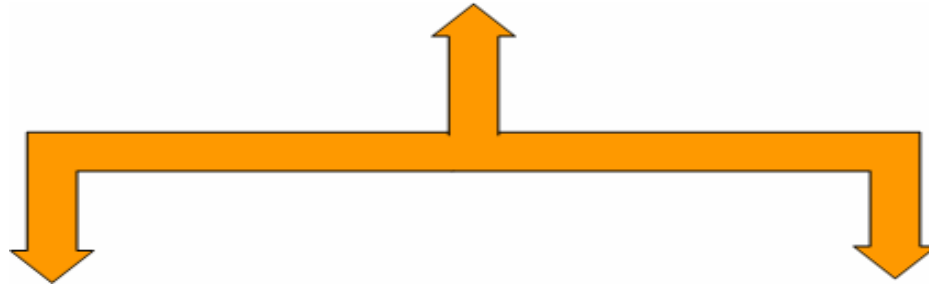


WP6 - Grid Economics and Business Models Project

**Project Investigators
Danny OH, Steven MILLER, HU Nan
16 May 2006**

The AE@SG Business Model Project



Experimental Perspectives

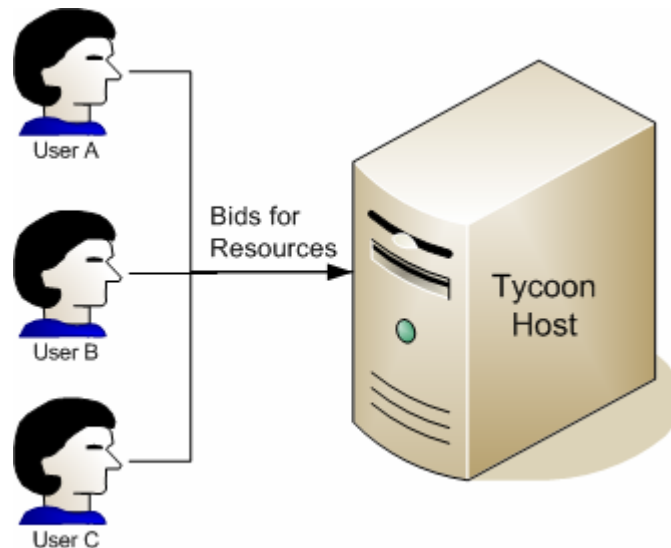
- Test how Tycoon system performs under different load and bidding patterns
- Explore how to leverage Tycoon as a grid resource allocation framework for a Grid Market Place in Singapore (e.g. the Grid Digital Media Portal)

Empirical Perspectives

- Analyze actual data on grid usage
 - Demand patterns
 - User behaviours
- Use empirical analysis in testing of Tycoon

