Markets and Clouds: Adaptive and Resilient Computational Resource Allocation inspired by Economics

Peter R. Lewis, Funmilade Faniyi, Rami Bahsoon and Xin Yao

Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA),
School of Computer Science, University of Birmingham, UK

April 2013

1 Introduction

Since the earliest days of computers, people have sought to apply them to the solving of large and complex problems. Indeed, computers' ability to solve large problems have brought benefits to humanity in fields as wide ranging as chess playing [1] and protein folding [2], amongst many others. However, key to the continued ability to apply computers to these kinds of problems is finding ways to enable them to scale massively, while remaining accessible to those who might use them. For example, it could be argued that the requirement either to own a supercomputer such as Deep Blue or have the funds and specialist knowledge to build a distributed platform such as that used by the Folding@Home project reduces accessibility.

Grid computing is one technology which attempts to address this. By providing a standard way to access computing power *on tap*, a grid platform allows users to run very large generic programs, distributed over many computational nodes [3]. Related technologies such as cloud computing [4] enable a similar standard means of access to potentially unbounded scalable computing, while service oriented architectures [5] provide a framework for distributed computational resources to be componentised and packaged up, such that distributed applications may be constructed from loosely coupled components. As platforms grow, localised failures become more likely, and systems can often no longer be assumed to be of a static nature, as its component nodes can be added and removed during run time. Therefore, key characteristics for resource allocation mechanisms to possess are that they are resilient to failures and also able to adapt in order to obtain high performance, taking advantage of system changes during run time.

Given such a range of approaches to scaling up computational capabilities, it is not surprising that computational resource allocation in such systems is not a single well defined problem. Instead, it is perhaps best described as a family of problems, each specific to the particular embodiment, but with much in common. At its heart however, the problem of computational resource allocation can be stated as follows: how should computational resources be made available to users, such as to achieve the objectives of the resource providers, the users and the system overall? It is important to note here that the term *user* does not apply solely to an end user of a computer system or their processes, but also to any *user node* which requires the use of a resource from a *provider node*.

In order to answer this question for a particular system, it is of course necessary to possess some further understanding of what is required. Does the *how* in the question refer to a particular outcome or endpoint, or perhaps instead a governing process, a set of rules or parameters to which the allocation must conform? Many approaches [6] focus on fairness and efficiency as global objectives. Furthermore, what are the objectives of the resource provider and user nodes? Are the providing nodes' objectives aligned, and do they align with the objective for the behaviour of the system as a whole, if one exists? If there is a conflict or tradeoff in achieving the objectives, how are these to be resolved?

In order to gain some perspective on these issues, it is interesting to consider Foster and Kesselman's [3] characterisation of computer systems as they scale. They note that in simple single *end machine* systems, resource allocation is typically dealt with at the operating system level, by a kernel or similar program which has absolute control over the resources in the machine. This enables it to achieve a tightly integrated system, but also provides a bottleneck, as resource requests must be fed through the kernel in order to be assigned. In *clusters*, many individual machines can communicate through message passing and file systems. Here increased scale is obtained at the expense of integration, as homogeneous nodes are controlled by a single machine responsible for job allocation. Larger still, *intranets* are characterised more by heterogeneity of nodes, which may be under administrative control of separate entities. Nodes may have different policies for use of their resources, different external demands and different capabilities. Here issues exist with regard to the availability of global knowledge. Nodes may attempt to map out the computing environment in order to plan the best use of resources, though the size and dynamic nature of such networks means that any one node is unlikely to have an accurate view of the system's current state [3]. The final category considered by Foster and Kesselman is perhaps the most interesting, that of *internets*. These forms of network span many organisations, locations and platforms and are large and heterogeneous. Here there is no central control and often no global objective with regard to resource allocation.

In this chapter, we consider the ability of economics-inspired techniques to achieve efficient allocations while also providing adaptivity and resilience. It has been shown that markets can be used to produce efficient and adaptive allocations in a range of resource allocation scenarios [7]. However, the type of market mechanism used, and how it is deployed, can have a large impact upon resilience. Many mechanisms require a centralised price fixing process such as an auctioneer or specialist, introducing a single point of failure. Other approaches use regional super-nodes within a network, creating bottlenecks and unnecessary weak points. There are also fully decentralised approaches which may be used, though these can introduce additional computational overhead. This chapter reviews these and argues that of the family of economics-inspired approaches, the retail-inspired posted offer market mechanism is a promising technique for efficient, adaptive and resilient computational resource allocation in the presence of increasing scalability.

2 Computational Resource Allocation

One prominent way of acquiring the large pool of resources required by modern software systems, such as social networks, is to rent them from the cloud [4]. The cloud makes it possible for infrastructure, platform, and software providers to publicly offer their resources on demand on a pay-per-use or subscription basis. The huge cost savings and rapid elasticity of resources, i.e. scaling out and scaling in, have made cloud a hot topic among academics and in industry. Beyond its appealing business model, cloud computing has raised interesting challenges in terms of how resources may be allocated to satisfy stakeholders' objectives.

Cloud-based systems are continuously faced with the challenge of coping with dynamics and uncertainties at run time. For example, the mode of use of cloud resources cannot be fully anticipated, hence workload patterns vary frequently. Furthermore, the cloud environment is highly volatile, as resources fail and network connections fluctuate in unexpected ways. To be successful, a cloud resource management solution must cater to these uncertainties instead of avoiding them. For these reasons, software agents are often relied upon to act on behalf of users and providers to reach their respective objectives. To clarify our understanding of what an *agent* is, we adhere to the following definition:

> Agents are computer systems [or components of such systems] that are capable of independent, autonomous action in order to satisfy their design objectives. [...] As agents have control over their own behaviour, they must cooperate and negotiate with others in order to satisfy their goals [8].

An agent's autonomy empowers it to decide whether to cooperate with other agents or not, depending on its objective. As opposed to the definition above, cooperation in our work isn't a mandatory requirement, instead, we view agents as self-interested and *fully* autonomous in their ability to make decisions.

The interaction between users and providers in the cloud environment may therefore be modelled using concepts in multi-agent systems (MAS). As advocated by Jennings [9], the MAS analogy is well suited to complex application domains, of which cloud is an example, characterised by:

- a large number of components,
- flexible (dis)connection between components, and

• complex component interconnections.

Next, we describe the cloud federation model, the objectives of its stakeholders, and the resource allocation problems it presents.

2.1 The Cloud Federation Model

The emergence of many cloud providers offering various services has propelled the vision of cloud federation [10, 11]. The proponents of the cloud federation model advocate that next generation of cloud providers will have the capacity to seamlessly interact among themselves, thereby taking advantage of economies of scale [12]. This would afford providers the possibility of outsourcing resources at run time in the event of failure of any cloud provider in the federation. Opencirrus[1] is an example of a test bed that is designed for cloud federation research. The cloud federation model is shown in figure 1.

Cloud users interact with the federation via a middleware by submitting their job requests and the associated Service Level Agreements (SLAs). The quality of service (e.g. availability, reliability and performance) expectations of the cloud users are specified in the SLAs. The middleware layer coordinates interaction with cloud users and interaction among cloud providers in the federation. Each cloud provider is equipped with a cloud manager component which interfaces with the middleware layer and coordinates the resources of its cloud. All interaction with a cloud provider is via its cloud manager component.

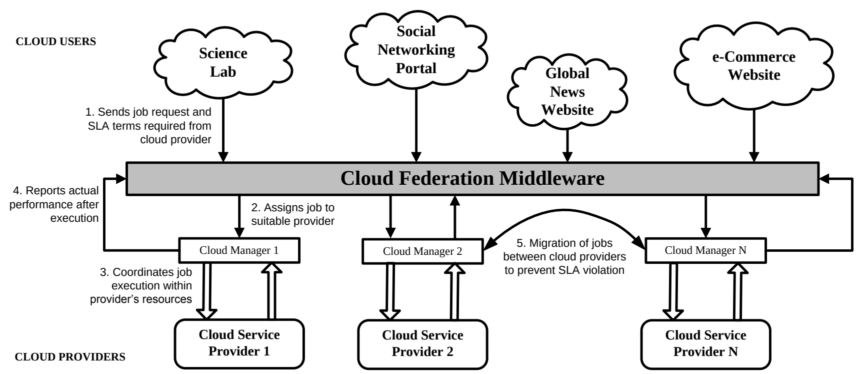

Figure 1: Cloud Federation Model [13]

To fully realise the cloud federation model, there are a number of open research problems which must be tackled. They include: formalism of a language to inform negotiation among cloud providers at run time [14], interoperability of data formats and interfaces (APIs) to facilitate inter-cloud communication [15], and middleware layer design for coordinating cloud federation resources [16]. In this chapter, we focus on the design of the cloud federation middleware.

[1] http://opencirrus.org/

According to [10], the two most important tasks of the middleware (referred to as Service Manager in their work) are:

- deploying and provisioning users' jobs based on specified configurations, and
- monitoring and enforcing SLA compliance by throttling the capacity of users' jobs.

These tasks are necessarily geared toward allocating cloud resources to users. Rochwerger et al. [10] further identified two approaches for reaching resource allocation objectives: explicit and implicit approaches. The former involves precise definition of resource allocation tactics, such as scalability thresholds and number of instances to launch or suspend. The elastic load balancer in AWS[2] is an example of a load balancer that follows the explicit approach. Implicit allocation, which is the approach taken in this work, relies on real-time monitoring and adjustment based on high-level service level objectives. Here, there is no explicit definition of resource allocation tactics, instead, the interaction of agents in a *market for resources* yields the allocation for each job request.

The importance of the middleware coordination layer in cloud federation is well acknowledged [11, 10]. This middleware layer (sometimes referred to as the Service Manager) is the highest level of abstraction responsible for coordination of cloud providers and cloud users in the federation [10]. Importantly, it ensures that cloud users' jobs are allocated to one or more cloud providers who are capable of executing those jobs without violating SLA constraints.

