It Costs How Much?

Rich Wolski and Andrew Mutz

Computer Science Department

University of California, Santa Barbara

UCSB

Computational Economics

- Actual economics
  - Need to study computational economics to understand investment
    - Facilities
    - Staffing
- Model for resource allocation systems
  - People think economically
  - Natural paradigm for tool design, service design, etc.
- Distributed control theory
  - Independent, self-interested, distributed agents
  - Large-scale systems are ultimately federated

Two Broad Approaches

- **Macro-economic**
  - Reason about aggregate behavior
  - "Supply" and "Demand"
  - Equilibrium theory
  - Resource stake-holder centric
- **Micro-economic**
  - Reason about the behavior of individuals
  - Game theory
  - Mechanism design
  - Auction theory
  - User centric
- **Goal: design a set of protocols and a system for implementing them that causes the system to exhibit some desired property**
  - Distinct from modeling, tool building, etc.
  - Impacts the way in which these concepts are investigated

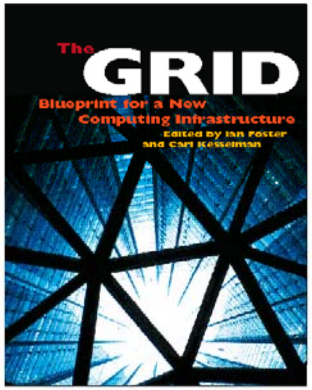
UCSB

Computational Grids as Macro-economies

- *To provide a seamless, ubiquitous, and high-performance computing environment using a heterogeneous collection of networked computers.*
- But there won't be one, big, uniform system
  - Resources must be able to come and go dynamically
  - The base system software supported by each resource must remain inviolate
  - Multiple languages and programming paradigms must be supported
  - The environment must be secure
  - Programs must run fast
- For distributed computing…The Holy Grail++

The Computational Grid

- Vision: Application programs "plug" into the system to draw computational "power" from a dynamically changing pool of resources.
  - Electrical Power Grid analogy
    - Power generation facilities == computers, networks, storage devices, palm tops, databases, libraries, etc.
    - Household appliances == application programs
- Scale to national and international levels
- Grid users (both power producers and application consumers) can join and leave the Grid at will.
- Secure, Reliable, Cheap, Fast…
- Can we build it?
  - If we do, will it be stable?



The Economics

- Current NSF Supercomputer allocation policy:
  - Committee (RAC) hands out allocations as node hours twice a year
  - Total allocation is determined by total number of nodes available from several sites
  - Users "spend" their allocations at various sites
  - NSF centers get credit for large, happy user communities
- RAC makes these decisions in an "ad hoc" way
- No really good picture of whether the process is efficient
- System is inflexible
  - No way to "trade" allocations between machines
  - Roaming allocation is a new idea

Modeling the Economics

- All agents make decisions based solely on self-interest
- Fictitious currency
  - $G (Pronounced "Grid Bucks")
- Producers and consumers are motivated to accumulate $G
- Producers and consumers are separate entities
  - Re-spending does not occur (to be investigated later)
- Agents, in aggregate, act rationally with respect to price
  - Lower price => less supply and greater demand

Transactions

- **Performance contract**
  - —A job consists of a list of resource requirements
  - —Resource requirement is a tuple:
    - (amount, duration)
  - —Producers and consumers negotiate over amount
    - A job will execute to completion once a transaction is initiated
  - —Price at the time the contract is signed persists for the duration
    - consumer is charged the negotiated price each time unit
    - producer accrues negotiated price at each time unit
  - —Entire contract must be negotiated before transaction is initiated

Pricing

- Explore two alternatives: auctions and commodities markets
- Auctions
  - Easy to implement
  - No need for global information
  - Few provable stability or equilibrium properties
- Markets (equilibrium pricing)
  - Provable stability and equilibrium characteristics
  - Accurately (fairly) reflect value
  - Requires global state information
  - More difficult to understand and implement than auctions