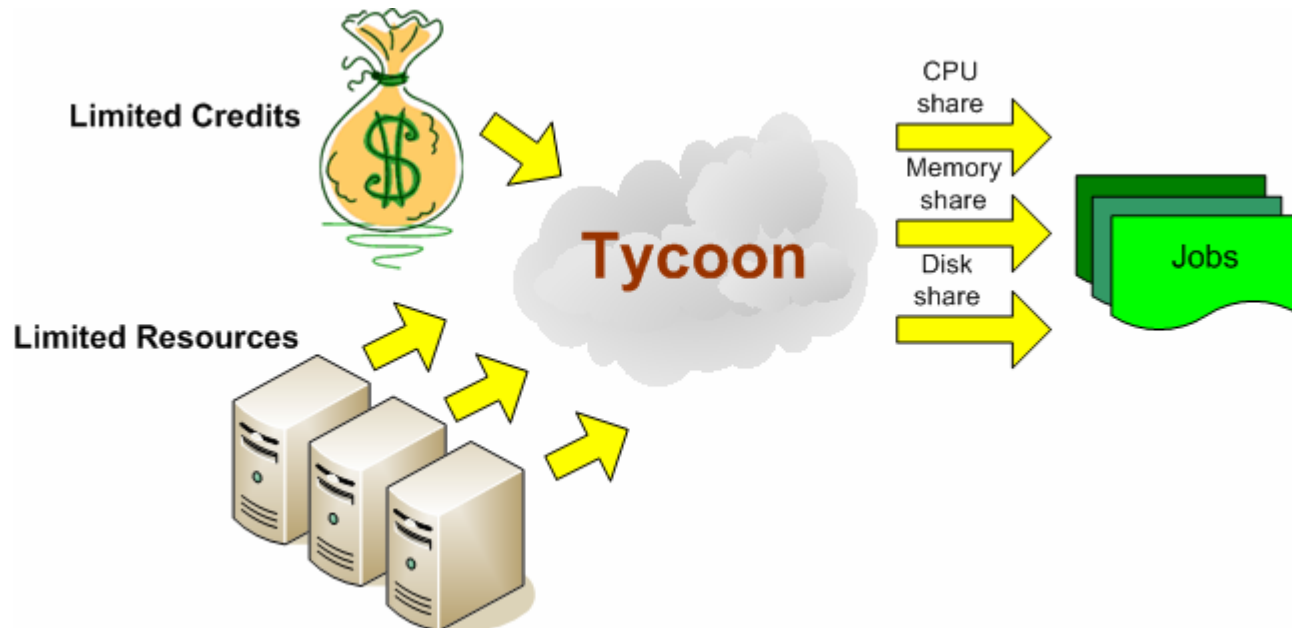
The Challenge Motivating Tycoon

- How to automatically and fairly allocate shared resources in a grid setting at low cost?
- Tycoon's answer:
 - Resources are allocated via continuous bids which last throughout the job.
 - Users specify value of job through bidding.

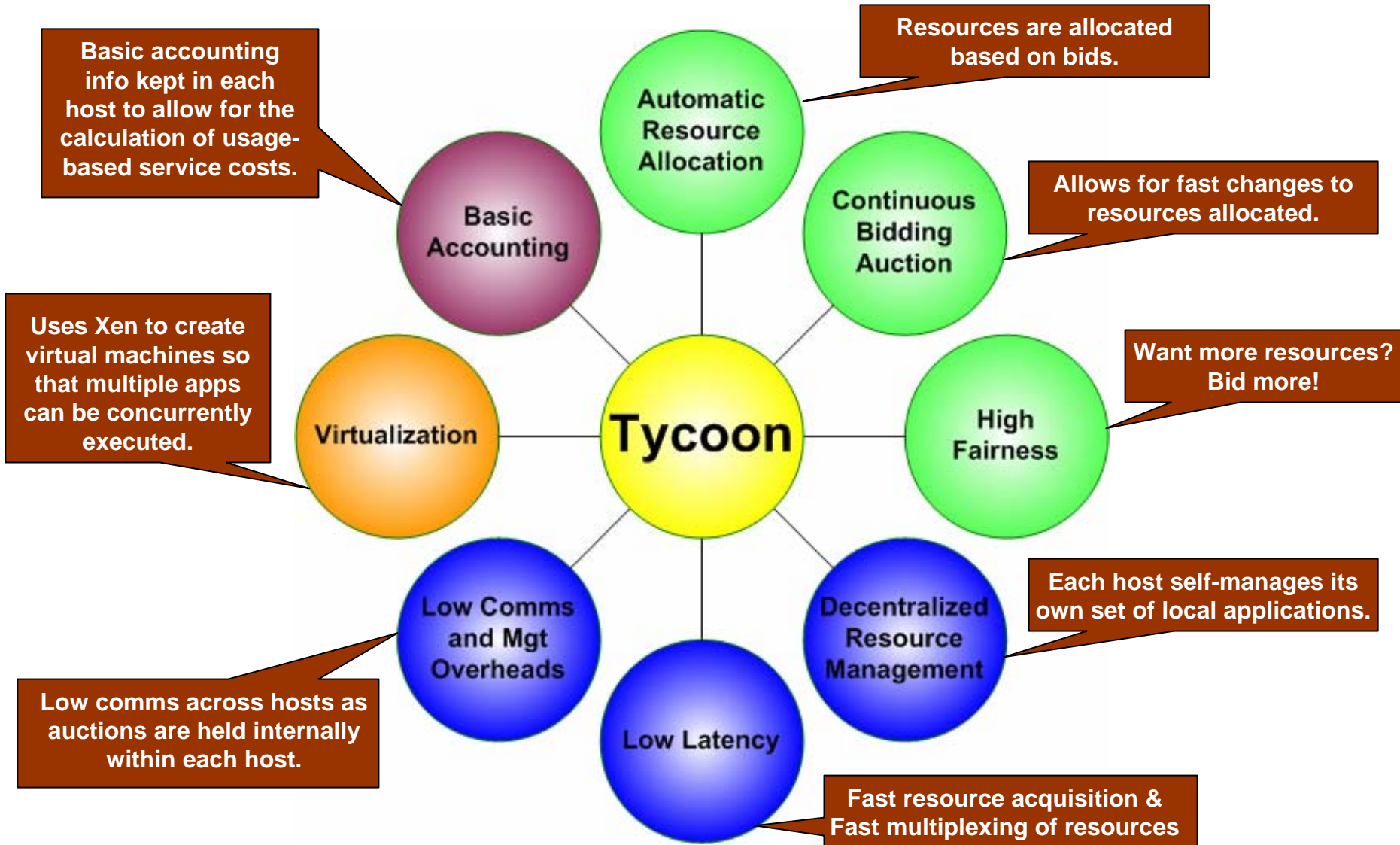


The Tycoon System for Grid Resource Allocation

- Tycoon is a grid resource allocation system which allows users to bid and pay for shared resources using credits.
- Tycoon was developed by the Information Dynamics Group at HP Labs, Palo Alto (Dr. Kevin LAI, principal investigator).
- Key assumptions:



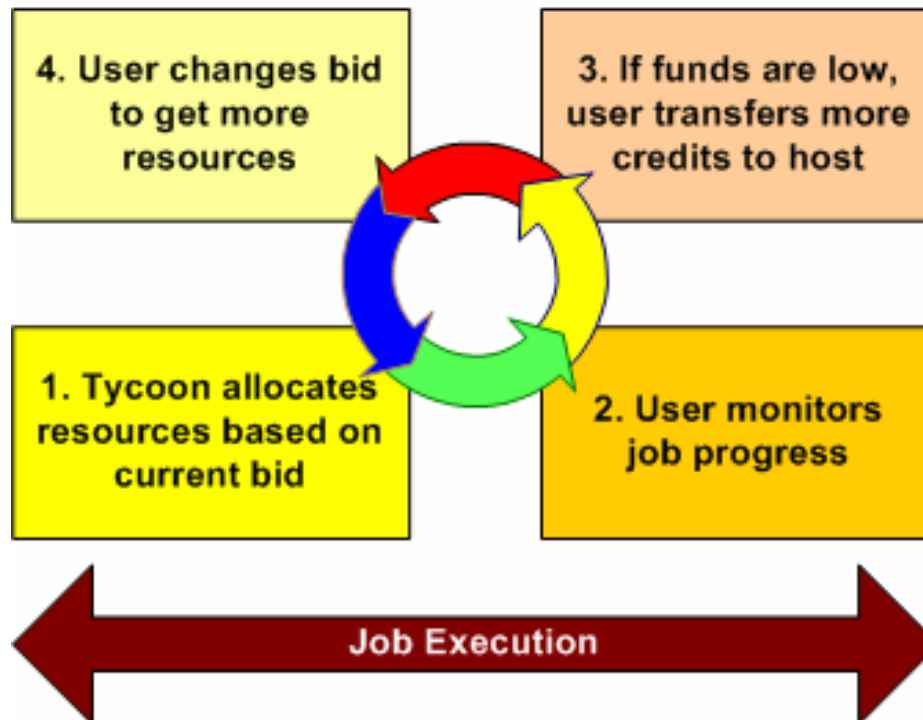
Overview of Tycoon



Key Properties of Tycoon

- Automatic Resource Allocation
 - Based on the total bids for all jobs, Tycoon allocates at regular intervals:
 - the share of CPU, memory and disk for each job

