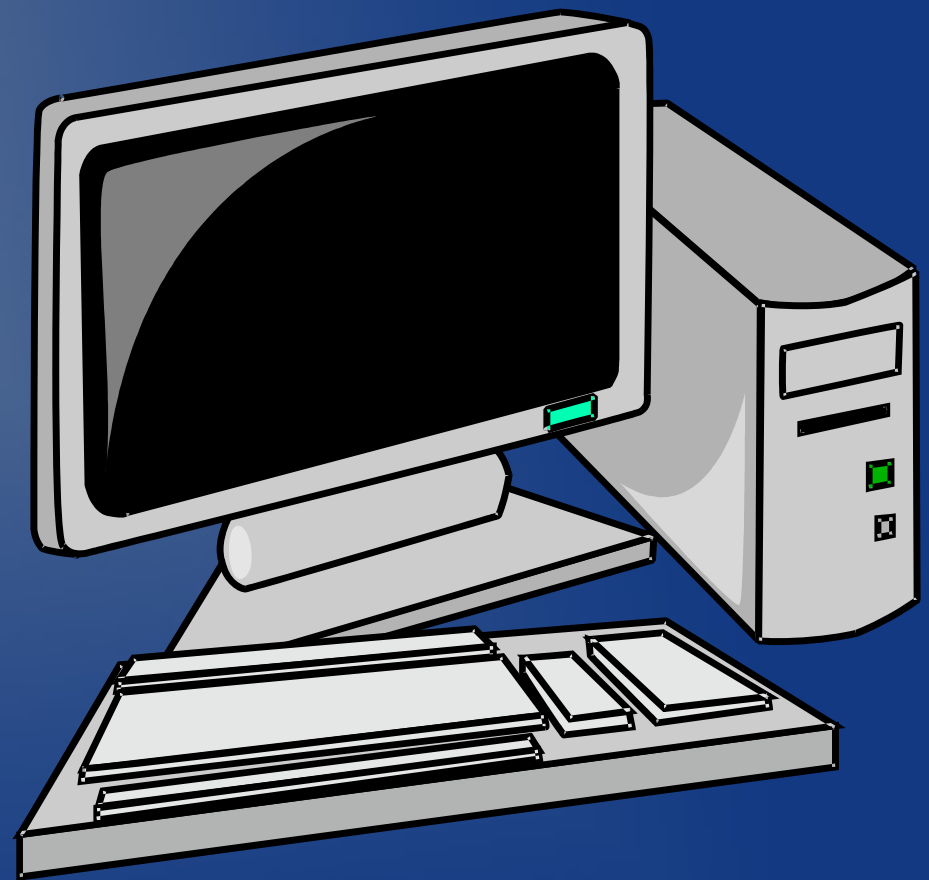
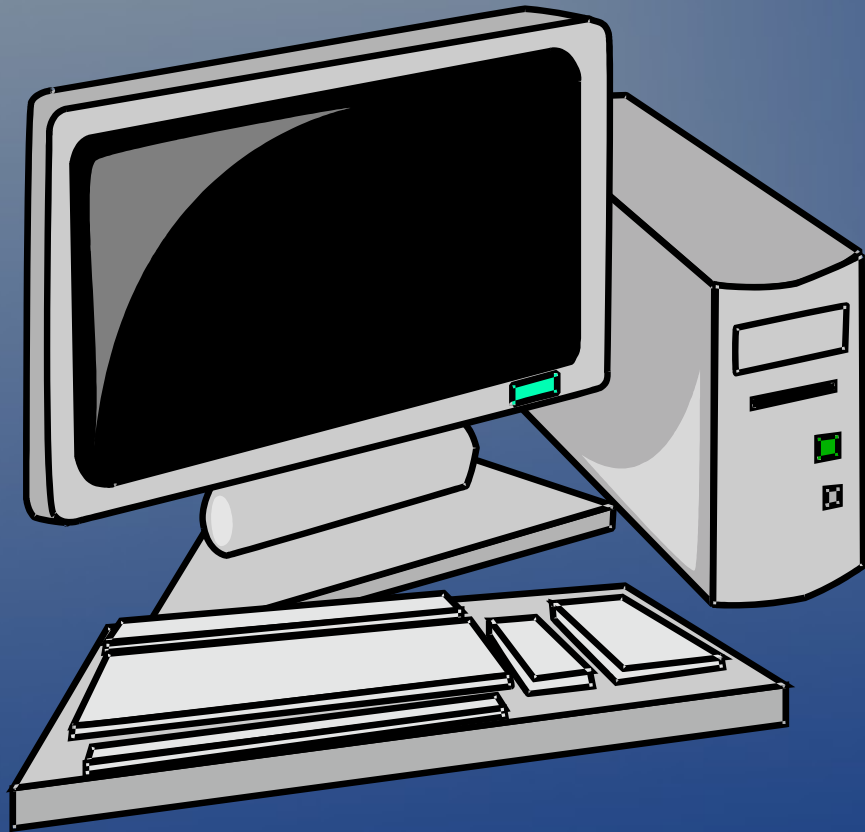
A bit about clouds