Markets

- **Theory**
  - Equilibrium Price: a price that equalizes supply and demand
  - Topology result (1911, 1959) says that an equilibrium must exist if all functions are continuous
  - Smale (1976) provides a constructive method for determining the equilibrium price based on Newton-Raphson
  - Price stability is harder
- **Practice**
  - Nothing is continuous => diagonalization proofs that optimality is impossible
  - Simulation is generally the final arbiter

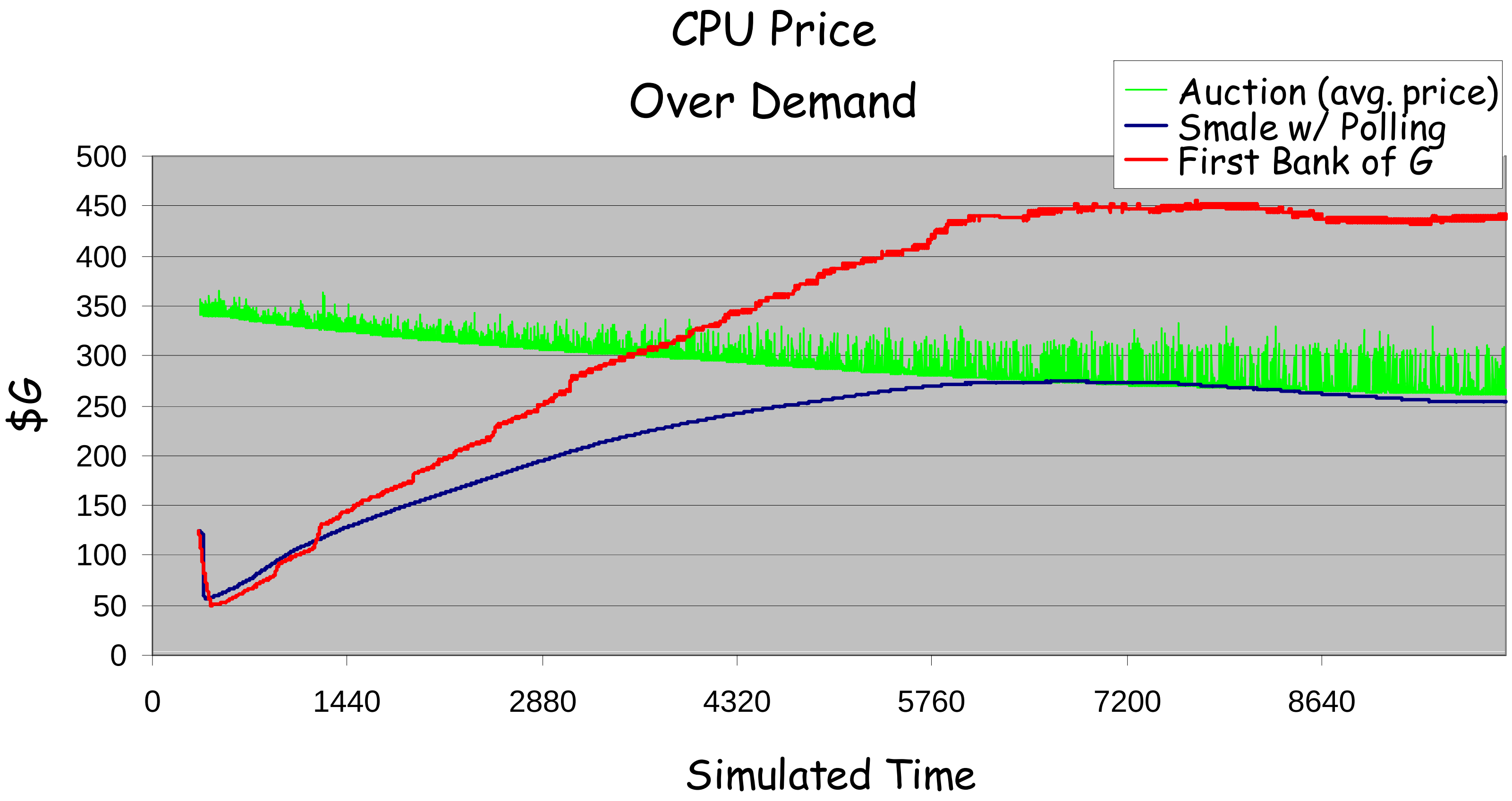
Auctions

- **Theory**
  - Uniform Second-price Auction (Vickery, 1961)
    - Sealed-bid
    - Highest bidder pays second-highest bid price
  - Reduces seller favoritism
  - Determines a prices that is "closer" to market consensus (cf. Winner's curse)
- **Practice**
  - Auctions work well when object that is for sale is unique
    - If not, buyer must participate in multiple auctions => centralized auction clearing house
  - Vickery auctions are not revenue maximizing => sellers prefer first-price English style