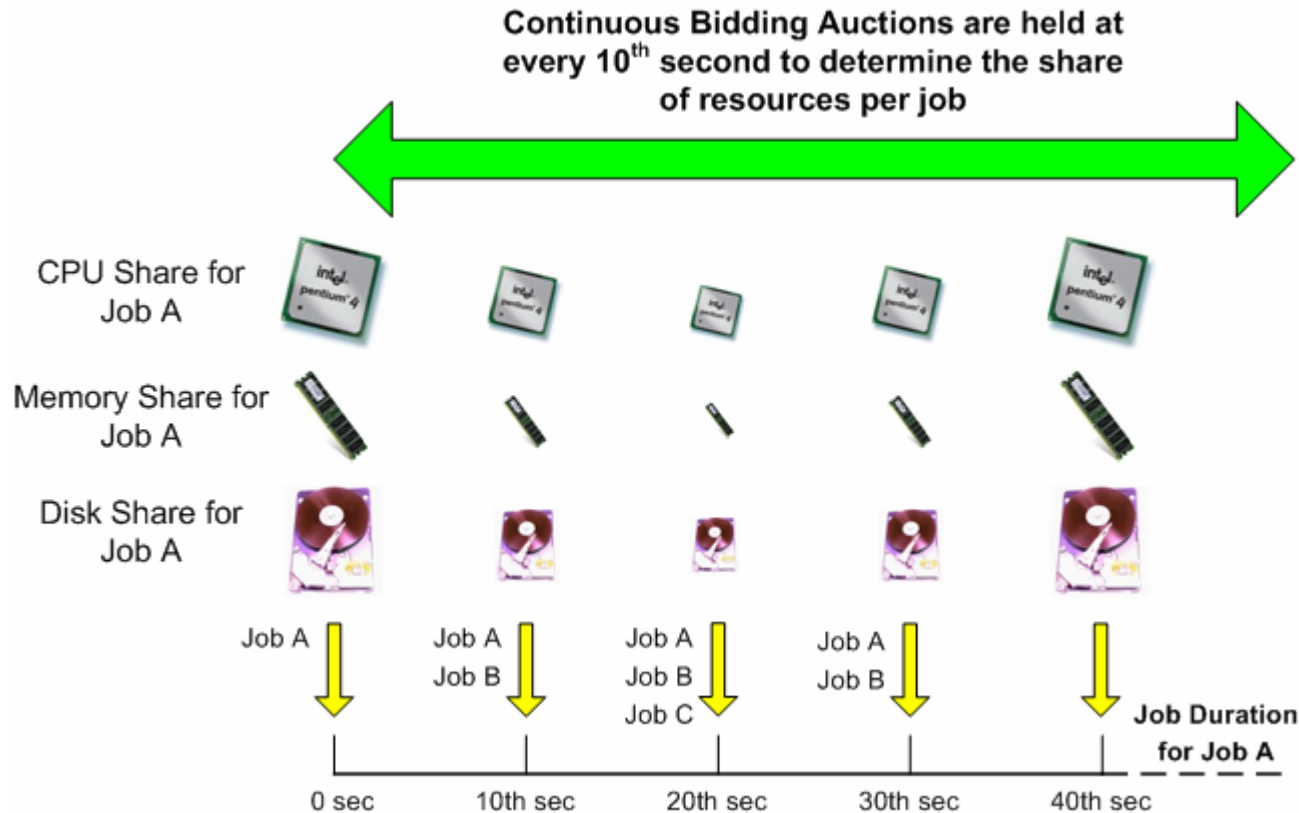
While a particular job is running



Key Properties of Tycoon

- Continuous Bidding Auction

- User can choose to submit 1 constant bid or change the bid amount anytime during the job execution period to get more or less resources.



Key Properties of Tycoon

- High Fairness

- Tycoon has been proven in simulations to exhibit high fairness.
- All users who use their entire allocation pay the same per unit of the resource.

Assuming 1 user submitted the following bid for his/her job,

User Name	Job Name	Bid Price (\$)	Bid Expiration Time (s)	Bid Price Per Second (\$/s)
User A	Job A	3	180	$3/180=1/60$

At the start of the first 10s period,

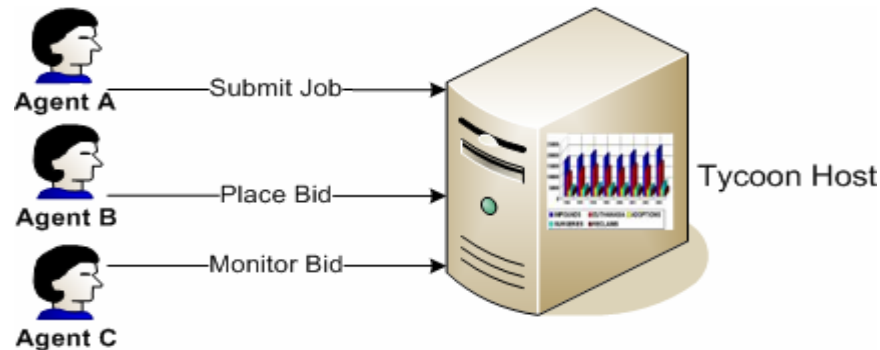
Job Name	Bid Price Per Second (\$/s)	Bid Price for CPU Per Second (\$/s)	Bid Price for Memory Per Second (\$/s)	Bid Price for Disk Per Second (\$/s)
Job A	1/60	$(1/60)/3 = 1/180$	$(1/60)/3 = 1/180$	$(1/60)/3 = 1/180$

Key Properties of Tycoon

- To date, Tycoon only operates at the level of a single server (1 server = 1 node).
- Tycoon can be installed on multiple nodes, but each node is independent of the other.
- Bids on one node are unrelated to bids on another node.

The Experimental Work with the Tycoon System for Grid Resource Allocation

- Investigate the economic performance of the Tycoon resource allocation method.
 - Run simulations of different load and bidding patterns.
 - Identify load and bidding patterns relevant to the digital media market scenario.



- Based on the simulation results,
 - Recommend improvements to Tycoon in the areas of resource allocation and bidding mechanisms.
 - Provide design recommendations for the resource allocation framework for a Grid Market Place in Singapore.