Jeff Epstein, on the occasion of the Fitzwilliam Graduate Student Conference, March 19 2011

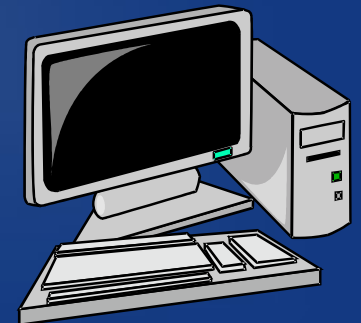
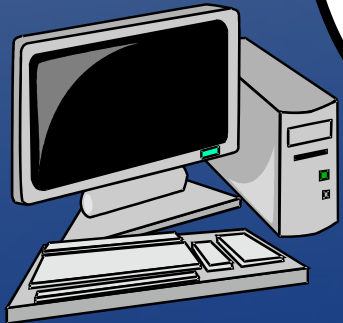
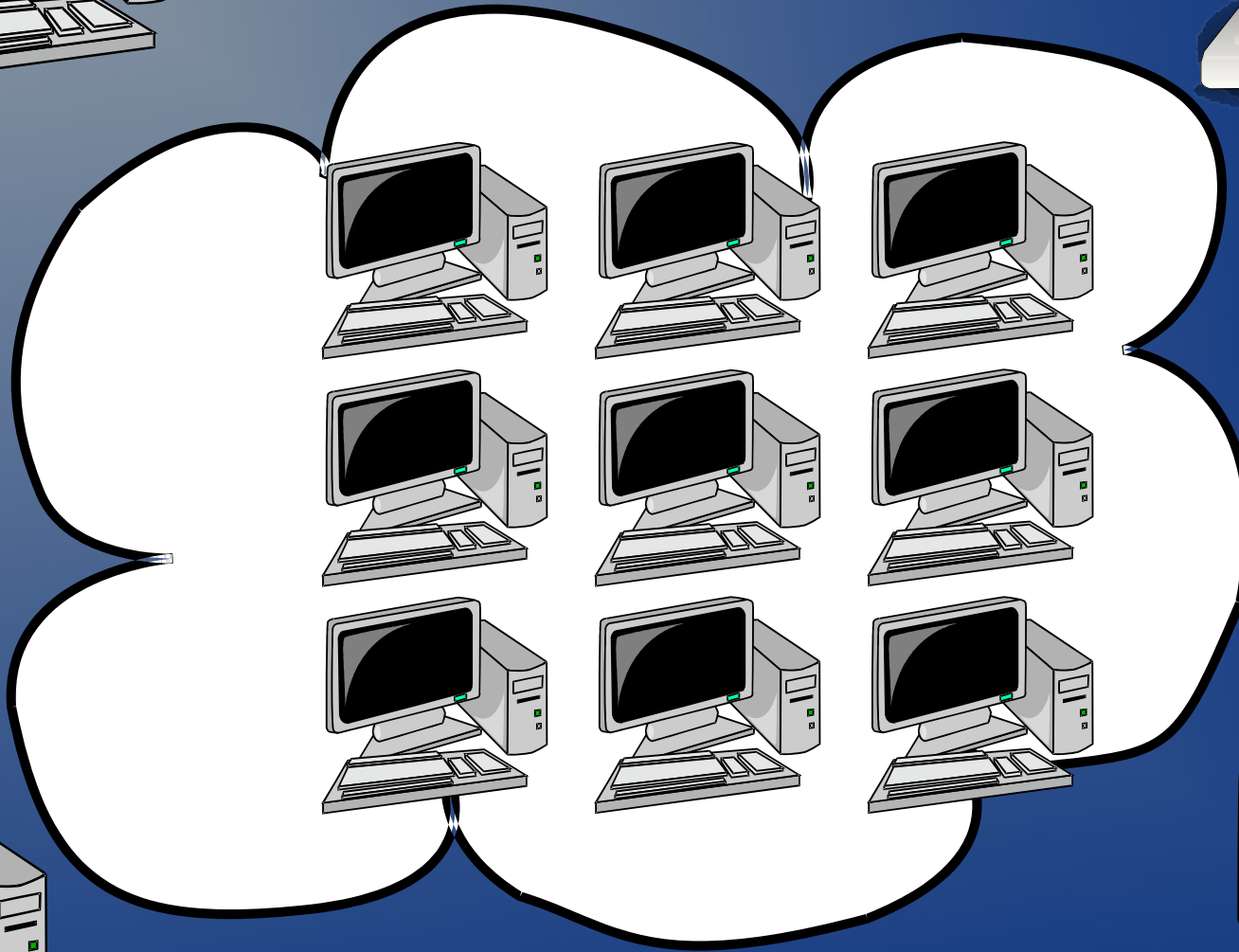
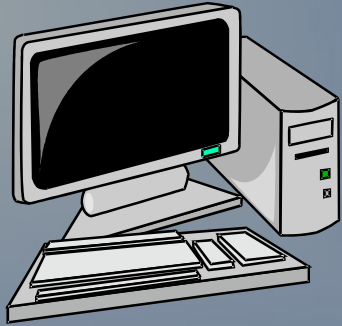
The multiuser model