Simulations

- Two commodities: CPU and Disk (complements)
  - One commodity is "easy"
  - Network is still a mystery
- All jobs require a random quantity of each for a random duration
  - All distributions are uniform (for now)
- Under demand and Over demand cases
  - 100 CPUs, 100 Disks, 100 Consumers
  - 100 CPUs, 100 Disks, 500 Consumers
- Individual agent simulations
  - Not a common simulation practice in the economics community

Price Stability: Over Demand

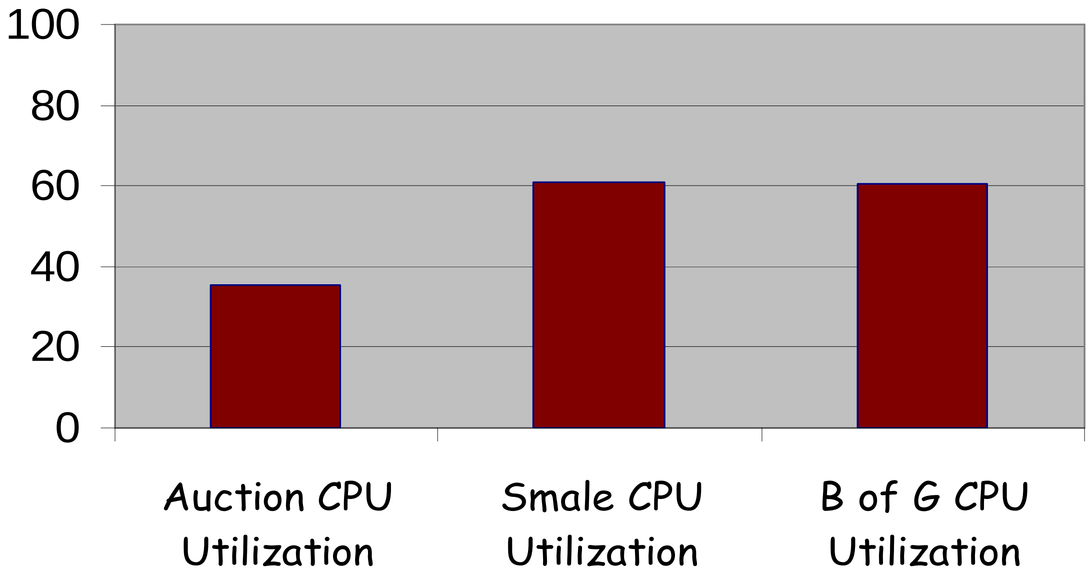


Equilibrium: Under Demand

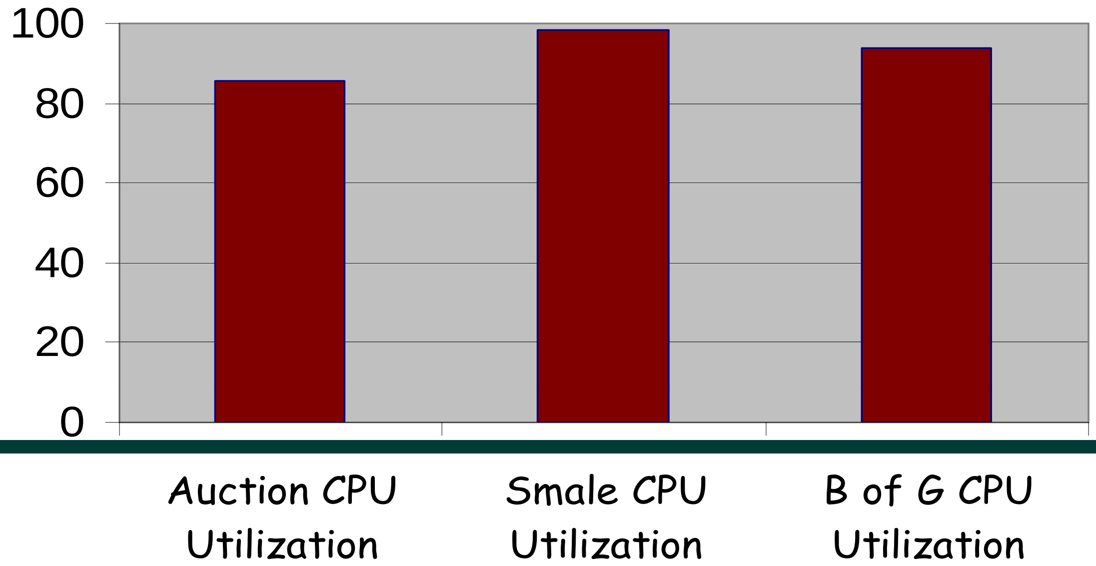