SLA management in the cloud is an active research area. An autonomic resource provision technique was employed by [16] to manage SLAs in federated clouds. While their work considers all phases of the SLA life cycle, there is no explicit provision for post-negotiation causes of SLA violations such as variation in workload. Brandic et al. [17] presented a proposal for SLA management in a single cloud infrastructure. Their work provides a method for mapping low-level resource metrics to high-level cloud user SLA specifications, and deducing the likelihood of SLA violations from this mapping. Another interesting approach is the autonomic resource allocator proposed by Ardagna et al. [18] for managing SLAs of multiple applications running on a single cloud. The authors considered SLA violation from the dimension of workload variation with the objective of maximising cloud providers' revenue. In reality, a broader set of events may cause these violations, namely, heterogeneous user requests, workload variations and unavailability of cloud providers in the federation.

2.2 Centralisation and Decentralisation

Classically, resource allocation objectives are achieved in a centralised manner, often relying on a single node responsible for, say, load balancing [19]. As an

[2] http://aws.amazon.com/elasticloadbalancing/

example, the AWS elastic load balancer functionality is often dedicated to a single virtualised instance, or, in larger sites, to multiple instances, in a centralised fashion. A balanced load, though by no means the only interesting outcome, can be used as an example of a desired resource allocation, an objective against which a particular approach to resource allocation may be tested. Load balancing is additionally in itself interesting, since it is useful in numerous real world scenarios, including telecommunications networks, road networks and electricity and water distribution networks. In many of these domains, even in very large scale systems, centralisation is the usual approach taken [19].

In addition to the explicit-implicit distinction described in section 2.1, resource allocation techniques can be divided into those which are stateless and those which are state-based [20]. Perhaps the most widely known and easily understood stateless approach, used to balance the load on web servers, is round-robin DNS. A more complex example is proportional share scheduling [6], in which resources are allocated to jobs according to a set of pre-determined weights. However, stateless approaches such as this are unable to take account of current server load or availability, leading to no guarantee that the desired outcome is achieved. Simple state-based extensions permit the usage of information about the resources being managed, and enable the proximity to the desired allocation to be measured. Examples of state-based resource allocation approaches include those which make use of geographical information and previous usage levels in order to determine an appropriate allocation of resource. A useful review and comparison of these approaches in the web server domain may be found in [20].

Centralised resource allocation methods do however have a number of drawbacks [21]. These include the requirements:

- that the environment remain static while the central coordinator is calculating the optimal resource allocation,
- that the coordinator has global knowledge of the system and all nodes within it,
- that all coordination messages must route through the central point, counteracting the benefit from having resources distributed about the network, reducing scalability [22] and creating a fundamentally brittle system [23].

The Internet in particular is a dynamic network, where the first two requirements are highly unlikely to be met [3]. Brittleness may be mitigated against to a certain degree, by introducing backup coordinator nodes, however even in these cases the wider system is reliant upon the existence and performance of a small number of key nodes. Failure at these key points in the network may well cripple wider functionality, at best [24].

These drawbacks lead to the need for a truly decentralised approach to the allocation of resources that does not rely on a central coordinator [21]. In the field of grid computing, examples include Cao et al.'s [25] hierarchical approach, and TURBO [22]. In the latter, allocations are achieved through the reliance on altruistic behaviour between cooperating peers, which collaborate in order to reach a global objective. Balanced overlay networks [26] are another effective and generic technique for balancing a load across a decentralised network. In this approach, resource providing nodes present an estimation of their availability to other local nodes to which they are connected. Newly arriving jobs take a random walk through the network and select the providing node with the highest availability. Upon accepting and completing a job, a provider node updates its availability estimate. In decentralised peer-to-peer storage systems Surana et al.'s [27] approach may also be used. Here the case is considered when moving loads around the network also uses bandwidth. Their objective is therefore a balance between achieving an even load and minimising the amount of load moved. Their fully decentralised approach is, in effect, tantamount to performing a centralised calculation at each node, periodically requiring cooperative reassignment of a load, based on global knowledge of the system.

2.3 Cooperation, Non-Cooperation and Self-Interest

Critically however, many decentralised approaches either rely on nodes' having complete global knowledge, or else cooperating to some extent in order to reach a shared objective [28]. As an example of this, in balanced overlay networks [26] resource users are self-interested within the bounds of the providers observed within their random walk, though the providers themselves are relied upon both to provide an honest and accurate account of their availability and to facilitate the random walk by exposing their local connections. In the case where such cooperation may not be relied upon, it is likely that the system's performance would deteriorate significantly. Similarly, Surana et al.'s [27] approach assumes both cooperation between nodes and global knowledge of the system. A non-cooperative, decentralised approach to resource allocation does exist in the domain of downloading replicated files. Dynamic parallel access schemes [29, 30] make use of self-interested smart clients to increase the speed of file downloads. It is not yet clear however, how this approach might be generalised to other service-based systems.

Buyya et al. [28] argue that we may not always be able to rely on cooperation between nodes, for several reasons. Amongst these are the possibility that a node behaves erroneously, perhaps due to a software or hardware error such as a virus, unforeseen circumstances or an external fault. Large systems are also likely to be noisy systems, as data is lost or corrupted in transit and the likelihood of measurements being inaccurate or misreported increases. Finally, limits on and delays in information transmission mean that nodes' actions may be misguided or insufficient. Crucially, Khan and Ahmad [31] show that in any decentralised cooperative approach, global optima can only be achieved when all the nodes cooperate. It is for these reasons that in seeking high resilience, we look towards approaches which do not rely on the assumed cooperation of nodes.

Some confusion does exist within the literature however in the treatment of the terms *non-cooperative* and *self-interested*. It is important to note that non-cooperation does not imply self-interest. Indeed, in Khan and Ahmad's [31] study of various games-based resource allocation methods, they describe a model in which non-cooperative agents bid for jobs based on an honest estimation of the estimated time to complete a job. Their agents, though not cooperating, act without consideration of the benefit they expect to derive from their actions. Clearly, such a consideration is a prerequisite for self-interested behaviour and hence the behaviour they describe is not self-interested.

Indeed, it is the assumption that an agent will behave either cooperatively or non-cooperatively, regardless of its predicament, that is at odds with self-interest. A self-interested agent may behave either cooperatively or non-cooperatively at certain times. The key factor is that this decision will be made by the agent, based on whether it is in its own perceived interest to do so. In making this decision, the agent must therefore consider the benefit it expects to gain from the options with which it is faced. If it does not, it cannot be said to be truly self-interested.

Therefore, when considering systems where nodes are owned or administered by separate parties, such as the very large distributed systems discussed by Foster and Kesselman [3], rather than consider agents on a cooperative / non-cooperative spectrum, it may instead be more useful to know whether or not an agent is self-interested. If it is possible to assume this of nodes, then as will be discussed in the following sections, the models and tools of economics allow for a great deal of progress to be made.

2.4 Inspiration from Economics

When selecting components with which to compose an application in a cloud system, appropriate resources may be available from a number of provider nodes. Similarly, large numbers of users may find themselves competing for access to the best resources, or a resource at a time more suited to their needs. If individual users and providers are acting in a self-interested manner in these types of computational systems, then the resulting interactions may be thought of as being an economy [7].

Indeed, large computer networks such as the Internet, made up of heterogeneous individuals with independent objectives can quite rightly be viewed as social networks as well as purely digital ones. It is perhaps of little surprise then that a social science such as economics might be useful in solving a problem such as decentralised computational resource allocation, since economics itself is concerned with the allocation of resources between individuals with different objectives in human societies. Therefore, in computational networks that are social, to what extent can economic theory be called upon in order to predict, and hopefully design the resource allocation behaviour of complex computational systems, where individual nodes are self-interested?

It is perhaps useful at this stage to present some relevant terminology. Firstly, according to Begg et al. [32] economics is "how [a] society resolves the problem of scarcity" (p3). Furthermore, they state that "a resource is scarce if the demand at a zero price would exceed the available supply" (p5). This is exactly the scenario with which we are faced in the computational resource allocation problem. There have of course been a number of different approaches to this problem in human history, but one which is particularly dominant is the use of markets. Rothbard [33] describes a *free market* as "an array of exchanges that take place in society. Each exchange is undertaken as a voluntary agreement between two people or between groups of people represented by agents." Similarly, Begg et al. [32] define a market as "a set of arrangements by which buyers and sellers are in contact to exchange goods or services" (p32). The important factors here are that there is an exchange between two or more individuals, and that this exchange is voluntarily entered into by all participants.

In order to facilitate such exchanges, a particular type of good is often agreed to serve as currency, in which case the individual giving away currency in order to obtain another good is termed the *buyer*, while that which receives the currency and gives away the other good is termed the *seller*. It is of course not required that this formal delineation be present, though it has been argued [34] that an economy will evolve towards common agreement on a particular good to treat as currency, typically that which the individuals find easiest to retain and exchange widely without additional cost. A mechanism through which voluntary exchanges between individuals are facilitated is called an *auction*, and there are many sets of rules for these, leading to a huge range of possible auction types.

A number of auction mechanisms can be found in common use, including the English auction, found amongst other places on Ebay[3]; the Dutch auction; Vickrey auction and Continuous Double Auction, often used in stock markets. Cliff [35] gives a useful introduction to and critique of several auction mechanisms, including those listed here, while Friedman and Rust [36] provide a more detailed look at the Continuous Double Auction. Purely electronic markets also make use of a range of auction mechanisms. In designing a mechanism, the aim is typically to achieve an efficient system overall, by making use of the self-interested nature of individuals. This is demonstrated by Phelps et al. [37], Byde [38] and David et al. [39] amongst others. For many, the ultimate aim of such research is the automation of the mechanism's design, appropriate to individual scenarios [40, 41, 42, 43]. Taking Cliff's [43] work as an example of this, a parametrised mechanism design space is specified, which may be searched in order to find high performing mechanisms for specific scenarios. Results from an evolutionary search demonstrate that classic, human-designed mechanisms are often far from optimal.

3 Economics-inspired Computational Resource Allocation

The application of economic ideas to resource allocation problems in computational systems is approached in the field of market-based control [7]. Using

[3]http://www.ebay.com/

the terminology of Casavant and Kuhl's [44] taxonomy of scheduling in distributed computing systems, this is a family of distributed mechanisms for dynamic global resource allocation. Such systems work by actions and decisions of resource providers and resource users nodes being automated by the use of software agents interacting in a (possibly artificial) market. The aim of a buyer agent might be to secure the fastest and most reliable resource at the lowest cost for its user. Conversely, a seller agent might aim to maximise the revenue for the resource provider, or perhaps generate high levels of business. Whatever the business strategy of the resource provider, the selling agent will be competing with similar agents from other providers for the same resource users. Each agent will therefore have to employ its own strategy for success in the market.

