Strategyproof Computing: Systems Infrastructures for Self-interested Parties

This post highlights the key findings of this somewhat propheticpaper:

Strategyproof Computing: Systems Infrastructures for Self-interested Parties

by Ng, Chaki, David C. Parkes, and Margo Seltzer. 2003.

This paper informs much of the current thinking of the SUAVE design as a marektplace for mechanisms. As such, I've highlighted the key finds with some deliberate notes. All credit to the authors.

h/t Andrew Miller and Quintus.

(1) Introduction

- SPC is a vision that explicitly recognizes and embraces the self-interest and incentives of individual parties in a distributed system

- Self-interest manifests in many compelling future scenarios for next-generation computing

- Scenarios in pervasive computing - depict a world where computing elements collaborate effortlessly behind the scenes in smart environments

- SPC considers the problem of designing systems with useful long-term equilibrium behavior, when users or their computation/agents have presumably figured out how to play the game to their advantage

2 threads to pull on

1. Market based approaches to resource allocation

2. Economic theory of mechanism design

3. Distributed systems should be populated with strategy proof mechanisms but without requiring a specific mechanism

4. Strategyproof computing is an infrastructure effort that seeks to enable multiple participants e.g.,

5. service providers,

6. resource providers,

7. intermediaries,

8. service providers,

9. resource providers,

10. intermediaries,

11. to describe and deploy new mechanisms, centered around services and resources on an open playing field

12. Without this validation, mechanisms cannot commit to implement particular rules and the useful simplifying effort that mechanisms can have on strategies facing participants can quickly unravel

A critical roles of the infrastructure is to validate the incentive properties of the mechanisms

(2) Strategyproof computing

Main ingredients

- Local strategyproof mechanisms

- Open market for mechanisms

- No single mechanism can possibly be appropriate for all requirements in a heterogenous system

- SPC provides the dial-tone equivalent for the development and deployment of computing services that explicitly handle incentives

(2.1) Guiding Principles

1. Incentives first

2. the self-interest of participants in distributed systems should be explicitly addressed by system designers, just like fault tolerance and security.

3. Utility-based

4. resource allocation decisions should be made in a framework that considers the utility of users for different outcomes as the basis for arbitration. Utility should be measured in terms of a common currency allowing for an explicit trade-off b/w multiple users and different components of the system.

5. Simple

6. simplify the decisions facing participants in distributed multi-agent systems such as the optimal statements to make about local capabilities e.g.,

7. available storage space,

8. speed of internet connection,

9. etc.,

or the optimal statements to make about the utility of a party for different levels of resources (e.g., bandwidth connectivity).

1. available storage space,

2. speed of internet connection,

3. etc.,

or the optimal statements to make about the utility of a party for different levels of resources (e.g., bandwidth connectivity).

1. Open systems

2. allow for innovation and competition of design of new services. A successful open system requires a lightweight infrastructure in which we only introduce universal and minimal components, to enable deployment of services without preventing future deployment of unanticipated applications.

3. Decentralized

4. the control structure in distributed computing systems must be decentralized, both to respect the autonomy of the nodes that own the property rights to resources, but also for reasons of computational scale and timeliness of information.

(2.2) Economic Foundations

2 theories from economics

1. Mechanism Design (MD)

2. explicitly models each participant as a game-theoretic agent. MD wants to design optimal incentives for preferred system-wide outcomes

3. Relaxed view

4. propose an open market for mechanisms that must only satisfy certain local incentive properties, essentially within the scope of the local deployment

5. In accepting multiple mechanisms each of limited scope, you move away from the perfect optimality and perfect incentive properties of pure MD.

6. this relaxation is necessary in the real world settings

7. open systems provide incentives for development and deployments of mechanisms

8. participants are now freed from complex modeling of other participants

9. this relaxation is necessary in the real world settings

10. open systems provide incentives for development and deployments of mechanisms

11. participants are now freed from complex modeling of other participants

12. Strategyproof mechanisms (SPM)

13. self-interested party communicates their own utility through straightforward truth-revelations about requirements and capabilities regardless of what the strategies and behaviors of other parties might be

14. truth revelation is technically the dominant strategy

15. an example of a SPM is a second price auction

16. truth revelation is technically the dominant strategy

17. an example of a SPM is a second price auction

Strategyproofness addresses the design principles of incentives first, utility based and simple.