UCSB

Resource Utilization

CPU Utilization
Under Demand Case



CPU Utilization
Over Demand Case

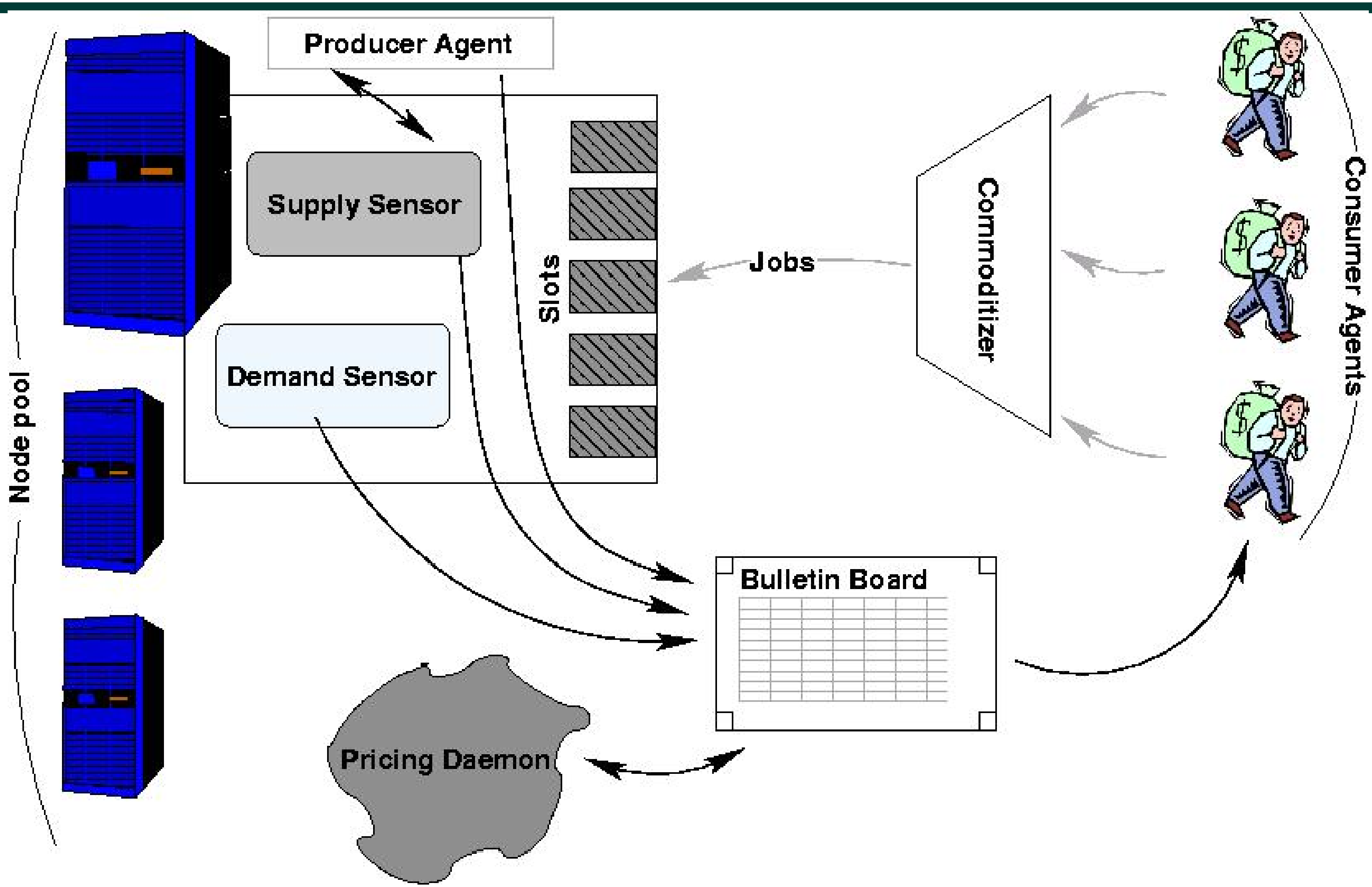


UCSB

Does it Work in Practice?

- **Build a trading infrastructure**
  - —Needs to be a "middleware" solution since we can't modify the OS
  - —Can't require privileged access
  - —Virtualization engine based on stop/start signals
    - – Each consumer gets a "slot" which represents a fraction of the available occupancy time
- **Measure its tolerances using emulation**