There are several examples of market-based control being used in decentralised computing systems. Brewer [19] proposes the idea of incorporating, into a request for web services, a notion of its value or cost. It is argued that this, along with the use of smart agents, would allow for responsive adaptation in the presence of changes to the network, as well as graceful degradation. Similarly, Gupta et al. [45] argue that in the provision of virtually zero cost per-use computational services, a mechanism involving pricing and user self-selection is preferable to the alternative of provider or regulator enforced limits: rationing. More recently, researchers have pursued in-depth study of a market-oriented cloud from various dimensions, including: price modelling [46], resource sharing among service providers [47, 48, 49], and resource allocation at the hardware layer [50]. A notable example that harnesses a game theoretic approach is the formulation and study of the service provisioning problem in cloud systems by Ardagna et al. [51]

Typically, resource owning or providing nodes are represented by selling agents, and resource users or tasks are represented by buying agents. Buyers then attempt to purchase sufficient resource to satisfy their task or user's requirements from the set of available sellers. Sellers charge an amount of (either real or artificial) money for the resource, determined by their strategy and dependent on factors such as the quantity or quality of the resource being provided. Since self-interested buyers can be expected to pay more for resources which they desire more, and self-interested sellers will charge what they can get away with in order to maximise their payoff, resources will tend to go to those who value them the most. Fundamentally, these approaches attempt to harness the rational behaviour of self-interested agents, which interact in some market environment in order to achieve resource allocation without reference to a central authority. Relying upon the theories of economics, through such repeated exchanges between utility maximising individuals, efficient resource allocations may be achieved.

3.1 Centralised Market Mechanisms

As in human economies, agents in a market-based computational system may interact through any of a number of different mechanisms [35]. Common examples include English, Dutch and Vickrey auctions, in which an auctioneer facilitates the bidding and determines the allocation of resources. Where scarcity exists on both the seller and buyer sides, double auctions such as the Continuous Double Auction and Clearing House provide an alternative approach [36]. Research in the field of automated mechanism design also suggests that other less obvious auction mechanisms may lead to more efficient outcomes in certain circumstances [43, 52, 53].

However, both Cliff and Bruten [54] and Eymann et al. [21] note that due to the mechanisms employed, a large proportion of market-based control systems are not decentralised, since they rely on a centralised price fixing process rather than the participants between them determining prices. This is true of Wolski et al.'s [55] G-Commerce model, which relies upon a central market maker. Cliff and Bruten [54] argue that the presence of such a centralised process or component removes the primary advantage of using a market-based system: its robust, decentralised, self-organising properties.

Examples of the application of centralised market mechanisms in cloud-based systems include [56] and [57]. In [56], the problem of running independent equal-sized tasks on a cloud infrastructure with a limited budget was studied. The authors concluded that a constrained computing resource allocation scheme should be benefit-aware, i.e., the heuristics for task allocation should incorporate the limited resource in supply within the system. Sun et al. [57] proposed a Nash Equilibrium based Continuous Double Auction (NECDA) cloud resource allocation algorithm for meeting performance and economic QoS objectives. In each round of the system run, each provider agent determines its requested value based on its workload, and each user agent determines its bid value based on the remaining time and resources [57]. A CDA is then used to decide the outcome resource allocation, and the existence of a Nash Equilibrium evaluated.

3.2 Distributed Market Mechanisms

A number of distributed auction mechanisms have also been proposed [58, 59, 60], which do not rely on one central coordinating node. These approaches reduce the fragility associated with reliance upon a single point, provide more scalability and allow for dynamic composition of auctions. Typically, either the central auctioneer is replaced by a number of local ones, which may communicate through some secure means, or else the auctioneer role is fulfilled by a spare, disinterested node. Double auctions for example, though relying on a specialist to match bids and asks [37], may be decentralised by the presence of multiple specialists between which the participants may choose [61, 62]. This is the approach taken by [63], where multiple specialists were used for service composition in a SaaS cloud. These techniques do reduce bottlenecks at certain points within the network and the removal of a single node cannot lead to system-wide failure. However, similarly to the replicated round-robin DNS approaches discussed in section 2.2 above, the system is still largely reliant on a small subset of its nodes.

However, it may be possible in systems such as this to scale up the number of auctioneers or specialists, in order to achieve a suitable degree of redundancy and decentralisation. This issue is an active area of research, though intuition suggests that with all else equal, a system which relies upon a set of super-nodes cannot provide the level of resilience of a system without such a need, even if the super-nodes were present in abundance. Approaches such as this also raise questions of motivation for those acting as super-nodes, as participation fees for example are set by auctioneers in most cases [61]. Therefore, if an approach exists without the need for such complexity, it should be preferred.

A further alternative is that individual provider nodes themselves host independent auctions for their resources. This approach is applied to computational resource allocation in Spawn [64]. Here, users' agents bid in sealed-bid auctions hosted by providers' agents, for their resources. In order to be effective, this requires a high level of strategic ability on the part of buyers, as they must decide in which auctions to participate. Of course, consumers may win multiple auctions, and questions then arise of how to handle these situations. Literature exists which explores the dilemma faced by buying agents bidding in multiple auctions, such as that by Gerding et al. [65, 66] though again this adds complexity.

3.3 Bargaining

Cliff and Bruten [54] conclude from their critique that, rather than depend upon a central node such as an auctioneer, market mechanisms should instead rely on the ability of intelligent agents to bargain between themselves in order to arrive at acceptable prices. This approach is taken in the AVALANCHE [67], and CATNET [68, 69, 21, 23] systems. These take inspiration from Agent-based Computational Economics (ACE) [70], an agent-based modelling technique which attempts to replicate the dynamics of human markets with complex cognitive agents.

These approaches are those which attempt to replicate human markets the most faithfully, since they rely on highly developed strategies, as agents negotiate bilaterally in order to determine the provision of a resource. It is likely in this approach that the development and operation of such strategies will themselves require significant computational overhead. While these approaches are indeed effective and widely applicable, if a simpler alternative exists, it should be preferred where possible. An additional point of interest is that in the mechanism used in CATNET [21], resource providing nodes are relied upon to forward requests to neighbouring hosts. They do this without any consideration of the effect of this on their own interests, which would appear to be at odds with the self-interested nature of the agents. The study of bargaining agents is a topic of ongoing research [71, 72, 73] and has a relevance in economics more widely than only for computational resource allocation.

3.4 Retail Markets and the Posted Offer Mechanism

Though they do not discuss them in detail, Cliff and Bruten [54] also briefly mention retail markets as an alternative to auctions and bilateral bargaining.

The mechanism used in modern retail markets is usually referred to as the *posted price* or *posted offer* model [74, 75], though in online content delivery it is sometimes referred to as the *quoted price* model [76]. It is a fully decentralised approach to the determination of price without the need for complex bilateral negotiation, and provides a potentially simpler alternative.

Wang [77] provides an interesting comparison of auction-based and posted offer selling, and shows that auctions are more commonly used in human markets where there is a greater dispersal of valuations of the good amongst the buyers. Where buyer valuations are more similar however, he favours the posted offer market mechanism. This can be reconciled with the idea that according to the most common mechanism design objectives, there exists no single dominant mechanism [53]. For an example of this in a specific case, the impossibility result due to Myerson and Satterthwaite [78] shows that no double auction can simultaneously be efficient and budget balanced while also ensuring that at least one participant would not be better off using a different mechanism. It is therefore appropriate that research into computational resource allocation continues to consider the impact of a range of mechanisms.

The application of posted offer markets [74] to computational resource allocation is the most recently proposed technique in the market-based control family [79]. The posted offer mechanism is a process in which sellers of multiple units of a good each post one price or offer, and buyers subsequently respond by stating the quantity which they wish to purchase from each seller. Exchanges then occur between buyers and sellers at these price and quantity values. Technically, the reverse process in which buyers quote prices and sellers state quantities is also a posted offer mechanism, though is less commonly encountered. Importantly, price quotations cannot be changed during the exchange period: no further negotiation is permitted, substantially reducing the burden on agents and simplifying the allocation process.

Some prior examples of the use of similar mechanisms in computational resource allocation do exist in the literature, though they are not faithful implementations and make additional assumptions. Chavez et al. [80] use an approach of this type in Challenger, where offers are broadcast to the nodes in a network, though instead of using price, bids contain an honest reporting of a job's priority. This honesty means that there is no competition between nodes and as discussed in section 2.3 this not self-interested behaviour. Xiao et al. [81] describe their system GridIS, in which buyers broadcast job requests and sellers reply by posting offers to perform them at a price. However, the behaviour of the sellers used requires certain global information in determining their price, both in the form of the latest accepted market price, which in a posted offer mechanism is considered to be private information, and also the level of aggregate supply of all the providers in the network. Again, the assumption of private information forbids this too.

One of the most faithful implementations of the posted offer mechanism in decentralised computational resource allocation is that by Kuwabara et al. [82], though they do not describe it as such. They propose an approach which in which sellers quote prices for their resources, and buyers subsequently decide the quantity (which may be zero) to purchase from each seller. Their analysis determines the quantities provided at the equilibria at which the markets arrive, and present this as a stable outcome allocation of resources. This is indeed fully decentralised, since no central component, such as an auctioneer or specialist is used; prices are determined privately by the sellers and then posted via a broadcast mechanism. More recently, we have extended this approach, investigating the behaviour of posted offer markets used for computational resource allocation in a range of homogeneous and heterogeneous contexts [79, 83].

3.5 Applicability of Economic-Inspired Approaches

Many computational resource providers, for example those in the cloud, bid to attract users by promising 'elastic' service provisioning with near-infinite scalability. In reality, clouds, just like data centres, are resource constrained and prone to failures at both node and network levels. However, in contrast to conventional data centres, cloud providers face the problem of not being able to fully anticipate the workload patterns imposed on their infrastructure in advance. These conditions make it hard to promise high qualities of service without incurring significant cost.