The personal computer



The cloud



What is the cloud?

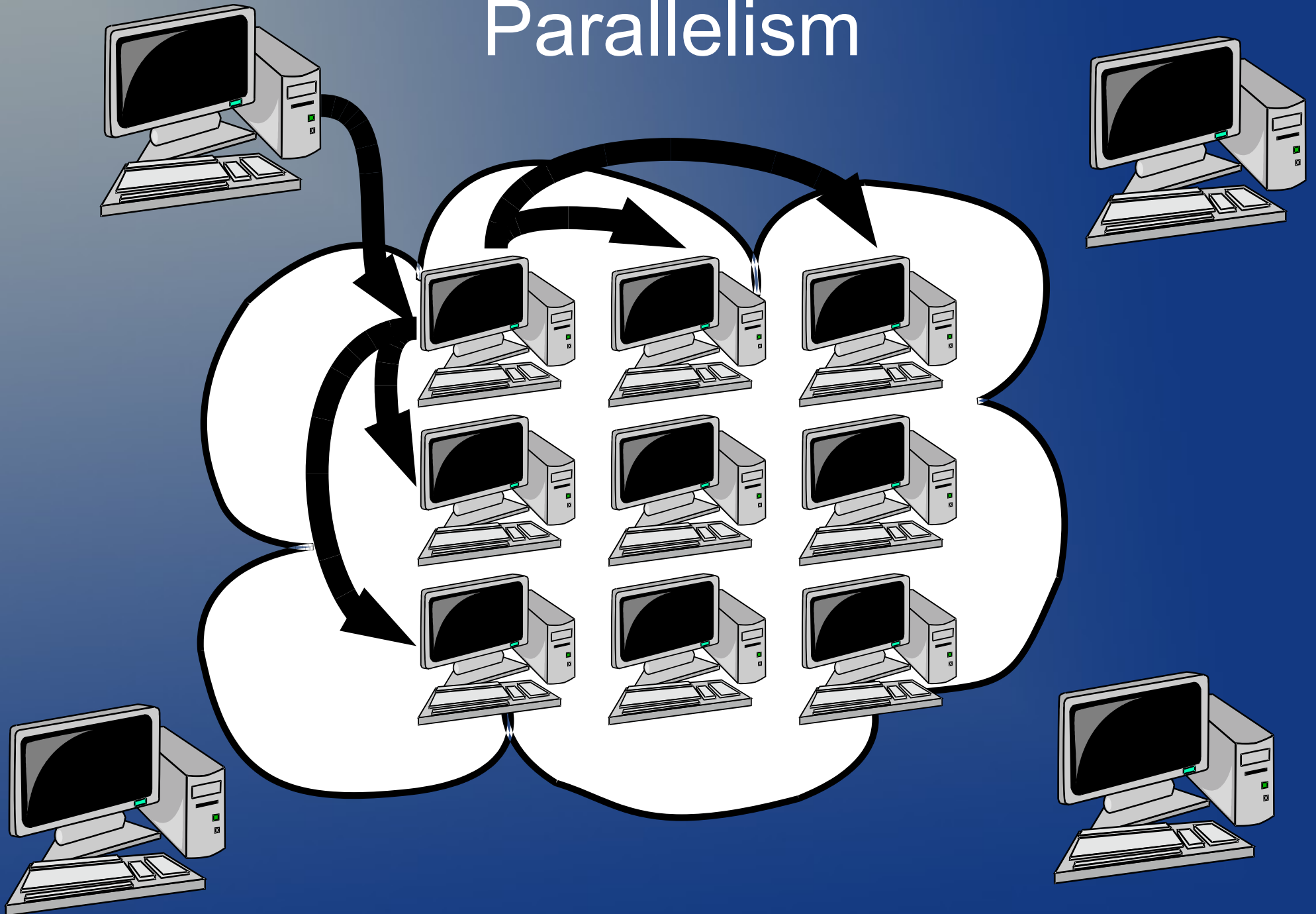
The National Institute of Standards and Technology says:

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

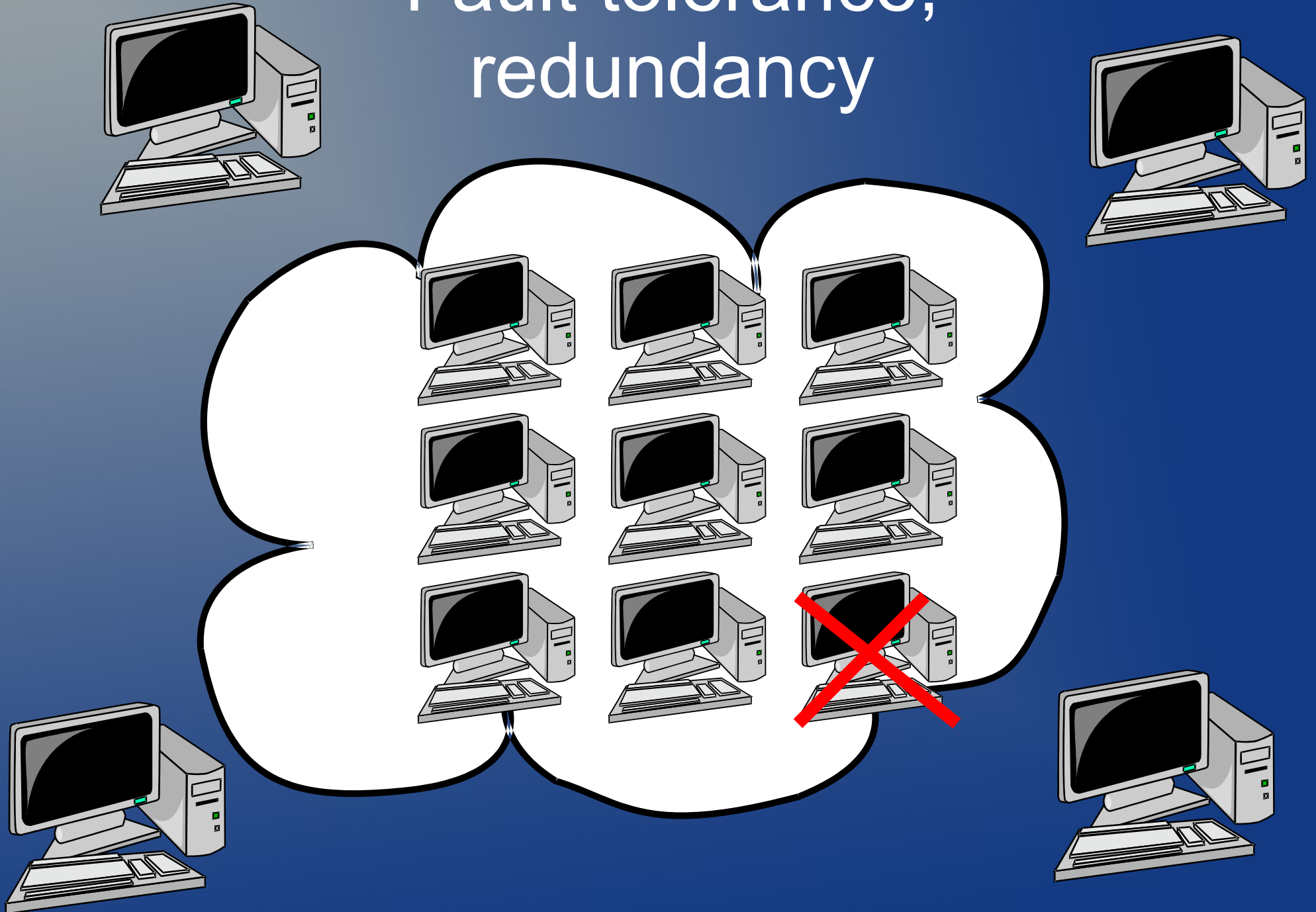
What is the cloud?