  - —We need to know how the system behaves under controlled conditions
- **Turn it loose with a bunch of "incentivized" users**
  - —What else are students good for?

First Bank of G Bricks and Mortar

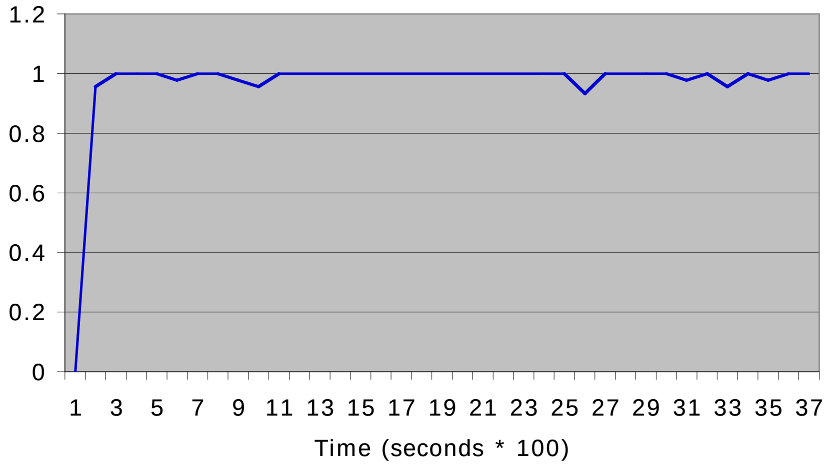


Emulation

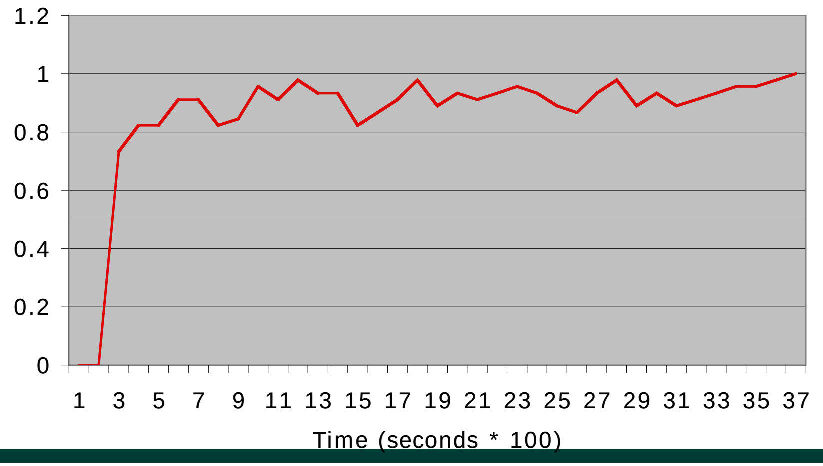
- 9 machines in the UCSB students labs
  - Two types (fast and slow)
- 5 time slots per machine
- 20 simulated consumers
  - Each computes pi sequentially
- Compare
  - Greedy scheduling
  - Economic scheduling (using Tatonnment)
- Approximately the configuration we'd use in a classroom setting