Economics-inspired methods offer a potential solution to alleviate this problem, providing increased resilience to internal unexpected changes (server or network failures) and adaptivity to dynamics caused by external sources (e.g. a spike or dwindle in workload) [84]. Ongoing research in this area investigates the use of market mechanisms to manage interaction of computing nodes in cloud systems. Results from cloud simulation studies (e.g. [83]) indicate that novel resource allocation methods inspired by economics can be more resilient to node failures. Due to the inherent decentralisation of many market mechanisms, they offer the capability to manage resources at the scale of cloud federations [13]. In the following section, we will give an introduction to this work, providing an example of how posted offer markets may be used for resource allocation both in an abstract problem and cloud federations.

4 Cloud Resource Allocation using Posted Offer Markets

Cloud federations are example of distributed environments in which the posted offer mechanism may be used to allocate resources. A cloud federation consists of single cloud providers who exchange (or trade) resources in order to improve their SLA compliance levels. Buyya et al. [11] envisioned the federated (or inter-cloud) model as an environment that could flexibly respond to variations in workload, network and resource conditions by dynamically coordinating multiple clouds in the federation. Since it is infeasible for a cloud provider to have data centres in every country, the federated cloud environment offers the additional benefit of rapidly scaling to meet the needs of geographically distributed cloud users than any single cloud provider [11]. The RESERVOIR project [10] also sets out a vision similar to [11] for an open federated cloud computing model to address the limited scalability of single cloud providers and lack of interoperability among them.

4.1 Motivating Example: Service Selection Problem

Consider an hypothetical on-line shopping cart application that dynamically composes its services to meet customer orders. An order typically consists of one or more products, which may be purchased and shipped from a pool of diverse services. For simplicity, we restrict the composition to four abstract services in a sequential workflow pattern (figure 2) and an order contains only one product. For more advanced workflow patterns see Jaeger et al. [85].

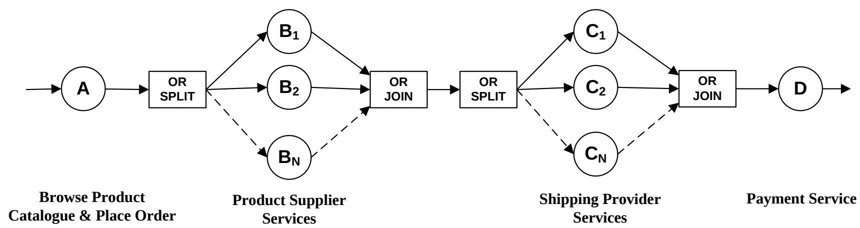

Figure 2: Online Shopping Cart Service Composition [13]

The responsibilities of the four services are defined as follows.

- Service A: renders the company's product catalogue in a browser and provides a means for customers to place orders.
- Service B: provides selected product(s) in the customer order at a specified cost. Suppose N product supplier services are available, possible options are B_1, B_2,...,B_N, of which only one is selected per product.
- Service C: offers shipping services for product(s) in the customer order within specified delivery time and at a cost. For N shipping service providers, possible options are C_1, C_2,...,C_N.
- Service D: provides payment service to collect funds from customers on behalf of the company.

Services A and D belong to the company, hence they are static. On the other hand, services B and C are provided by software-as-a-service cloud service providers (CSP), which are selected dynamically at run time. This implies that product supplier services (B_1, ...,B_N) are substitutable, similarly, supplier services (C_1, ...,C_N) are substitutable, subject to the following constraints:

- minimise product cost, and
- minimise delivery time and shipping cost.

The company's SLA objective is to *minimise the cost of meeting orders (i.e. product and shipping cost) without exceeding the promised delivery time.*

The setup of the cloud federation is shown in figure 3. The on-line shopping cart company interfaces with the cloud federation to find concrete CSP instances of the product supplier and shipping services. For each requested service, the order and associated SLA terms are submitted to the cloud federation. Here, an order request i is denoted by O_i. To meet the SLA constraint specified in customer orders, the following simplified SLA models are defined for the dynamic services.

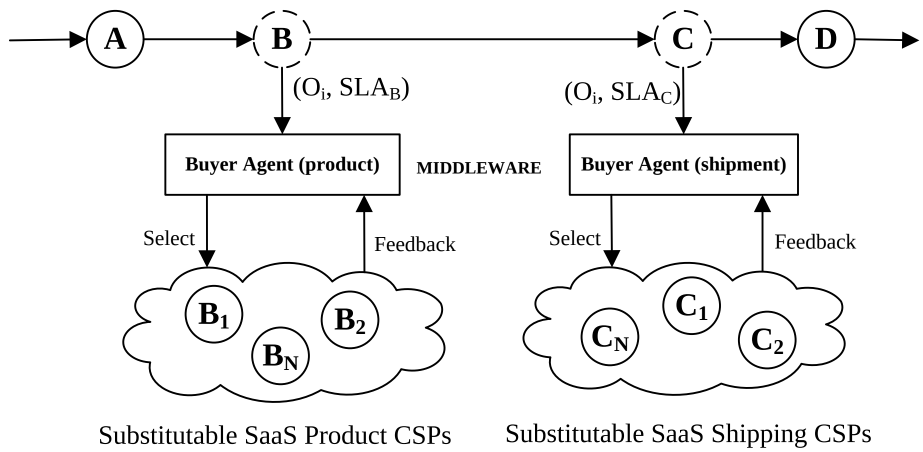

Figure 3: On-line Shopping Cart Application interfacing with two specialised cloud federations. The buyer agent (product) acts on behalf of the application to select product supplier service, while the shipment buyer agent selects shipment services on behalf of the application [13].

Each CSP (a seller in the posted offer market) publishes its cost and delivery time offerings via its cloud manager interface. In terms of the posted offer mechanism, this represents the posting of an offer. Buyer agents are specialised for their respective objectives, and hence select CSPs according to their offers. These CSPs are then instantiated in the workflow.

4.2 Bertrand-based Load Balancing

Bertrand's [86] model of economic competition is one of the simplest to account for the interactions between individual sellers who compete on price to provide homogeneous goods. The posted offer mechanism is qualitatively similar to Bertrand's model, in that it also accounts for sellers that compete on price to provide a homogeneous good to a population of buyers. In the types of computational resource allocation problems investigated here, including the above example, the good is considered homogeneous, since the buyers do not care from whom they purchase equivalent resources or services, so long as it fulfils the necessary requirements.

Many market-based resource allocation mechanisms, such as those discussed in section 3, are concerned with achieving system-wide efficient allocations. In this section however, we consider how to achieve particular outcome resource allocations in a given scenario. The approach, which is described more fully in [79] begins from the starting point of a desired allocation of resources which the system designer or owner wishes to achieve. An artificial market is then created in order to bring this allocation into effect, under the assumptions of decentralisation and self-interest.

By means of an abstract problem model, consider a scenario consisting of a set of resource providing nodes (or CSPs in the above example), S, each member of which provides an equivalent, quantitatively divisible resource π, which may vary only in price. The members of S are assumed to be self-interested. Subsequently imagine a large population of resource users or buyers, B, each member of which aims to consume some of the resource π (e.g. use the service), at regular intervals.

If s_i is a node in S and b_j is a node in B, q_{ij} is used to denote the quantity of the resource π provided by s_i to b_j. The total quantity of π provided by s_i at a given instant, its *load*, l_{s_i}, is therefore:

$$l_{s_i} = \sum_{j=1}^{|B|} q_{ij} \; . \tag{1}$$

As an example of a desired outcome resource allocation, we consider the ability of Bertrand competition to bring about a balanced load, such that at any instant, each resource providing node in S is providing an equal amount of π across the population of resource users. A particular resource allocation such as this, a configuration for the provision of π by the nodes in S at a given instant, may be expressed by the vector $\vec{L_S} = \langle l_{s_1}, l_{s_2}, ..., l_{s_n} \rangle$, where $n = |S|$. For convenience and ease of comparison between scenarios, we often normalise this vector by the total resource being provided. An evenly balanced load may therefore be written as $\langle \frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n} \rangle$.

Though this is a trivial problem when central control or cooperation may be assumed, here the objective is to achieve this using only self-interest, in a fully decentralised manner with no central or regional control, and with only private information available.

4.2.1 Mechanism and Assumptions

A posted offer mechanism is used to decide what quantity of the resource π is provided to which user node and from which provider node. At a given instant, a resource providing node, $s_i \in S$, advertises π at the price $p^{\pi}_{s_i}$ per unit via a broadcast mechanism. Each resource user, a buyer in this case, then has the option of purchasing some of the resource π, should it be in their interest to do so at the price offered. The system iterates, with sellers able to independently adapt their prices to the market conditions over time.

Each time-step, each buyer, if it chooses to buy, may purchase any amount of π from any number of resource providers in S, subject to the constraint that the total amount purchased per time-step is equal to its total required quantity (here often normalised to one unit). If no offer from any seller in S is acceptable, the buyer may instead purchase nothing. These constraints mean therefore that $\sum_{i=1}^{|S|} q_{ij} \in \{0, 1\}$ for all $b_j \in B$.

4.2.2 Buyer Behaviour

In this model, both buyers and sellers accrue a payoff, or utility gain, from their interactions in the marketplace. For buyers, this is deemed to be the value they associate with the price paid subtracted from the value they associate with the purchased resource. If buyer b_j's unit valuation of π is denoted by $v^{\pi}_{b_j}$, then its payoff from a unit transaction with s_i will be $v^{\pi}_{b_j} - p^{\pi}_{s_i}$. Since any buyer accepting a price above $v^{\pi}_{b_j}$ would lead to a negative payoff, this is its reserve or limit price. From a buyer's perspective, if a seller's price would not lead to a negative payoff for the buyer, then the price is described as being *acceptable*. S_{b_j} is used to denote the subset of S which contains exactly those sellers in S whose price is *acceptable* to buyer b_j. When buyers are homogeneous in so far as they have the same reserve prices, such that $v^{\pi}_{b_j} = v^{\pi}, \forall b_j \in B$, a set of sellers acceptable to the buyer population B exists, and is denoted as S_B. Of course $S_B \subseteq S$, or more precisely $S_B = \{s_i : s_i \in S, p^{\pi}_{s_i} \leq v^{\pi}\}$.