Simulation Design

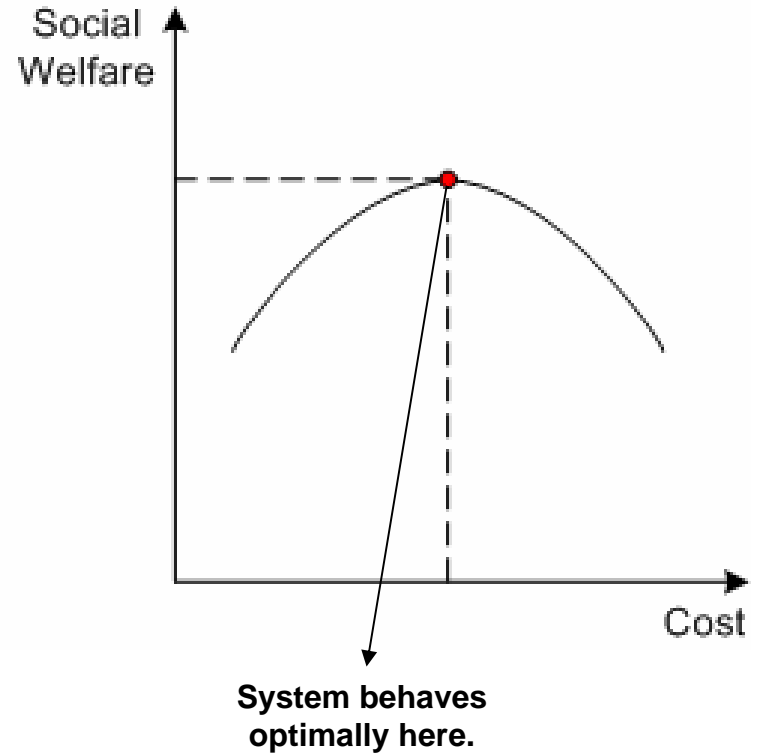
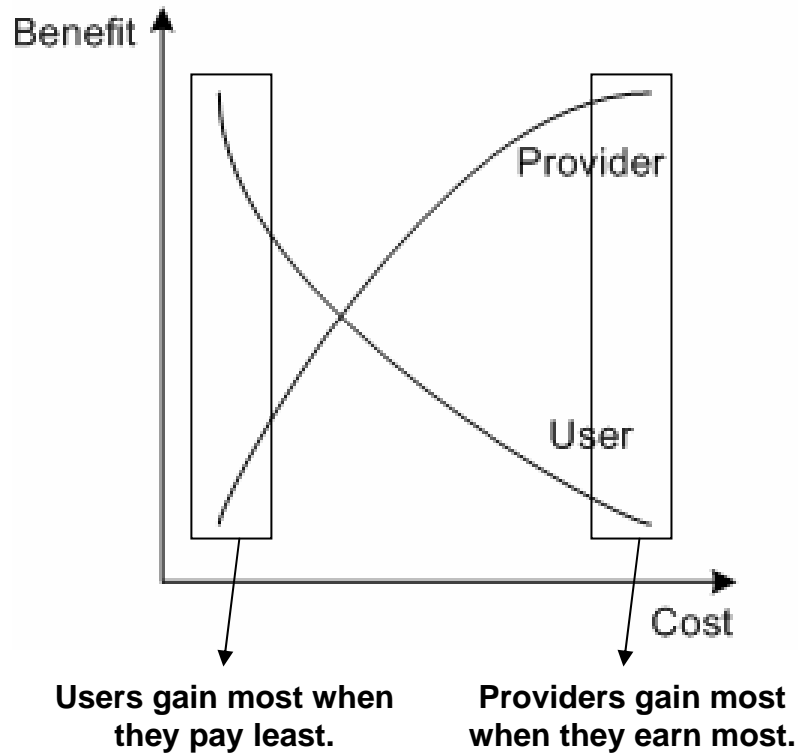
- Run different loads on non-Tycoon host to determine
 - Total amount of credits consumed per job
 - To derive this, we will run simulations with equal bid per job on Tycoon.
 - Total time taken per job
- Run different loads on Tycoon host to determine:
 - Total amount of credits consumed per job
 - Total time taken per job

Analysis of Simulation

- Based on the measurements taken, determine:
 - User Benefit = $S_b + S_t$
 - S_b = Initial Bid – Actual Credits Used
 - S_t = (Initial Bid / Initial Bid Duration) * (Initial Bid Duration - Actual Job Completion Time)
 - Provider Benefit = C_b
 - C_b = actual credits used
 - Assumed that no operational costs involved
 - Social Welfare = User Benefit + Provider Benefit

Tycoon Host		Non-Tycoon Host	
Initial Bid	\$100	Initial Bid	\$100
Initial Bid Duration (s)	1000	Initial Bid Duration (s)	1000
Actual Credits Used	\$50	Actual Credits Used	\$60
Job Completion Time (s)	500	Job Completion Time (s)	600
User Benefits	$(\$100 - \$50) + (\$100 / 1000) * (1000 - 500) = \100	User Benefits	$(\$100 - \$60) + (\$100 / 1000) * (1000 - 600) = \80
Provider Benefits	\$50	Provider Benefits	\$60
Social Welfare	$\$100 + \$50 = \$150$	Social Welfare	$\$80 + \$60 = \$140$

Analysis of Simulation



Simulation Input Variables

- Demand Characteristics
 - Regular, periodic demand : Jobs arrive at regular interval
 - Irregular, bursty demand : Random bursts of jobs which can result in high peaks with different arrival times.
- Job Size (Resource Requirements)
 - High : 500-frame job
 - Moderate : 300-frame job
 - Low : 100-frame job
- Willingness to bid more
 - Agents which make an initial bid and do not change bid.
 - Agents which make an initial bid, and modify bid during execution if resources allocated have dropped by certain pre-defined percentage.