Utilization

Greedy Scheduling

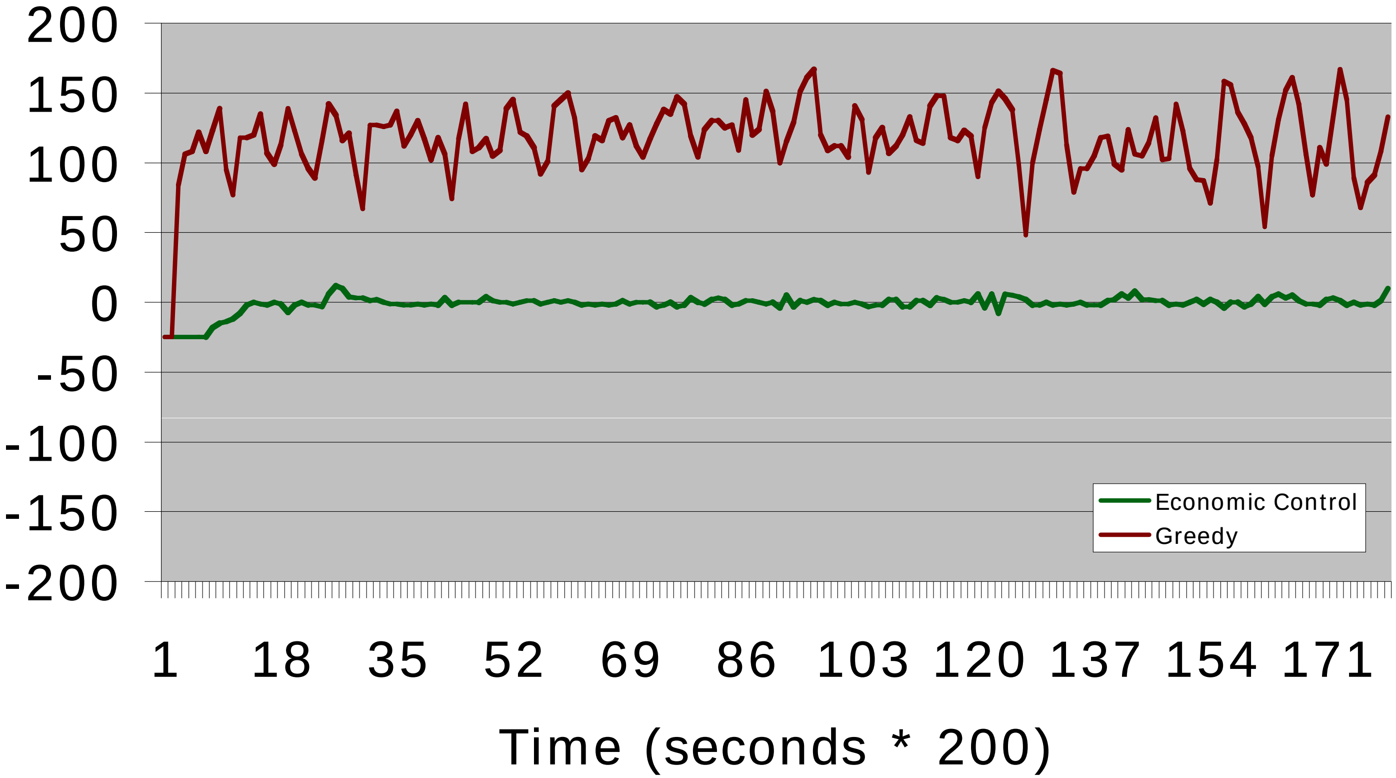


Economic Scheduling



Equilibrium

Greedy vs. Economic Control
CPU Type 1



Classroom Experiment

- 20 students
  - Believed their grade depended on economic efficiency
- Each had 6 weeks to write one or more purchasing agents
  - First Bank of G was open for business
  - First four weeks were spent coding a distributed application
- Live Market
  - Reset the bank accounts
  - Random distribution of initial prices
  - 90 minute time limit
- Result: Market Crash
  - Some purchasing agents behaved irrationally (at first)
    - Increase in price prompted increase in demand
    - Irrational agents quickly ran out of money
  - Rational agents waited until just before the time limit and then bought