As with sellers, buyers are assumed to be self-interested and boundedly rational, at least insofar that they prefer higher payoffs to lower ones. As with real economic actors, this is manifested through the following of some behavioural strategy. The strategy incorporates a decision function, which given a situation describes the quantity (which may be zero) to buy from each seller. A similar approach is taken by Greenwald and Kephart [87], who model buyers as either hyperrational *bargain hunters*, seeking out the best possible price, or else *time savers* who will purchase from any *acceptable* seller, chosen at random[4]. In our work [79] we consider these two buyer behaviour models and also a third behaviour called *spread buyers*, simple risk-averse buyers, which prefer to spread their purchases across a number of sellers. The possibility of complex and arbitrary buyer decision functions means that there may not be a straightforward mapping between sellers' prices and buyer valuations, and the subsequent outcome allocation. Determining the outcome is therefore non-trivial.

Though buyers may adopt any of a number of behavioural strategies, in this chapter three representative buyer types are considered. These are Greenwald and Kephart's [87] hyperrational *bargain hunters* and *time savers* from and a further type, a risk-averse *spread buyer* behaviour [88]. These are now described.

[4] Greenwald and Kephart [87] refer to *time savers* as *any seller* or *type A* buyers.

Bargain Hunters Bargain hunters always attempt to maximise their instantaneous payoff. In each iteration, they check the prices of all the sellers, selecting the one seller which provides the most attractive offer (i.e. the lowest price). If this price is acceptable, then the buyer purchases its entire unit of π from that seller. In the event that more than one seller provides an equally attractive and acceptable offer, the buyer purchases an even proportion of π from each such seller. This is the basic model of consumers used by Bertrand [86].

Time Savers Time savers do not check the price of every seller in the system when deciding from whom to buy. Instead, they select a seller at random, and if its price is acceptable, then they purchase the entire unit of π from that seller. If it is not, then they continue selecting previously unchecked random sellers until they find an acceptable price. If no seller has an acceptable price, then they purchase nothing.

Spread Buyers *Spread buyers* are simple risk-averse agents, preferring to spread their purchases across a number of sellers. At each time-step, the buyer looks at all the available offers, and purchases a proportion of π from each seller with a price below $v^{\pi}_{b_j}$, relative to the expected utility gain from purchasing from that seller. Specifically, the quantity purchased by buyer b_j from seller s_i is determined according to the following calculation:

$$q_{ij} = \frac{(v^{\pi}_{b_j} - p^{\pi}_{s_i})}{(nv^{\pi}_{b_j} - \sum_{k=1}^{n} p^{\pi}_{s_k})} \,. \tag{2}$$

Spread buyers only consider those sellers with an acceptable price.

It is worth reinforcing that although three buyer behaviours are considered here, many other potential behaviours will exist, and can be analysed using this game theoretic methodology.

4.2.3 Seller Behaviour

Sellers also receive a payoff, defined by their payoff function. Seller s_i's payoff is denoted as P_{s_i}. In its simplest form, this is its revenue from the sale of π:

$$P_{s_i} = \sum_{j=1}^{|B|} p^{\pi}_{s_i} q_{ij} \,, \tag{3}$$

or indeed

$$P_{s_i} = p^{\pi}_{s_i} \times l_{s_i} \,. \tag{4}$$

Clearly, a seller wishing to maximise its revenue would aim to increase both its price and the quantity of its resource sold to the buyers, its market share. However as we have seen from the buyers' behaviour, the market share will depend upon the relationship between its price and those of its competitors, specifically a higher price is likely to lead to a lower market share.

4.2.4 Outcome Behaviours and Allocations

One motivation for employing an artificial market is that competition between self-interested sellers drives the system towards equilibrium. It is at this equilibrium that the system is stable in the long term, and thus we refer to the allocation of resources in this stable state as the *outcome resource allocation.*

The model described here is, in essence, a generalised version of the Bertrand game [86]. The classic Bertrand game consists of two sellers, both of whom offer to sell a certain homogeneous good to a population of buyers. Each seller must decide what price to charge for the good, and then supply the quantity subsequently demanded by the buyers. The buyers in the classic Bertrand game behave hyperrationally, as with the *bargain hunters* studied here, always buying from the seller with the lowest price, or half from each seller if the prices are identical.

In this game either seller can take the entire market by offering a price only fractionally lower than its competitor. However, since this applies to both sellers, the non-cooperative Nash equilibrium for the game is for both sellers to charge as little as possible, their zero-profit price. If each seller's costs are equal, then the equilibrium price for each seller will also be equal. This leads to the sellers sharing the market equally at equilibrium, and it is this basic idea which provides us with a balanced load in the simplest case.

However, in the more general case, where buyers may follow any of a number of strategies, calculating the expected outcome resource allocation may be a more complex task. In [79] we describe and exemplify a game theoretic methodology for calculating the expected outcome resource allocation, by determining the sellers' best response at each iteration. This is done by solving payoff equations constructed from the given buyer behaviour. This enables us to identify the Nash equilibrium outcome, where each and every seller's best response is equal to its previous position.

In the following illustrations, it is assumed that the buyers have an identical reserve price, $v^\pi = 300$, and therefore that we have a single acceptable set of sellers, S_B. Any seller in S but not in S_B will of course attract no buyers at all, and will hence receive no payoff and have a load of zero. For the sake of clarity, in the remainder of this section, only those sellers in S_B are considered.

Bargain Hunters Let us first consider a scenario with two identical resource providing nodes, such that $S = \{s_1, s_2\}$, each with costs of zero. Recalling the sellers' payoff function, given in equation 4, we have that

$$P_{s_1} = p^\pi_{s_1} \times l_{s_1} \,. \tag{5}$$

and

$$P_{s_2} = p^\pi_{s_2} \times l_{s_2} \,. \tag{6}$$

As in Bertrand competition, B is a large population of hyperrational buyers, *bargain hunters*, as described in section 4.2.2. Recalling the decision function for these buyers, and the assumption that each buyer wishes to purchase exactly one unit of π, we may therefore say that

$$P_{s_1} = \begin{cases} |B| \times p^{\pi}_{s_1} & \text{if } p_{s_1} < p_{s_2}; \\ 0.5 \times |B| \times p^{\pi}_{s_1} & \text{if } p_{s_1} = p_{s_2}; \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

and the equivalent for s_2 respectively.

From a game theoretic perspective, given an observed value for their competitor's price, both s_1 and s_2 will wish to respond with the best response. In this case, this will be to undercut the competitor's price, if possible, in order to receive the payoff given by the first case in equation 7. The competing seller will of course act similarly, leading to a price war where each undercuts the other until their zero-payoff price is reached. Assuming that a seller would rather not participate than receive a negative payoff, once $p_{s_1} = p_{s_2} = 0$, the rational course of action is to maintain a price of 0, accepting the second case.

Recalling that the current load on a resource providing node is given by equation 1 above, we therefore have that at equilibrium,

$$l_{s_1} = 0.5 \times |B|, \tag{8}$$

and

$$l_{s_2} = 0.5 \times |B|. \tag{9}$$

This is indeed an evenly balanced load, i.e.

$$\vec{L_S} = \langle \frac{1}{2}, \frac{1}{2} \rangle. \tag{10}$$

The theory of Bertrand competition (which is described more fully in [86]) demonstrates that when competing on price alone, two sellers are enough for the perfectly competitive outcome described here. Since the same logic applies to larger number of sellers, this evenly balanced outcome also holds for larger systems under the same assumptions. This idea was was first presented in [88] and elaborated upon in [79].

Time Savers Intuitively, a population of *time savers* will posses less of the *all or nothing* nature of *bargain hunters*, as each will prefer potentially any seller whose price is acceptable. Considering the simple two node example described above, what outcome should we expect with a population of *time savers*? Recalling that only those sellers in S_B are considered at present, the payoff for s_1 and s_2 should be expected to be

$$P_{s_1} = \frac{p^{\pi}_{s_1}}{|S_B|} \tag{11}$$

$$P_{s_2} = \frac{p^{\pi}_{s_2}}{|S_B|} \tag{12}$$

Here, unlike with *bargain hunters*, there is no advantage for a seller in undercutting the price of a competing seller, since this will only serve to reduce its payoff. Instead, the dominant position is to charge the highest possible price while still remaining in S_B; the equilibrium is at $p_{s_1} = p_{s_2} = v^\pi$.

Similarly to *bargain hunters* however, since $p_{s_1} = p_{s_2}$, then $\vec{L_S} \approx \langle \frac{1}{2}, \frac{1}{2} \rangle$. Note that due to the probabilistic nature of the buyers' decision function, the allocation will tend towards this as the probabilities average out.

Spread Buyers For a population of *spread buyers*, as described in section 4.2.2, the sellers' payoff functions for the simple two node case are

$$P_{s_1} = \sum_{j=1}^{|B|} \frac{v^\pi - p^\pi_{s_1}}{2v^\pi - (p^\pi_{s_1} + p^\pi_{s_2})} \times p^\pi_{s_1} \,, \tag{13}$$

and

$$P_{s_2} = \sum_{j=1}^{|B|} \frac{v^\pi - p^\pi_{s_2}}{2v^\pi - (p^\pi_{s_2} + p^\pi_{s_1})} \times p^\pi_{s_2} \,. \tag{14}$$

Sellers s_1 and s_2 will each then attempt to maximise their respective payoff function as before. The outcome resource allocation occurs when the system is at equilibrium. Figure $4a$ illustrates an example payoff function for s_1, when $v^\pi = 300$ and $p^\pi_{s_2} = 250$.

Clearly, the best response price for s_1 is less than $p^\pi_{s_2}$; in fact in this instance it is 217.71. However, given this value as $p^\pi_{s_1}$ subsequently, s_2 is then faced with the payoff function illustrated in figure $4b$. Of course, s_2 will respond to this value for $p^\pi_{s_1}$. Its best response is in this case 204.92. By using the sellers' payoff functions to iteratively calculate each seller's best response, this particular system is found to be at equilibrium when $p^\pi_{s_1} = p^\pi_{s_2} = 200$.

Clearly at this point the market share, and hence load, of each seller is also equal: $\vec{L_S} = \langle \frac{1}{2}, \frac{1}{2} \rangle$.

4.3 Deployment of Posted Offer Markets in the Cloud

Here we present an evaluation of homogeneous buyer and seller populations under two cases of the cloud service selection case study. Two buyer strategies, namely, *time savers* and *bargain hunters* (cf. section 4.2.2) are considered to understand if a trade-off exists between the timeliness of meeting an order and the selling price. In all experiments, results presented are averaged over 10 independent simulation runs to account for stochasticity.

