Bandwidth-sharing in LHCONE, an analysis of the problem

Wildish T[1]

[1] Princeton University, New Jersey, USA

E-mail: awildish@princeton.edu

Abstract.

The LHC experiments have traditionally regarded the network as an unreliable resource, one which was expected to be a major source of errors and inefficiency at the time their original computing models were derived. Now, however, the network is seen as much more capable and reliable. Data are routinely transferred with high efficiency and low latency to wherever computing or storage resources are available to use or manage them.

Although there was sufficient network bandwidth for the experiments' needs during Run-1, they cannot rely on ever-increasing bandwidth as a solution to their data-transfer needs in the future. Sooner or later they need to consider the network as a finite resource that they interact with to manage their traffic, in much the same way as they manage their use of disk and CPU resources.

There are several possible ways for the experiments to integrate management of the network in their software stacks, such as the use of virtual circuits with hard bandwidth guarantees or soft real-time flow-control, with somewhat less firm guarantees. Abstractly, these can all be considered as the users (the experiments, or groups of users within the experiment) expressing a request for a given bandwidth between two points for a given duration of time. The network fabric then grants some allocation to each user, dependent on the sum of all requests and the sum of available resources, and attempts to ensure the requirements are met (either deterministically or statistically).

An unresolved question at this time is how to convert the users' requests into an allocation. Simply put, how do we decide what fraction of a network's bandwidth to allocate to each user when the sum of requests exceeds the available bandwidth? The usual problems of any resource-scheduling system arise here, namely how to ensure the resource is used efficiently and fairly, while still satisfying the needs of the users.

Simply fixing quotas on network paths for each user is likely to lead to inefficient use of the network. If one user cannot use their quota for some reason, that bandwidth is lost. Likewise, there is no incentive for the user to be efficient within their quota, they have nothing to gain by using less than their allocation.

As with CPU farms, some sort of dynamic allocation is more likely to be useful. A promising approach for sharing bandwidth at LHCONE is the 'Progressive Second-Price auction', where users are given a budget and are required to bid from that budget for the specific resources they want to reserve. The auction allows users to effectively determine among themselves the degree of sharing they are willing to accept based on the priorities of their traffic and their global share, as represented by their total budget. The network then implements those allocations using whatever mix of technologies is appropriate or available.

This paper describes how the Progressive Second-Price auction works and how it can be applied to LHCONE. Practical questions are addressed, such as how are budgets set, what strategy should users use to manage their budget, how and how often should the auction be run, and how do we ensure that the goals of fairness and efficiency are met.

1. Introduction

Run-1 of LHC was a great success for the LHC experiments. The computing models of the experiments were developed and refined over many years, originally having highly segregated network traffic between sites (e.g. [1]). The Tier-0 would transfer only to Tier-1 sites, which would forward data to Tier-2 sites for analysis. Any Tier-2 that wanted data from another Tier-2 would often have to wait for the data to be staged from the source via Tier-1 intermediaries, since direct Tier-2 to Tier-2 transfers were not allowed.

The network played little direct part in those models. In the days before the WLCG grid[2] was established, the network was considered to be an unreliable component, hard to debug when it failed, and a limiting factor for the overall performance of the system. Instead, before and during Run-1, the network turned out to be one of the most reliable and performant components of the grid. Bandwidths available to the experiments were much higher than originally anticipated, and transfer failures were almost invariable associated with a site rather than with the network fabric, and were therefore easily diagnosed.

As a result, the LHC experiments relaxed their computing models to allow transfers between any pair of sites. Tier-2 to Tier-2 transfers accounted for 1/6 of the total network traffic for CMS, for example. Both ATLAS and CMS have created data-federations, based on xrootd[3], to allow analysis jobs running on any worker node to fall-back to fetching data from any storage element that hosts it, wherever it might be. This extra flexibility in transfers and data-access has had a major impact on the abilities of the experiments to produce physics results, and is expected to be an important component of Run-2.

Despite exceeding expectations, the network is still not yet considered as an active component of the experiments' computing models. It is still used like a utility that is free and infinite, rather than a resource to be scheduled. However, computing models are evolving for Run-2. The experiments face tighter constraints for CPU and storage, and new resource-types (cloud, opportunistic, volunteer) are being integraged into the computing models.

There is now a growing awareness that intelligent use of the network can improve the performance of the entire system[4]. Speed alone is not the only desirable feature, determinism can be as important, if not more. Nor is it necessary that bandwidth guarantees be hard, soft guarantees that are statistically valid can be equally useful.

For example, the ability to place data with deterministic schedules can lead to more effective scheduling of storage and CPU, by making sure the storage is occupied with data only when and where it is needed. Just-in-time data placement becomes possible, and managing CPU-related deadlines becomes more feasible with it.

Networking groups are making good progress towards providing such capabilities from the network level, through projects such as DYNES[5] and NSI[6] which allow the creation of virtual-circuits between sites. The ANSE[7] project has built on their work by integrating the creation of virtual circuits into experiments middleware. As a result, PhEDEx[8], the data-placement management tool of CMS, is now capable of booking and using virtual circuits wherever a suitable interface is available.

N.B. virtual circuits are not the only possible way to provide bandwidth guarantees, other choices are possible, such as the use of multi-path network flows. For convenience, in this paper, the term 'circuit' or 'virtual circuit' will be used to represent any mechanism that can be used to reserve or schedule part of the resources a network can provide.

2. Scheduling the Network

As with most successful grid-related middleware, once you solve the problem of enabling the use of a new technology, you get a new problem: how to prevent abuse or over-subscription of resources. If any user or group can reserve a virtual circuit without constraint then there is nothing to prevent what amounts to denial-of-service attacks on competing experiments, for

example. A complete network-scheduling sytem therefore needs not only to allow capacity to be reserved, but to provide a mechanism to share that capacity fairly among all users.