- MD is beautiful in theory but brittle and unworkable in its purest form
- md advocates for one single central mechanisms for the entire system
- this is in direct conflict with the design principles of open and decentralized
- md ignores computational constraints on a solution and required that users agree on set of desiderata
- md advocates for one single central mechanisms for the entire system
- this is in direct conflict with the design principles of open and decentralized
- md ignores computational constraints on a solution and required that users agree on set of desiderata
- The desired simplicity that comes from strategyproofness comes at an economic cost, and there are a number of well-known economic limitations of strategyproof mechanisms.
- thus relax the goal of md
- thus relax the goal of md
- Combinatorial auctions
- heterogenous auctions are simultaneously offered to sale and participants can submit bids for explicit bundles.
- cannot simply build a combinatorial auction for the entire world
- spc realizes that mechanisms must have limited scope
- maintain reasonable computational and informational scale
- must limit the sphere of influence of a single party
- maintain reasonable computational and informational scale
- must limit the sphere of influence of a single party
- cannot simply build a combinatorial auction for the entire world
- spc realizes that mechanisms must have limited scope
- maintain reasonable computational and informational scale
- must limit the sphere of influence of a single party
- maintain reasonable computational and informational scale
- must limit the sphere of influence of a single party

SPC considers the existence of multiple mechanisms, each of which is locally strategyproof within its scope

- Local strategyproof mechanism M

is one in which truth revelations is a dominant strategy for any participant that

- can express their utility for resources and services within the scope of M

independently of other events,

- and that chooses to submit requests for these resources exclusively to the mechanim

- can express their utility for resources and services within the scope of M

independently of other events,

- and that chooses to submit requests for these resources exclusively to the mechanim

- SPC infrastructure must provide support for multiple users to design and support local strategyproof (LSP) mechanisms in a market for mechanisms.

- Achieve the second set of design principles - open

and decentralized

systems

- open marketplace leads to mechanisms with the right scope

- Achieve the second set of design principles - open

and decentralized

systems

- open marketplace leads to mechanisms with the right scope

This decision represents a trade-off between providing a large enough scope to sufficiently simplify the game-theoretic decisions facing a participant while maintaining a small enough scope to build computationally reasonable resource allocation mechanisms. The degree to which market forces lead to the emergence of mechanisms with right scope is an important research question.

(2.3) Basic Concepts

1. LSP mechanism

2. Public Interface

3. Validation

LSP Mechanism

- Resource privders or intermediaries will create LSP mechanisms for admittance into distributed systems.

- intermediaries profit by facilitating between buyers and providers

- intermediaries profit by facilitating between buyers and providers

Public Interface

- Each LSP mechanism provides a public interface that facilitates how parties can find mechanisms and composes services across mechanisms

- Part of the interface will make claims about incentive properties, and statistical properties on the utility of participants that hit that mechanisms

The interface must specify

1. Service request language - language to describe home others can query the service. At a minimum the language should Include a

2. service request message allowing parties to express their value and the resource they request.

3. a service response message for the party to accept or reject

4. service request message allowing parties to express their value and the resource they request.

5. a service response message for the party to accept or reject

6. Restrictions - on the preferences of parties that are submitting requests, for which the incentive properties of the

mechanism hold. This is important because it will depend on the particular type of goal a user wants to accomplish

7. Utility statistics - provide information on the average utility that parties have realized with the mechanism in the past

Validation

The infrastructure will assume the key role of providing validation of the incentives and statistical properties of LSP mechanisms.