The two cases considered are shown in table 1. An order request, O_i has a priority - High, Medium, or Low - that indicate the urgency of the request. For each order request, a buyer agent is assigned to search the seller agents (CSPs) capable of fulfilling that order.

Given an SLA, the buyer utility function is defined as

$$U_b(O_i) = w_b + (k * \beta_{price}) \tag{15}$$

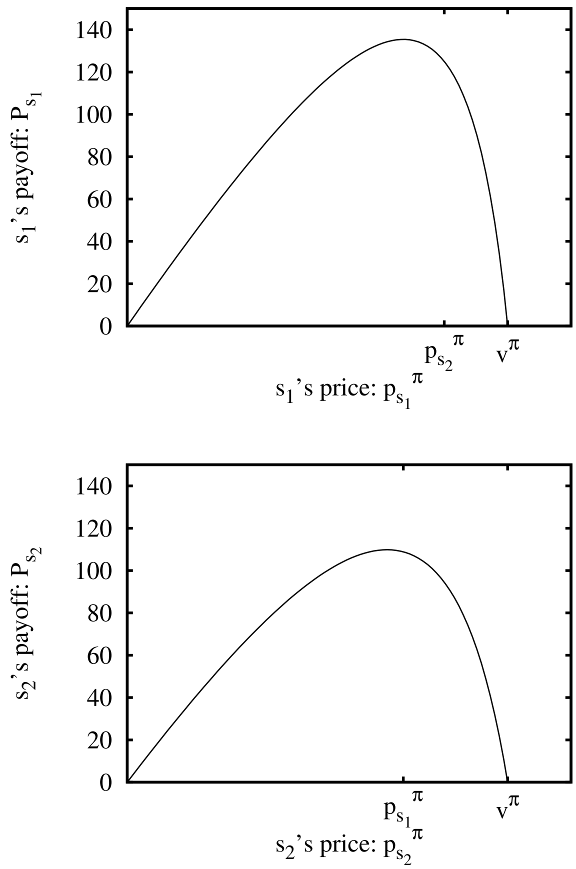

Figure 4: (a) Seller s_1's payoff function with one competitor and a population of *spread buyers*, and (b) s_2's subsequent payoff function from s_1's best response.

The value of w_b is initialised based on the order's priority. For results presented here, the tuple is defined as (Priority, w_b) = {(High, w_b=2), (Medium, w_b=1),(Low, w_b=0)}. SLA priorities are randomly assigned to orders, following a normal distribution. k is a sensitivity factor for tuning the valuation of the buyer agent. $k = 0.1$ for all experiments considered here. The value of β_{price} is derived from summation of non-functional (NF) attributes of the buyer agent, i.e. availability, reliability, and performance. Each NF attribute is randomly initialized to a value in the interval [80, 99.999].

Given an order request, the seller utility is defined as

$$U_s(O_i) = w_s * \theta_{price} \tag{16}$$

The value of w_s is initialised based on the rule

	No. Orders	Arrival rate	No. Service B	No. Service C
Case A	50	10 ticks	25	25
Case B	100	10 ticks	25	25

Table 1: Scenarios

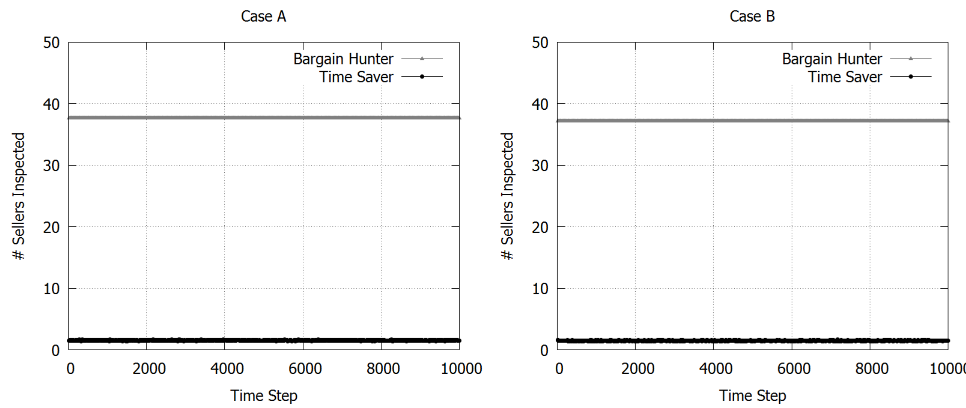

Figure 5: Trading Overhead

(Priority, w_s) = {(High, w_s=0.1), (Medium, w_s=0.01),(Low, w_s=0.001)}. Similar to β_{price}, the value of θ_{price} is derived from summation of NF attributes of the seller agent, and set to 100 for the three NF attributes.

It is worth noting that this formulation ensures that it is always possible to meet an order, although the time spent making the trading decision is non-determinant.

The overhead of using a trading strategy is measured by the number of seller agents inspected before a trading decision is made. From figure 5, it can be observed that in both cases considered, *time savers* incurred a lower overhead than *bargain hunters*. Therefore, *time savers* strictly dominate *bargain hunters* when timeliness is the critical factor.

However, on the price dimension, figure 6 shows that in both cases, *bargain hunters* always meet the order at lower prices when compared to *time savers*. This strict dominance indicates that a trade-off exists between price and timeliness when considering these strategies.

In practice, this indicative result may be used to guide the design of real software agents. That is, specific software agents may be deployed in order to implement the appropriate strategy for the context of the order at hand.

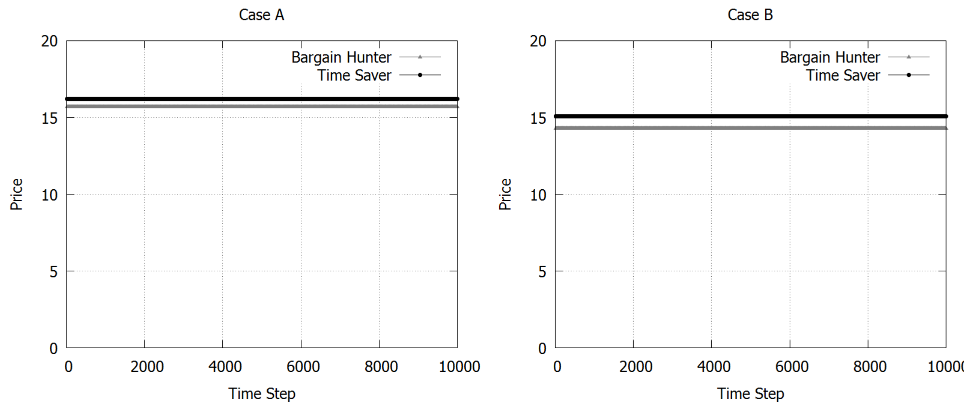

Figure 6: Trading Price

5 Conclusions

Emerging paradigms for the development and deployment of massively distributed computational systems allow resources to span many locations, organisations and platforms, connected through the Internet. In such systems, both resource providing and resource using nodes may arrive, organise and dissipate, as computational capabilities are formed and reformed as needed, without reference to a central authority or coordinator.

As these systems mature, it is predicted that the majority of their interactions will be carried out by autonomous software agents on behalf of their owners. In such distributed systems, where there exists a distribution of work to be done or resource to be provided about a network of nodes, neither control nor even full knowledge of key resources may be assumed, as they may be owned or administered by different organisations or individuals and as such have independent objectives. There is a need to find novel ways to understand and autonomically manage and control these large, decentralised and dynamic systems. As part of this, there remains the problem of how to allocate distributed resources amongst the nodes in an adaptive and resilient way.

In this chapter, we have described a range of techniques that take inspiration from economics. These provide methods for modelling such problems and reconciling conflicting nodes' objectives in an adaptive manner. In particular, game theoretic analysis is a useful tool with which to reason about the interactions between self-interested adaptive agents. A number of different approaches to implementing economics-inspired resource allocation have been proposed. However, these approaches vary in terms of the resilience (or lack of resilience) that they provide. Single and double sided auctions, typically either require a centralised price fixing process such as an auctioneer or specialist, or else regional super-nodes able to perform this function in a distributed manner. Both approaches require information to be channelled through one or more coordinating nodes, introducing weak points in the system and potentially creating bottlenecks. An alternative to this is bilateral bargaining, and this shows a great deal of promise as a fully decentralised and more resilient approach. However it seems likely that this requires highly complex agent capabilities throughout the system, which will come with their own computational overhead. Furthermore, when agents are unable to fulfil this role, they will most likely be disadvantaged. The simpler retail-inspired posted offer market mechanism provides a further promising alternative. Here agents are not required to possess complex strategic capabilities, reducing computational overhead, and the mechanism does not require global or regional coordination nodes, increasing resilience.

We have shown how posted offer markets may be applied to an abstract problem model, motivated by the service selection problem in cloud computing, in order to achieve a balanced load across multiple resource providers. We outlined a methodology for analysing outcome resource allocations, given arbitrary buyer behaviour models. We argued that different buyer behaviour types may indeed be relevant for deployment in cloud-based systems, since they possess different characteristics representative of users' preferences over quality of service attributes.

In this chapter we focussed on cloud computing as a case study, but economics-inspired techniques for resource allocation can equally be applied to a wide range of computational and engineering problems. Other recent examples include the use of auctions to adaptively allocate object tracking responsibilities among nodes in smart camera networks [89], and for conflict resolution in multi-user active music systems [90]. In the smart camera network case, analysis of the performance of the system in the presence of node and network failures, and node additions during run time, have shown high levels of resilience and advantageous adaptivity [91]. On the subject of computational resource allocation generally, there is much knowledge transfer between research in clouds and in other decentralised systems, and each application area brings with it its own set of challenging assumptions.

From a conceptual perspective, future research into market-based control and economics-inspired computation must therefore consider that behaviours of participating agents can not be assumed to be theoretically optimal and may adapt in unpredictable ways. Similarly, mechanisms used may need to vary by deployment, as different mechanisms themselves possess characteristics suitable for different assumptions and quality of service requirements, most notably resilience. It is therefore important that research into economics-inspired computational resource allocation continues to consider a wide range of behavioural strategies and market mechanisms.