Important Simulation Assumptions Regarding Agent Bidding Behaviour

- Agents are rational.
 - Agents will try to spend the least to run their jobs.
 - Therefore, agents will not bid their entire budget on the first bid in order to get the job done as quickly as possible.
- Each agent submits the same type of job.
 - The size of the job and the number of concurrent jobs are the only variables affecting load and total processing times.

Simulation Loads

Pattern 1: Equal Load

Rendering jobs of same duration submitted at start of simulation.

Time (min)	5	10	15	20	25	30	35	40	45	50	55	60
Agent A (300-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent B (300-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent C (300-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent D (300-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent E (300-frame job)	■	■	■	■	■	■	■	■	■	■	■	■

Pattern 2: Bursty Load

Long rendering jobs submitted at the start of the simulation run.

Short rendering jobs submitted at pre-determined intervals.

Time (min)	5	10	15	20	25	30	35	40	45	50	55	60
Agent A (100-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent B (100-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent C (500-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent D (500-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent E (500-frame job)	■	■	■	■	■	■	■	■	■	■	■	■

Pattern 3: Increasing Load

Rendering jobs of same duration submitted at regular intervals.

Time (min)	5	10	15	20	25	30	35	40	45	50	55	60
Agent A (300-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent B (300-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent C (300-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent D (300-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent E (300-frame job)	■	■	■	■	■	■	■	■	■	■	■	■

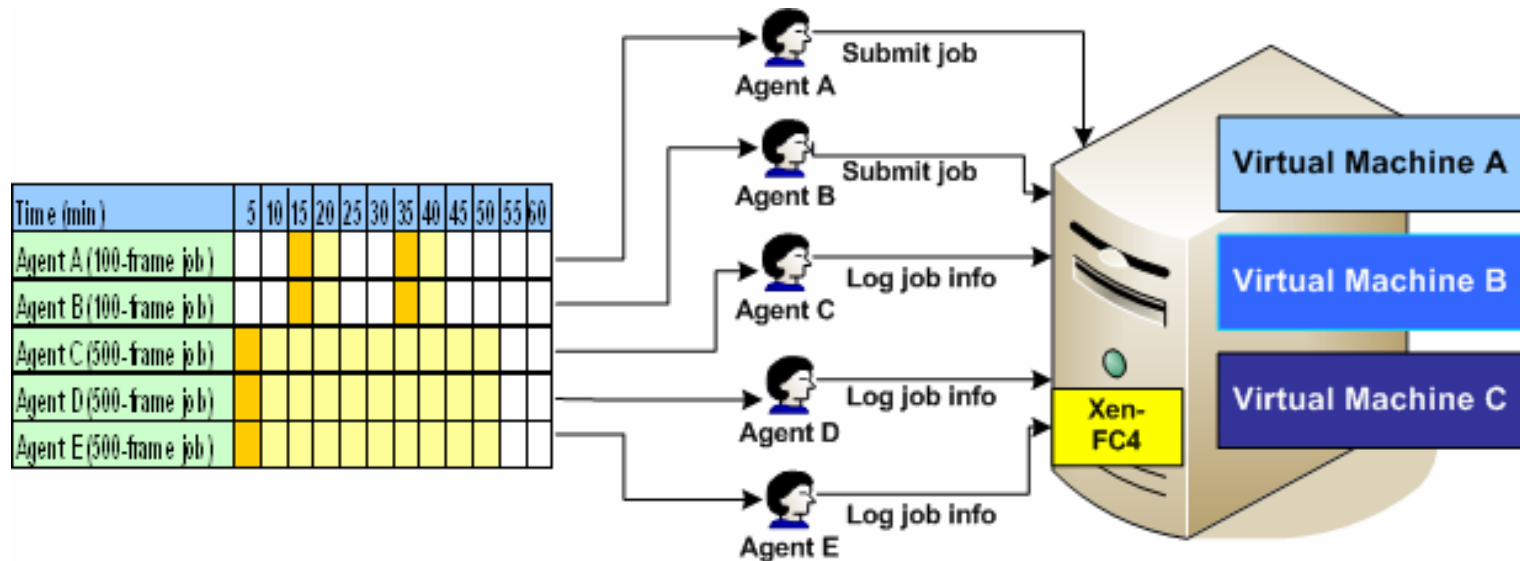
Pattern 4: Decreasing Load

Rendering jobs with decreasing durations submitted

at the start of the simulation run.