- Suppose a new mechanism claims it is LSP and is admitted into the system. How can we precent it from lying?
- This is the reason for validation schemes must be provided by the infrastructure for strategyproof systems to function
- Truth revealing equilibrium of well defined mechanism can unravel without the ability to commit to a set of rules
- validation can be provide without requiring the direct checking of the rules of mechanisms
- This is the reason for validation schemes must be provided by the infrastructure for strategyproof systems to function
- Truth revealing equilibrium of well defined mechanism can unravel without the ability to commit to a set of rules
- validation can be provide without requiring the direct checking of the rules of mechanisms

Proposal: simply check the input

and the output

streams for correct behavior

- This provides only a weak from of violation - there is no evidence to support the failure of incentive properties.
- it seems useful in that it checks for failures around the operating point of mechanisms
- it seems useful in that it checks for failures around the operating point of mechanisms
- Infrastructure will assign a certificate to a newly-adopted LSP mechanism. This certificate can be presented to other parties so they can interact truthfully with confidence.
- mechanism must validate an LSP mechanism as frequently as necessary to maintain the integrity of the certificate
- mechanism must validate an LSP mechanism as frequently as necessary to maintain the integrity of the certificate

Proposal: check the claimed incentive properties by inspection of the requests made to a mechanisms and its responses

1. Active monitoring
2. with police agents that belong to the infrastructure
3. poll for LSPs and check for deviations from the claimed incentive properties
4. must be able to masquerade as a user
5. poll for LSPs and check for deviations from the claimed incentive properties
6. must be able to masquerade as a user
7. Passive monitoring
8. subset of requests and responses are monitors
9. requires that data must be stored for auditing purposes
10. similar technique to check statistical claims of LSP
11. requires that data must be stored for auditing purposes
12. similar technique to check statistical claims of LSP

(3) Validation Experiments

- Validation schemes are crucial for robustness

Proposal: validation is performed by the infrastructure without any knowledge of the auction algorithms.

- Inputs for the validation will be the bids submitted to LSPs along with the resulting allocation and payments
- Challenge - understand how to do this efficiently in terms of computational and space complexity

Experiment: determine what are the typical amount of auctions needed to detect a non-LSP mechanism in a simple setting

- Consider a multi-unit allocation problem with M

identical items and single minded agents that require particular bundle of items

- Model the valuation of agent i

by choosing the number of units

- k_i

$\sim U(1,10)$

with the value v_i

$\sim U(1, K_i)$

- consider a setting with N

= 20 agents chose from {20, 30, 40}

- simulate truthful agents to keep things simple
- Model the valuation of agent i

by choosing the number of units

- k_i

$\sim U(1,10)$

with the value v_i

$\sim U(1, K_i)$

- consider a setting with N

= 20 agents chose from {20, 30, 40}

- simulate truthful agents to keep things simple
- Validation scheme does not require truthful agents only a stream of inputs and outputs into the mechanism
- to enable the collection of counterfactual information to demonstrate failure of incentive properties
- to enable the collection of counterfactual information to demonstrate failure of incentive properties
- Consider a simple 1st price greedy auction which is a non-LSP mechanism.
- agent i

can submit a bid (x_i, p_i)

pair for x_i

units at a total price less than or equal to p_i

- auction sorts the bids in descending order of unit price p_i/x_i

and allocates items to a bid while there are items to allocate

- winning bids pay the bid price
- auction is not strategyproof because
- a winning bidder can reduce their payment and increase their utility by bidding the minimal unit price at which their bid is successful
- a winning bidder can reduce their payment and increase their utility by bidding the minimal unit price at which their bid

is successful

- agent i

can submit a bid (x_i, p_i)

pair for x_i

units at a total price less than or equal to p_i

- auction sorts the bids in descending order of unit price p_i/x_i

and allocates items to a bid while there are items to allocate

- winning bids pay the bid price

- auction is not strategyproof because

- a winning bidder can reduce their payment and increase their utility by bidding the minimal unit price at which their bid is successful

- a winning bidder can reduce their payment and increase their utility by bidding the minimal unit price at which their bid is successful

- The utility of agent i

with value (k_i, v_i)

given a bid (x_i, p_i)

is simply stated as

$$u_i(k_i, v_i; x_i, p_i) = \left\{ \begin{array}{l} v_i - p_i \\ 0 \end{array} \right. \text{ if successful and } x_i > k_i$$

- Let B

denote the set of bids submitted to an auction

- In validation take bids (x_i, p_i)

$\in$

B

and

- check whether any agent could have increased their stated utility by submitting the bid of any agent

- check whether any agent could have increased their stated utility by submitting the bid of any agent