Acknowledgment

This research was supported by the EPiCS project and received funding from the European Union Seventh Framework Programme under grant agreement

n^{o} 257906. `http://www.epics-project.eu/`

References

[1] F. Hsu, *Behind Deep Blue: Building the Computer that Defeated the World Chess Champion*. Princeton University Press, 2002.

[2] A. L. Beberg, D. L. Ensign, G. Jayachandran, S. Khaliq, and V. S. Pande, "Folding@home: lessons from eight years of distributed computing," in *IEEE International Symposium on Parallel and Distributed Processing (IPDPS)*, pp. 1–8, 2009.

[3] I. Foster and C. Kesselman, eds., *The Grid: Blueprint for a New Computing Infrastructure*. Morgan-Kaufman, 1999.

[4] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," in *2008 10th IEEE International Conference on High Performance Computing and Communications*, pp. 5–13, IEEE, 2008.

[5] M. P. Singh and M. N. Huhns, *Service-Oriented Computing: Semantics, Processes, Agents*. Chichester, West Sussex: John Wiley and Sons, 2005.

[6] K. Lai, "Markets are dead, long live markets," *ACM SIGecom Exchanges*, vol. 5, no. 4, pp. 1–10, 2005.

[7] S. H. Clearwater, ed., *Market-Based Control: A Paradigm for Distributed Resource Allocation*. Singapore: World Scientific, 1996.

[8] P. Ciancarini and M. Wooldridge, "Agent-oriented software engineering," in *Software Engineering, 2000. Proceedings of the 2000 International Conference on*, pp. 816–817, 2000.

[9] N. Jennings, "Agent-oriented software engineering," in *Multi-Agent System Engineering* (F. Garijo and M. Boman, eds.), vol. 1647 of *Lecture Notes in Computer Science*, pp. 1–7, Springer Berlin Heidelberg, 1999.

[10] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Caceres, M. Ben-Yehuda, W. Emmerich, and F. Galan, "The reservoir model and architecture for open federated cloud computing," *IBM Journal of Research and Development*, vol. 53, no. 4, pp. 4:1 –4:11, 2009.

[11] R. Buyya, R. Ranjan, and R. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Algorithms and Architectures for Parallel Processing* (C.-H. Hsu, L. Yang, J. Park, and S.-S. Yeo, eds.), vol. 6081 of *Lecture Notes in Computer Science*, pp. 13–31, Springer Berlin / Heidelberg, 2010.

[12] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to enhance cloud architectures to enable cross-federation," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pp. 337 –345, 2010.

[13] F. Faniyi, R. Bahsoon, and G. Theodoropoulos, "A dynamic data-driven simulation approach for preventing service level agreement violations in cloud federation," *Procedia Computer Science*, vol. 9, no. 0, pp. 1167 – 1176, 2012.

[14] I. Brandic, D. Music, and S. Dustdar, "Service mediation and negotiation bootstrapping as first achievements towards self-adaptable grid and cloud services," in *Proceedings of the 6th international conference industry session on Grids meets autonomic computing*, GMAC '09, (New York, NY, USA), pp. 1–8, ACM, 2009.

[15] N. Loutas, E. Kamateri, F. Bosi, and K. Tarabanis, "Cloud computing interoperability: The state of play," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, pp. 752 –757, 2011.

[16] P. Rubach and M. Sobolewski, "Autonomic SLA management in federated computing environments," in *Proceedings of the 2009 International Conference on Parallel Processing Workshops*, ICPPW '09, (Washington, DC, USA), pp. 314–321, IEEE Computer Society, 2009.

[17] I. Brandic, V. C. Emeakaroha, M. Maurer, S. Dustdar, S. Acs, A. Kertesz, and G. Kecskemeti, "LAYSI: A layered approach for SLA-violation propagation in self-manageable cloud infrastructures," in *Proceedings of the 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops*, COMPSACW '10, (Washington, DC, USA), pp. 365–370, IEEE Computer Society, 2010.

[18] D. Ardagna, M. Trubian, and L. Zhang, "SLA based resource allocation policies in autonomic environments," *J. Parallel Distrib. Comput.*, vol. 67, pp. 259–270, 2007.

[19] E. A. Brewer, "Lessons from giant-scale services," *IEEE Internet Computing*, vol. 5, no. 4, pp. 46–55, 2001.

[20] V. Cardellini, M. Colajanni, and P. S. Yu, "Dynamic load balancing on web server systems," *IEEE Internet Computing*, vol. 3, no. 3, pp. 28–39, 1999.

[21] T. Eymann, M. Reinicke, O. Ardaiz, P. Artigas, L. D. de Cerio F., Freitag, R. Messeguer, L. Navarro, D. Royo, and K. Sanjeevan, "Decentralized vs. centralized economic coordination of resource allocation in grids," *Lecture notes in computer science*, pp. 9–16, 2004.

[22] R. Alfano and G. D. Caprio, "TURBO: An autonomous execution environment with scalability and load balancing features," in *Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and its Applications (DIS06)*, pp. 377–382, 2006.

[23] O. Ardaiz, P. Artigas, T. Eymann, F. Freitag, L. Navarro, and M. Reinicke, "The catallaxy approach for decentralized economic-based allocation in grid resource and service markets," *Applied Intelligence*, vol. 25, no. 2, pp. 131–145, 2006.

[24] V. Ramasubramanian and E. G. Sirer, "Perils of transitive trust in the domain name system," Tech. Rep. TR2005-1994, Cornell University, Ithaca, New York, USA, 2005.

[25] J. Cao, D. P. Spooner, S. A. Jarvis, and G. R. Nudd, "Grid load balancing using intelligent agents," *Future Generation Computer Systems*, vol. 21, no. 1, pp. 135–149, 2005.

[26] J. S. A. Bridgewater, P. O. Boykin, and V. P. Roychowdhury, "Balanced overlay networks (BON): An overlay technology for decentralized load balancing," *IEEE Transactions on Parallel and Distributed Systems*, pp. 1122–1133, 2007.

[27] S. Surana, B. Godfrey, K. Lakshminarayanan, R. Karp, and I. Stoica, "Load balancing in dynamic structured peer-to-peer systems," *Performance Evaluation*, vol. 63, no. 3, pp. 217–240, 2006.

[28] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economic models for resource management and scheduling in grid computing," *Concurrency and computation: practice and experience*, vol. 14, no. 13-15, pp. 1507–1542, 2002.

[29] P. Rodriguez and E. W. Biersack, "Dynamic parallel access to replicated content in the internet," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 455–465, 2002.

[30] R. Chang, M. Guo, and H. Lin, "A multiple parallel download scheme with server throughput and client bandwidth considerations for data grids," *Future Generation Computer Systems*, vol. 24, no. 8, pp. 798–805, 2008.

[31] S. U. Khan and I. Ahmad, "Non-cooperative, semi-cooperative, and cooperative games-based grid resource allocation," in *20th International Parallel and Distributed Processing Symposium, 2006*, p. 10, 2006.

[32] D. K. Begg, S. Fischer, and R. Dornbusch, *Economics.* London, UK: McGraw-Hill, 7th ed., 2002.

[33] M. N. Rothbard, "Free market," in *The Concise Encyclopedia of Economics.* http://www.econlib.org/.

[34] R. M. Starr, "Commodity money equilibrium in a Walrasian trading post model: An example." eScholarship Repository, University of California, 2006.

[35] D. Cliff, "Minimal-intelligence agents for bargaining behaviors in market-based environments," Tech. Rep. HPL-97-91, HP Labs, Bristol, UK, 1997.

[36] D. P. Friedman and J. Rust, *The Double Auction Market: Institutions, Theories, and Evidence*. Boulder, Colorado, USA: Westview Press, 1993.

[37] S. Phelps, S. D. Parsons, and P. McBurney, "An evolutionary game-theoretic comparision of two double-auction market designs," in *Proceedings of the 6th Workshop on Agent Mediated Electronic Commerce*, (New York, NY, USA), 2004.

[38] A. Byde, "A comparison between mechanisms for sequential compute resource auctions," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-agent Systems*, (Hakodate, Japan), 2006.

[39] E. David, A. Rogers, N. R. Jennings, J. Schiff, S. Kraus, and M. H. Rothkopf, "Optimal design of english auctions with discrete bid levels," *ACM Transactions on Internet Technology*, vol. 7, no. 2, p. 12, 2007.

[40] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. D. Parsons, C. Sierra, and M. Wooldridge, "Automated negotiation: Prospects, methods and challenges," *International Journal of Group Decision and Negotiation*, vol. 10, no. 2, pp. 199–215, 2001.

[41] A. Byde, "Applying evolutionary game theory to auction mechanism design," in *IEEE International Conference on E-Commerce*, pp. 347–354, 2003.

[42] V. Conitzer and T. Sandholm, "Self-interested automated mechanism design and implications for optimal combinatorial auctions," in *Proceedings of the 5th ACM Conference on Electronic Commerce*, (New York, NY, USA), pp. 132–141, ACM Press, 2004.

[43] D. Cliff, "Explorations in evolutionary design of online auction market mechanisms," *Journal of Electronic Commerce Research and Applications*, vol. 2, no. 2, pp. 162–175, 2003.

[44] T. L. Casavant and J. G. Kuhl, "A taxonomy of scheduling in general-purpose distributed computing systems," *IEEE Transactions on Software Engineering*, vol. 14, no. 2, pp. 141–154, 1988.

[45] A. Gupta, D. O. Stahl, and A. B. Whinston, "The economics of network management," *Communications of the ACM*, vol. 42, no. 9, pp. 57–63, 1999.

[46] Y. C. Lee, C. Wang, A. Y. Zomaya, and B. B. Zhou, "Profit-driven service request scheduling in clouds," in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, CCGRID '10, (Washington, DC, USA), pp. 15–24, IEEE Computer Society, 2010.

[47] I. Fujiwara, K. Aida, and I. Ono, "Applying double-sided combinational auctions to resource allocation in cloud computing," in *Proceedings of the 2010 10th IEEE/IPSJ International Symposium on Applications and the Internet*, SAINT '10, (Washington, DC, USA), pp. 7–14, IEEE Computer Society, 2010.

[48] B. Song, M. M. Hassan, and E.-n. Huh, "A novel heuristic-based task selection and allocation framework in dynamic collaborative cloud service platform," in *Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science*, CLOUDCOM '10, (Washington, DC, USA), pp. 360–367, IEEE Computer Society, 2010.