Software as a service, rather than a product.

Parallelism



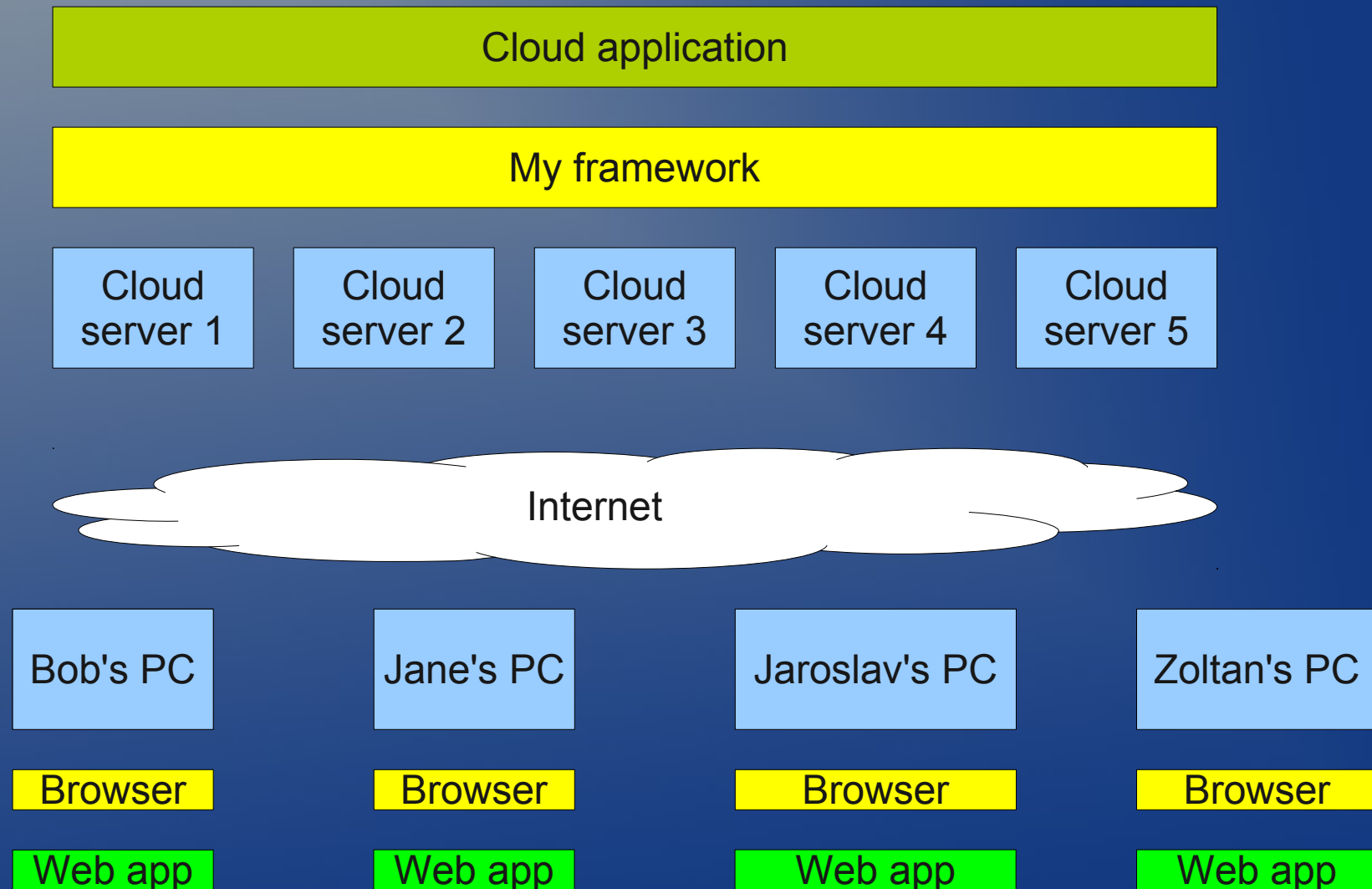
Fault tolerance, redundancy



What I do

- I'm creating a "framework": a tool for software developers
- Its goal is to let programmers develop software for the cloud quickly and easily
- This is a hot topic in industry, and also has some interesting technical challenges

My framework

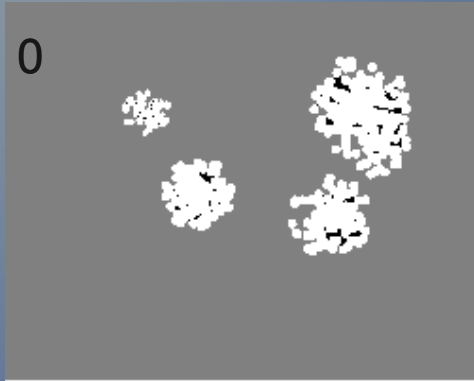


A real example: K-means