- If a mechanism is LSP, then it is necessary that

$$u(x_i, p_i; x_i, p_i) \geq u(x_i, p_i; x_j, p_j) \text{ (1) }$$

- for all pairs of bids $i, j \in B$

- the left hand side is the utility that agent i

receives from their bid w.r.t. their stated valuation

- the right hand side is the utility that agent i

would have received from the bid of agent j

, w.r.t. their stated valuation

- the left hand side is the utility that agent i

receives from their bid w.r.t. their stated valuation

- the right hand side is the utility that agent i

would have received from the bid of agent j

, w.r.t. their stated valuation

- To check for failure of LSP it is sufficient to find a single pair of bids of which condition (1) fails.
- informally, (1) fails when a losing bidder could have won their desired number of items but for a lower price by submitting some other bid
- informally, (1) fails when a losing bidder could have won their desired number of items but for a lower price by submitting some other bid
- All three curves in figure 2 detect non-LSP auctions early
- most detections happen during the first four auctions
- most detections happen during the first four auctions
- Validation becomes more effective with higher numbers of items
- more winners, more useful counterfactual information to detect incentive violations
- losing bids can never provide useful information on the RHS of expression (1)
- more winners, more useful counterfactual information to detect incentive violations
- losing bids can never provide useful information on the RHS of expression (1)

(4) System Research Challenges

AI related challenges

- computational MD
- design of description languages
- inference methods for verification
- methods of automated composition
- machine learning methods to identify
- scope of new mechanisms
- automatically tune existing mechanisms
- scope of new mechanisms
- automatically tune existing mechanisms

(4.1) LSP Mechanisms

Basic research goal - establish the core validity of local strategyproofness in the context of system settings

- SPC mechanisms are often price-based or based around generalizations of the philosophy enshrined in Vickrey's auction
- How expressive are strategyproof mechanisms, given we allow them to be limited in scope?
- to establish core validity, must provide the development tools that will help devs to program and debug mechanisms
- to establish core validity, must provide the development tools that will help devs to program and debug mechanisms

(4.2) Validation and Verification

- Active and passive monitoring techniques are most effective if the responsible parties cooperate.
- the "police" in active monitoring must be fully trusted by the whole system and not suspected of fraud
- passive monitoring requires every party in the system to provide true, timely and accurate information
- implies research efforts into understanding and building trust and reputation techniques
- computational and economic efficiency of monitoring must be qualified

- non-LSP mechanisms must be detected early

- non-LSP mechanisms must be detected early

- the "police" in active monitoring must be fully trusted by the whole system and not suspected of fraud

- passive monitoring requires every party in the system to provide true, timely and accurate information

- implies research efforts into understanding and building trust and reputation techniques

- computational and economic efficiency of monitoring must be qualified

- non-LSP mechanisms must be detected early

- non-LSP mechanisms must be detected early

- There are many types of mechanisms and many are one time participants - must have ways to stop short-lived mechanisms from hurting other parties

- would an evaluation period where new mechanisms are validated before deployment be necessary or limiting for services?

- how much information should the audit trails provide for good validation?

- how should this be adapted for fully decentralized p2p systems?

- do we need something like third party orgs like certificate authorities to provide LSP certificates and validation?

- can groups of parties assume ad hoc tole in performing decentralized validation?

- do we need something like third party orgs like certificate authorities to provide LSP certificates and validation?

- can groups of parties assume ad hoc tole in performing decentralized validation?

- would an evaluation period where new mechanisms are validated before deployment be necessary or limiting for services?

- how much information should the audit trails provide for good validation?

- how should this be adapted for fully decentralized p2p systems?

- do we need something like third party orgs like certificate authorities to provide LSP certificates and validation?

- can groups of parties assume ad hoc tole in performing decentralized validation?

- do we need something like third party orgs like certificate authorities to provide LSP certificates and validation?

- can groups of parties assume ad hoc tole in performing decentralized validation?

(4.3) Scalability and Decentralization

- Traditional MD is completely centralized; a single node collects info from all parties and computes an outcome.