UCSB

Relevance to Cloud Computing

- **The G-commerce experiments turned out to be much harder to conduct than we first anticipated**
  - Translation of theory to implementation was arduous
  - No really good models available to simulate producers and consumers
  - Very difficult to device a human factors experiment to test at any scale
  - No good theory for reboots
    - How does an economy start/restart?
- **Scale is both good and bad news**
  - Good news: scale is stabilizing
    - One "bad egg" can't crash the system
  - Bad news: scale makes efficiency difficult
    - Time lag => disequlibrium

Macro versus Micro

- **Really good reasons why equilibrium theory is not a good choice for control theory**
  - Fundamentally a commodities approach
    - Grids are heterogeneous in, Hardware, System software, administrative policy
    - The heterogeneity is dynamically changing
  - Doesn't really take user-experience into account
    - Grids must focus on user-experience for the foreseeable future
    - Commercial scalable systems venture worry a great deal about user-experience
- **Idea: use auction design theory to craft user protocols that try to ensure specific properties**

The Classical Properties

- **Incentive Compatibility**
  - Participants in the auction find truth revelation to be a dominant strategy
  - Prevents user-induced inefficiencies
    - Hoarding, conspiracy, etc.
    - Allows schedulers to "know" the true value users place on goods
- **Budget balance**
  - Net effect of transactions in the system on aggregate value is zero
  - Eliminates the need for an external entity to control inflation/deflation
    - Ensures that the system is self-sustaiing and somewhat tamper proof
- **Individual rationality**
  - It is always in the best interest of each participant to participate and not to "opt out"
- **Can have two of the above, but not all three**

Auction Systems for Space Sharing

- Best-effort batch-queued systems
  - Users queue jobs waiting for a partition of the machine
  - Scheduler tries to keep utilization high while maximizing fairness according to a hidden priority scheme
- Theory: expected externalities
  - Possible to design a protocol that achieves a Nash equilibrium with respect to truth revelation
    - Weaker than incentive compatibility
  - Budget balanced
    - Auctioneer is assumed to be altruistic
  - Not individually rational
- Working implementation
  - Protocol may not be useful in pristine form
    - Heuristics?
  - *Mutz, A, and Wolski, R., and Brevik, J., Eliciting Honest Value Information in a Batch-Queue Environment, in Proceedings of Grid2007*

UCSB

Auction Systems for Resource Reservations

- The problem is different if users wish to make and pay for reserved access
  - Combinatorial auction in its general form
    - Finding an optimal allocation of goods to bidders is NP-Hard
- For resource occupancy it may be possible to design a customized auction protocol that has tractable complexity
  - GVA based
  - Incentive compatible and budget balanced
  - "Pure" protocol may not be useful for HPC
    - Heuristics?
- Working implementation
  - Paper in preparation
- For dynamic provisioning systems, could be an approach to QoS problem

UCSB

Conclusions

- **As control theory, computational economics could be a powerful tool**
  - Provides a way to build self-sustaining, scalable resource allocation protocols for federated systems
  - Properties of the resulting systems are known a priori
    - Helps with capacity planning
    - Powerful diagnostic tool (e.g. it is clear when something has gone wrong)
- **Two approaches**
  - Equilibrium theory (macro)
    - Good for commodity systems at scale
    - Bad for user-experience
  - Auction mechanism design (micro)
    - Good for user experience
    - Bad for stability and efficiency proofs

Thanks

- NSF NGS and OCI programs
- SDSC, NCSA, PSC, TACC, TeraGrid
- John Brevik, Cal State Long Beach
- James Plank, University of Tennessee
- Todd Bryan, Rightscale
- Fred Tu, Peace Corps
- rich@cs.ucsb.edu, amutz@cs.ucsb.edu
- http://www.cs.ucsb.edu/~rich