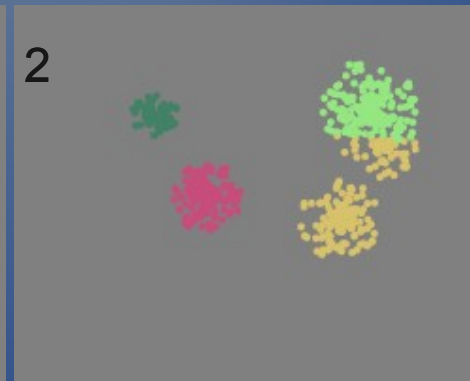
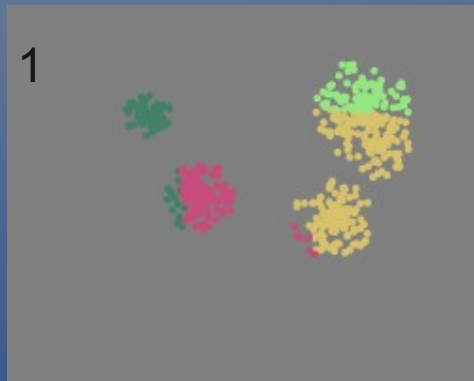
- Statistical algorithm for locating natural clusters in data points
 - Input: data points
 - Output: assignment of points to clusters
- Highly parallel: each node in the cloud computes part of the algorithm simultaneously; their results are then combined
- Coordinating all the nodes is complicated, especially in case of failure

K-means example

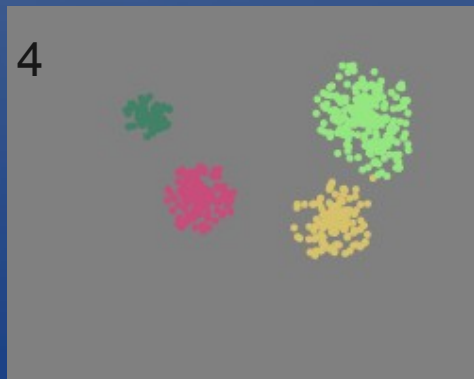
Initial input:



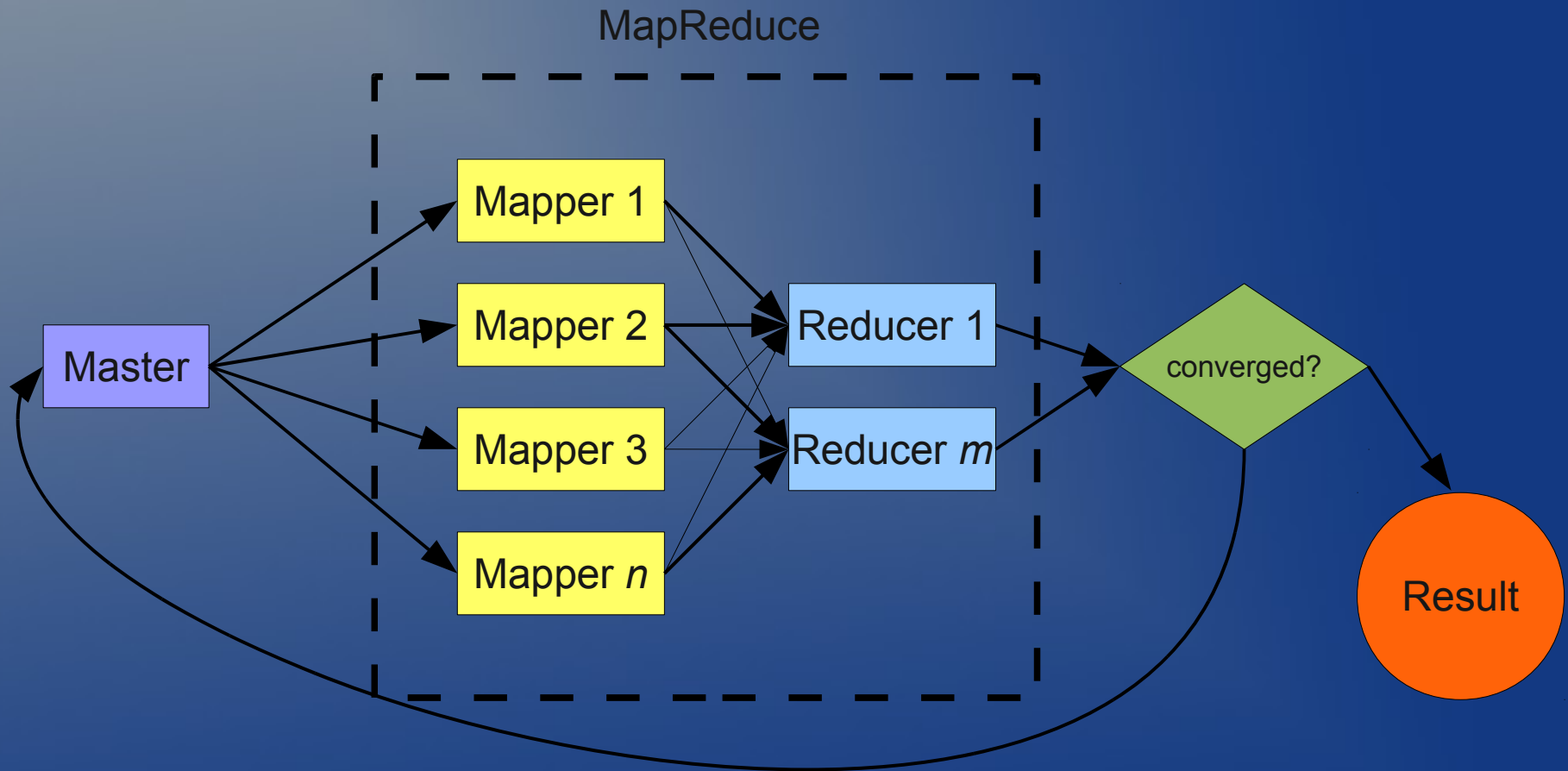
Intermediary results:



Final result:



K-means in overview



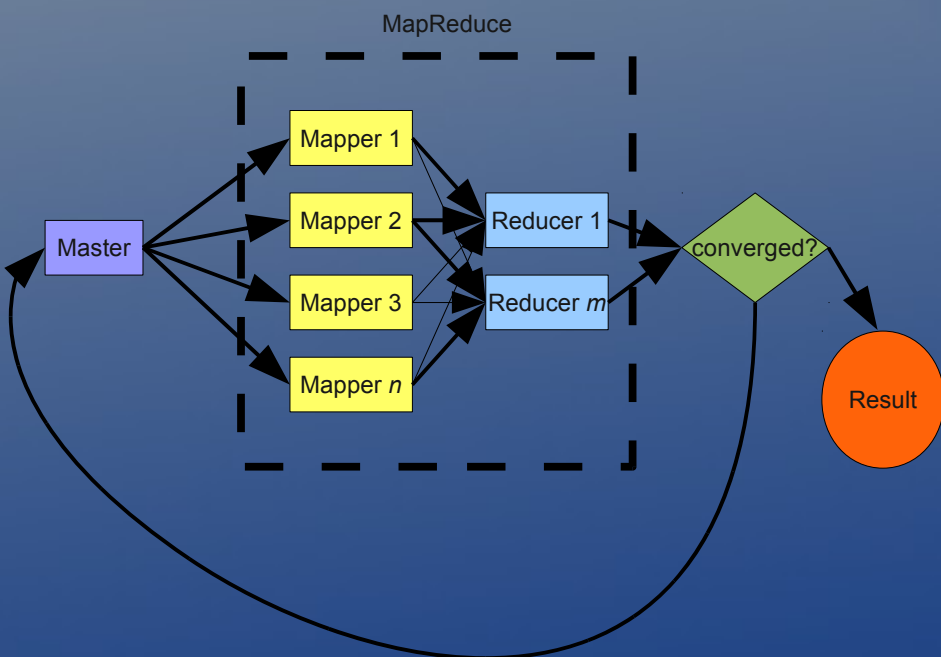
How does my framework make implementing k-means easier?

- We introduce the idea of a "task"
 - a restartable, locality-neutral block of code
 - Failure transparency: programmer doesn't care if computer explodes, just give me results!
- Communication framework for sending data between tasks
- Resolves data dependencies with the notion of a "promise"
 - represents a value that has yet to be computed

Framework abstraction

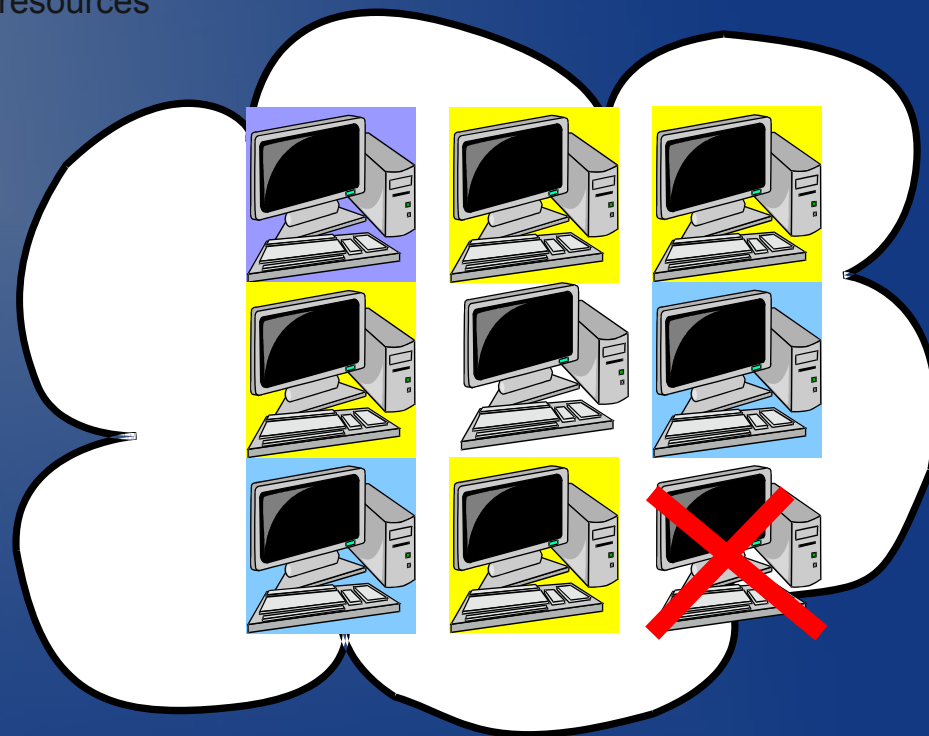
What the application programmer sees

Tasks with data dependencies



The framework translates that into

An allocation of tasks to currently available hardware resources



Okay, so what?

- Cloud application and web services will be more reliable, more available
- Programmers will be less stressed, more social
- The PC revolutions ends and we finish where we started: centralized computing services