[49] S. Shang, J. Jiang, Y. Wu, G. Yang, and W. Zheng, "A knowledge-based continuous double auction model for cloud market," in *Proceedings of the 2010 Sixth International Conference on Semantics, Knowledge and Grids*, SKG '10, (Washington, DC, USA), pp. 129–134, IEEE Computer Society, 2010.

[50] X. You, X. Xu, J. Wan, and D. Yu, "RAS-M: Resource allocation strategy based on market mechanism in cloud computing," in *ChinaGrid Annual Conference, 2009. ChinaGrid '09. Fourth*, pp. 256 –263, 2009.

[51] D. Ardagna, B. Panicucci, and M. Passacantando, "Generalized nash equilibria for the service provisioning problem in cloud systems," *Services Computing, IEEE Transactions on*, vol. PP, no. 99, p. 1, 2012.

[52] V. Walia, A. Byde, and D. Cliff, "Evolving market design in zero-intelligence trader markets," in *Proceedings of the IEEE International Conference on E-Commerce*, pp. 157–163, IEEE Computer Society, 2003.

[53] S. Phelps, P. McBurney, and S. Parsons, "Evolutionary mechanism design: a review," *Autonomous Agents and Multi-Agent Systems*, vol. 21, no. 2, pp. 237–264, 2009.

[54] D. Cliff and J. Bruten, "Simple bargaining agents for decentralized market-based control," Tech. Rep. HPL-98-17, HP Laboratories, Bristol, UK, 1998.

[55] R. Wolski, J. Plank, J. Brevik, and T. Bryan, "Analyzing market-based resource allocation strategies for the computational grid," *International Journal of High Performance Computing Applications*, vol. 15, no. 3, p. 258, 2001.

[56] W. Shi and B. Hong, "Resource allocation with a budget constraint for computing independent tasks in the cloud," in *Cloud Computing Technology*

and Science (CloudCom), 2010 IEEE Second International Conference on, pp. 327 –334, 2010.

[57] D. Sun, G. Chang, C. Wang, Y. Xiong, and X. Wang, “Efficient nash equilibrium based cloud resource allocation by using a continuous double auction,” in *Computer Design and Applications (ICCDA), 2010 International Conference on*, vol. 1, pp. V1–94 –V1–99, 2010.

[58] M. Esteva and J. Padget, “Auctions without auctioneers: Distributed auction protocols,” in *Lecture Notes in Artificial Intelligence*, vol. 1788, ch. II, pp. 20–28, Berlin, Germany: Springer-Verlag, 2000.

[59] D. Hausheer and B. Stiller, “Peermart: The technology for a distributed auction-based market for peer-to-peer services,” in *Proceedings of the IEEE International Conference on Communications*, vol. 3, pp. 1583–1587, 2005.

[60] H. Kikuchi, S. Hotta, K. Abe, and S. Nakanishi, “Distributed auction servers resolving winner and winning bid without revealing privacy of bids,” in *Proceedings of the Seventh International Conference on Parallel and Distributed Systems: Workshops*, (Washington, DC, USA), p. 307, IEEE Computer Society, 2000.

[61] J. Niu, K. Cai, S. Parsons, E. Gerding, P. McBurney, T. Moyaux, S. Phelps, and D. Shield, “JCAT: a platform for the TAC market design competition,” in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: Demo papers*, pp. 1649–1650, International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[62] E. Robinson, P. McBurney, and X. Yao, “How Specialised Are Specialists? Generalisation Properties of Entries from the 2008 and 2009 TAC Market Design Competitions,” in *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets* (E. David, E. Gerding, D. Sarne, and O. Shehory, eds.), vol. 59 of *Lecture Notes in Business Information Processing*, pp. 178–194, Springer Berlin Heidelberg, 2010.

[63] V. Nallur and R. Bahsoon, “A decentralized self-adaptation mechanism for service-based applications in the cloud,” *Software Engineering, IEEE Transactions on*, 2012. doi:10.1109/TSE.2012.53.

[64] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart, and S. Stornetta, “Spawn: A distributed computational economy,” *IEEE Transactions on Software Engineering*, vol. 18, no. 2, pp. 103–117, 1992.

[65] E. H. Gerding, R. K. Dash, D. C. K. Yuen, and N. R. Jennings, “Optimal bidding strategies for simultaneous vickrey auctions with perfect substitutes,” in *Proceedings of the 8th Workshop on Game Theoretic and Decision Theoretic Agents*, (Hakodate, Japan), 2006.

[66] E. H. Gerding, R. K. Dash, D. C. K. Yuen, and N. R. Jennings, "Bidding optimally in concurrent second-price auctions of perfectly substitutable goods," in *Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-07)*, (Honolulu, Hawaii, USA), pp. 267–274, 2007.

[67] T. Eymann, "Co-evolution of bargaining strategies in a decentralized multi-agent system," in *AAAI Fall 2001 Symposium on Negotiation Methods for Autonomous Cooperative Systems*, 2001.

[68] T. Eymann, B. Padovan, and D. Schoder, "The catallaxy as a new paradigm for the design of information systems," in *Proceedings of the 16th World Computer Congress of the International Federation for Information Processing*, 2000.

[69] T. Eymann, M. Reinickke, O. Ardaiz, P. Artigas, F. Freitag, and L. Navarro, "Self-organizing resource allocation for autonomic networks," in *Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on*, pp. 656–660, 2003.

[70] L. Tesfatsion and K. Judd, *Handbook of computational economics*. Elsevier, 2006.

[71] E. H. Gerding and J. A. La Poutr, "Bilateral bargaining with multiple opportunities: Knowing your opponent's bargaining position," *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 36, no. 1, pp. 45–55, 2006.

[72] F. Lopes, M. Wooldridge, and A. Q. Novais, "Negotiation among autonomous computational agents: principles , analysis and challenges," *Artificial Intelligence Review*, vol. 2008, no. 29, pp. 1–44, 2008.

[73] A. Chandra, P. S. Oliveto, and X. Yao, "Co-evolution of optimal agents for the alternating offers bargaining game," in *EvoApplications*, vol. 1, pp. 61–70, 2010.

[74] C. R. Plott and V. L. Smith, "An experimental examination of two exchange institutions," *The Review of economic studies*, pp. 133–153, 1978.

[75] J. Ketcham, V. L. Smith, and A. W. Williams, "A comparison of posted-offer and double-auction pricing institutions," *The Review of Economic Studies*, vol. 51, no. 4, pp. 595–614, 1984.

[76] S. Jagannathan and K. C. Almeroth, "Price issues in delivering e-content on-demand," *ACM SIGecom Exchanges*, vol. 3, no. 2, pp. 18–27, 2002.

[77] R. Wang, "Auctions versus posted-price selling," *The American Economic Review*, vol. 83, no. 4, pp. 838–851, 1993.

[78] R. B. Myerson and M. A. Satterthwaite, "Efficient mechanisms for bilateral trading," *Journal of Economic Theory*, vol. 29, no. 2, pp. 265–281, 1983.

[79] P. Lewis, P. Marrow, and X. Yao, "Resource allocation in decentralised computational systems: an evolutionary market-based approach," *Autonomous Agents and Multi-Agent Systems*, vol. 21, no. 2, pp. 143–171, 2010.

[80] A. Chavez, A. Moukas, and P. Maes, "Challenger: A multi-agent system for distributed resource allocation," in *Proceedings of the First International Conference on Autonomous Agents*, 1997.

[81] L. Xiao, Y. Zhu, L. M. Ni, and Z. Xu, "GridIS: An incentive-based Grid Scheduling," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium*, p. 65b, 2005.

[82] K. Kuwabara, T. Ishida, Y. Nishibe, and T. Suda, "An equilibratory market-based approach for distributed resource allocation and its applications to communication network control," in *Market-Based Control: A Paradigm for Distributed Resource Allocation* (S. H. Clearwater, ed.), pp. 53–73, Singapore: World Scientific, 1996.

[83] F. Faniyi and R. Bahsoon, "Self-managing SLA compliance in cloud architectures: a market-based approach," in *Proceedings of the 3rd international ACM SIGSOFT symposium on Architecting Critical Systems*, ISARCS '12, (New York, NY, USA), pp. 61–70, ACM, 2012.

[84] F. Faniyi and R. Bahsoon, "Engineering proprioception in SLA management for cloud architectures," in *Software Architecture (WICSA), 2011 9th Working IEEE/IFIP Conference on*, pp. 336 –340, 2011.

[85] M. Jaeger, G. Rojec-Goldmann, and G. Muhl, "QoS aggregation for web service composition using workflow patterns," in *Enterprise Distributed Object Computing Conference, 2004. EDOC 2004. Proceedings. Eighth IEEE International*, pp. 149 – 159, 2004.

[86] A. Mas-Colell, M. D. Whinston, and J. R. Green, *Micro-Economic Theory.* Oxford: Oxford University Press, 1995.

[87] A. R. Greenwald and J. O. Kephart, "Shopbots and pricebots," *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, vol. 1, pp. 506–511, 1999.

[88] P. R. Lewis, P. Marrow, and X. Yao, "Evolutionary market agents for resource allocation in decentralised systems," in *Parallel Problem Solving From Nature - PPSN X*, vol. 5199 of *Lecture Notes in Computer Science*, pp. 1071–1080, Springer, 2008.

[89] L. Esterle, P. R. Lewis, X. Yao, and B. Rinner, "Socio-Economic Vision Graph Generation and Handover in Distributed Smart Camera Networks," *ACM Transactions on Sensor Networks*, no. 3, 2013.

[90] A. Chandra, K. Nymoen, A. Voldsund, A. Jensenius, K. Glette, and J. Torresen, "Market-based control in interactive music environments," in *Post-proceedings of 9th Int. Symposium on Computer Music Modeling and Retrieval (CMMR 2012)* (S. Y. et al., ed.), LNCS, Springer, 2013 (To appear).

[91] L. Esterle, B. Rinner, P. R. Lewis, and X. Yao, "Improved adaptivity and robustness in decentralised multi-camera networks," in *Proceedings of the Sixth International Conference on Distributed Smart Cameras (ICDSC 2012)*, pp. 1–6, IEEE Press, 2012.