Time (min)	5	10	15	20	25	30	35	40	45	50	55	60
Agent A (100-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent B (300-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent C (300-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent D (500-frame job)	■	■	■	■	■	■	■	■	■	■	■	■
Agent E (500-frame job)	■	■	■	■	■	■	■	■	■	■	■	■

Benchmark Simulation Without Using Bidding Mechanism



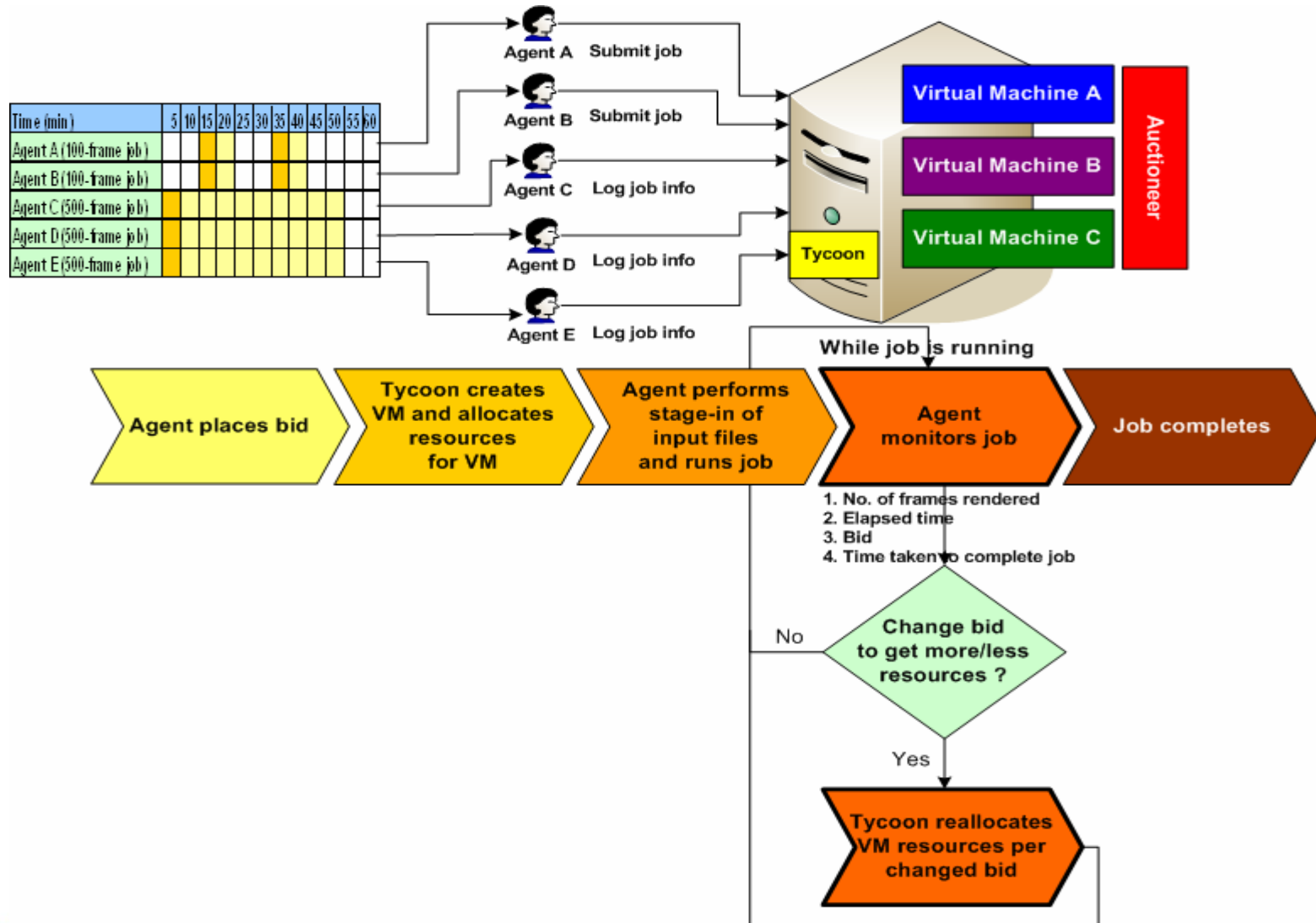
Time (min)	5	10	15	20	25	30	35	40	45	50	55	60
Agent A (100-frame job)												
Agent B (100-frame job)												
Agent C (500-frame job)												
Agent D (500-frame job)												
Agent E (500-frame job)												