- we need to target strategies and mechanisms for decentrlaized MD

- we need to target strategies and mechanisms for decentrlaized MD

Challenges

: mechanism implemented in the nodes that stand to gain from resource allocation decision, which means that it must be rational for these nodes to choose to implement the computation and the message passing required to deploy the mechanism

- Develop fundamental building blocks including redundancy, partitioning, and dynamic selection to address challenges

(4.4) Composition

- LSP mechanisms that a party interacts with are not strategy proof necessarily - the party may behave strategically to obtain the highest possible utility from one of the mechanisms

How can we scale up with distinct LSP mechanisms?

- Need to be able to discover and compose LSP mechanisms to form larger- scope LSP mechanism sets - very SUAVE

like.

- treat the individual mechanism as the building block for all of these sets

- normal scope an average party would deal with

- normal scope an average party would deal with

- parties will not be satisfied dealing with just one mechanism for each resource

- treat the individual mechanism as the building block for all of these sets

- normal scope an average party would deal with

- normal scope an average party would deal with

- parties will not be satisfied dealing with just one mechanism for each resource

- One method is to develop mechanisms that provide real options to users rather than just outcomes from which a user cannot decommit

- ex. a party that wants to compare several LSP mechanisms may first iteratively interact with each mechanism truthfully to obtain an option

- users can simply compare the options and decide which option to exercise

- users can simply compare the options and decide which option to exercise

- ex. a party that wants to compare several LSP mechanisms may first iteratively interact with each mechanism truthfully to obtain an option

- users can simply compare the options and decide which option to exercise

- users can simply compare the options and decide which option to exercise

(4.5) Measurements

The negative effects of self-interest within a system must be clearly quantified

- Past attempts point out ways in which self-interested parties can alter a system's topology, but there does not seem to be much urgency in addressing incentive issues.

- Note that the infrastructure proposal herein exposes utility information of different parties

- Note that the infrastructure proposal herein exposes utility information of different parties

- The main difficulty is what to measure?

- crucial factors may be how many parties are or are not contributing resources

- other factors may be how much efficiency the overall system achieves in utility terms

- crucial factors may be how many parties are or are not contributing resources

- other factors may be how much efficiency the overall system achieves in utility terms

(4.6) Deployment Issues

Challenge

: Until real Strategyproof mechanisms join a particular system, the parties in the system may not observe direct benefits of strategyproof computing and hence may leave. Likewise there will be less incentive for anyone to develop a LSP mechanism.

- How can we begin to deploy strategyproof systems?

- The internet is a good place

- Planetlabs

- The internet is a good place

- Planetlabs

- What if we take the hybrid approach and target up-and-coming systems such as specific grid systems?

- larger user base then startups

- larger user base then startups

- SPC will affect parties in a system in both positive and negative ways

- important to identify effects earl on to determine incentives of particular parties who participate

- important to identify effects earl on to determine incentives of particular parties who participate

- Overall, users have much to gain from strategyproof systems as they can interact truthfully and minimize interactions to get their resources while still extracting desirable utilities

- Resource providers have costs to develop LSP mechanisms - an investment that will pay off in the long-run with user activity

- running LSP mechanisms does not imply the inability to extract revenue

- sophisticated users and resource providers in current systems may be hesitant to join strategyproof systems

- running LSP mechanisms does not imply the inability to extract revenue

- sophisticated users and resource providers in current systems may be hesitant to join strategyproof systems

(5) Conclusion

SPC is an incentives-first agenda that proposes research into the design of an appropriate infrastructure to provide the ability to embrace incentives and allow the build-out of mechanisms to handle self-interest in distributed systems.

- It is important to recognize Incentive issues in next generation distributed systems, such as p2p systems, and to design protocols to simplify strategic choices facing users

- Any new vision takes years if not decades to realize if it all.

- Systems are getting more complicated, tensions among parties are only increasing

- Any new vision takes years if not decades to realize if it all.

- Systems are getting more complicated, tensions among parties are only increasing

- The next steps

- quantify strategyproof computing costs and benefits

- provide real development methodology controls

- transform some real existing distributed systems into strategyproof ones.

- quantify strategyproof computing costs and benefits

- provide real development methodology controls

- transform some real existing distributed systems into strategyproof ones.