While job is running

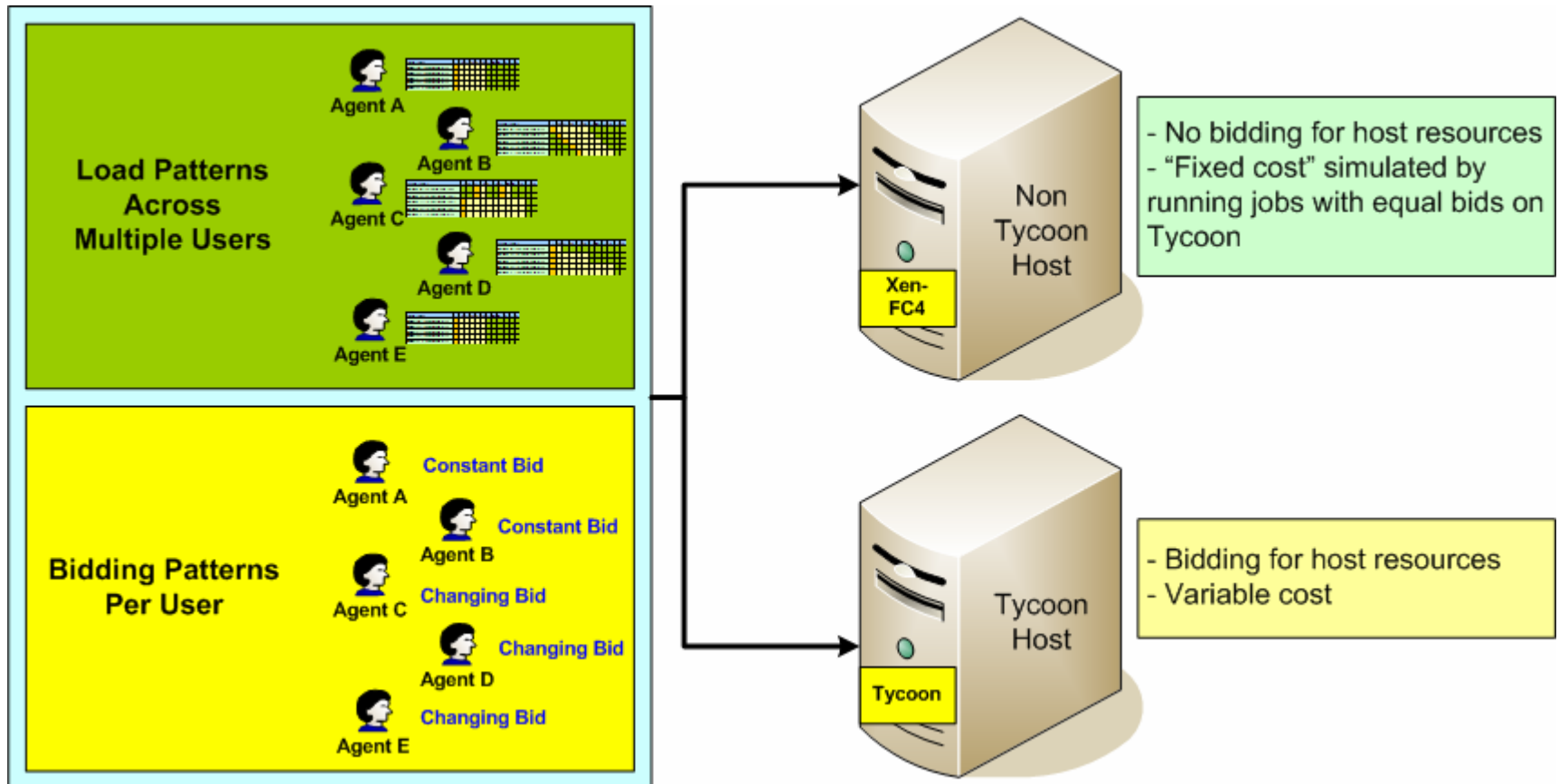


1. No. of frames rendered
2. Elapsed time
3. Time taken to complete job

Comparison Simulation using Tycoon and its Bidding Mechanism



Highlighting Differences Between the Simulations With and Without Tycoon



Empirical Perspectives - Status and Issues

- NDA obtained for NGPP Grid nodes.
- Data set from the Multi-Organizational Grid Accounting System (MOGAS) is still small.
- While NGPP usage has increased substantially, only a fraction of the usage is captured by MOGAS.
 - Reason : MOGAS is currently capturing accounting info only for jobs that are submitted using Globus.
 - Many jobs are submitted to the new Meta-Scheduler, and not through Globus and hence not being captured by MOGAS.

Empirical Perspectives - Status and Issues

- Steps to get empirical data on grid usage
 - Over time, MOGAS developer will try to integrate with Meta-Scheduler to increase fraction of submitted jobs captured by MOGAS.
 - Look for additional data sets.

To Learn More ...

**Contact us to learn more
about our work with Tycoon or
to explore possible collaboration on
Grid Economics, Grid Business Models,
and Grid Resource Allocation:**

Danny OH

dannyoh@smu.edu.sg

Prof. Steven MILLER

stevenmiller@smu.edu.sg

Asst. Prof. HU Nan

hunan@smu.edu.sg