So what constitutes a good bandwidth-sharing system? Fixed quotas are clearly too inflexible and lead to wasted resources. Instead, a good bandwidth-sharing system should have certain properties:

- Automatic/responsive: Circuits should be set up in a timely manner and without the need for manual intervention.
- Lightweight: Circuits should only be created where they are actually needed. This helps to avoid scaling or reliability issues, but also avoids creating circuits needlessly, where only a single flow is 'competing' for a given network link.
- Elastic: Network shares should be able to grow and shrink over time, following the timescale at which the needs of the users fluctuate. This is probably of the order of an hour, rather than days or minutes.
- Efficient: It should allow the network bandwidth to be fully used at all times.
- Fair: It should not be possible for any user to be starved of resources by other users, either on short timescales (hours) or averaged over longer timescales (days or weeks).

CPU-farms solve very similar problems with their scheduling algorithms. However, these are not directly applicable to sharing network resources. CPU cores are typically allocated in discrete quanta (1, 2, 4, 8 cores...) while network bandwidth is a continuous quantity. CPU cores are also typically interchangeable, you don't care which cores you get as long as you get your quota. Network bandwidth, on the other hand, is not interchangeable. An allocation between sites A and D can depend on the bandwidth available between sites B and C along the path, and if that bandwidth is already taken by another user then it's not available.

Scheduling network resources therefore requires something different from scheduling CPUs, and candidate solutions can be found in the field of auction-theory.

3. Second Price Auctions

A 'classic' second price auction is one in which a single, indivisible item is offered for sale. The winner is the bidder who bids the highest amount for the item, but the price they pay is not the price they bid, it's the price bid by the second-highest bidder.

This sounds like a strange concept, since it might not yield maximal profit, but it's actually how many auctions work. For example, the typical English auction, in which bidders call out progressively higher bids, is a second price auction. Two bidders will bid against each other, each raising the price in small increments until eventually one drops out. The other then wins the auction, but the price they pay has been determined by the bidder who withdrew from the bidding, not by their own evaluation of the item being sold.

Second price auctions have the very desirable property that they converge on a Nash equilibrium, meaning that no single bidder can improve their 'utility' by changing their bid. They also have the property that the optimal strategy for any bidder is to bid their true estimation of the value of the item being sold. This makes them the mechanism of choice for allocating resources in many social or market situations, where maximising profit is not the prime consideration.[1], rather the goal is to achieve a socially-acceptable distribution of resources.

[1] While the English auction may not maximise profit for *rational* bidders, people often get carried away by the moment, and bid beyond their true valuation. Human psychology plays a large part in designing real-world auctions!

4. The Progressive Second Price Auction

The Progressive Second Price Auction (PSP)[9] is an extension of the second price auction for single goods, specifically designed for allocating continuously divisible goods such as network bandwidth. This makes it suitable to form the core of a system that satisfies the requirements outlined earlier. The auction proceeds as follows:

1) The network offers a quantity **Q** of bandwidth on a given network link.
2) Bidders have a given budget, not necessarily the same among bidders.
3) Bidders specify their bids in the form of a (quantity,unit-price) pair, ($\mathbf{q_i}$,$\mathbf{p_i}$).
4) The PSP algorithm calculates the allocation and total cost for each bidder, ($\mathbf{a_i}$,$\mathbf{c_i}$). The calculation is explained below.
5) PSP sends all allocations and costs to all bidders.
6) Bidders revise their bids if they aren't satisfied, and resubmit them.
7) Steps 3-6 are repeated until convergence.

As shown in [9], convergence of the auction is guaranteed if players are rational. Rationality means, in essence, that they are willing to pay a higher unit-price to increase a small allocation than to increase a large one. In other words, the more they have, the less they are willing to pay to get even more. This is simply the market law of 'diminishing marginal returns'.

The PSP algorithm itself calculates allocations first, then the total cost per bidder. The allocations are calculated as follows:

1) Bidders are ranked in decreasing order of the unit-price they have bid.
2) The bidder with the highest unit-price gets his/her allocation first. If there is sufficient bandwidth for them to get the full quantity they requested, the request is granted in full. Otherwise, they get the remaining bandwidth, however much it is.
3) If there is any remaining bandwidth, step 2 is repeated for the next-highest bidder.

This process is illustrated in figure 1

The cost charged to a bidder is based on the 'exclusion compensation' principle, which extends the second-price principle. The fact that a given bidder i has bid may mean that other bidders are not granted the same allocation they would have if bidder i were not participating. Bidder i therefore pays for the amount by which they inconvenience those other bidders. I.e. for all bidders other than i who receive an allocation when i doesn't participate, the change in their allocation with bidder i participating is multiplied by the unit-price they were willing to pay, and the sum is the cost charged to bidder i.

Figure 2 illustrates this.

Also shown in [9] are results from simulations with several tens of bidders all bidding on a single network link. The addition of a small 'reserve price' per unit bandwidth helps to ensure rapid convergence by avoiding bids that change only in the lower decimal places, and still maintains the property of convergence on an ε-Nash equilibrium.

5. The PSP on LHCONE

The PSP as described works well for single network links, with a single divisible item auctioned among several bidders. However, LHCONE is a far more complex beast, as can be seen in figure 3.

Fortunately, the PSP extends naturally to multiple links, where different bidders are interested in different network paths of the network. A full analysis is shown in [10]. The PSP properties are maintained by holding multiple independent auctions for each network link. Bidders have a global budget each, which again can vary from bidder to bidder, from which they

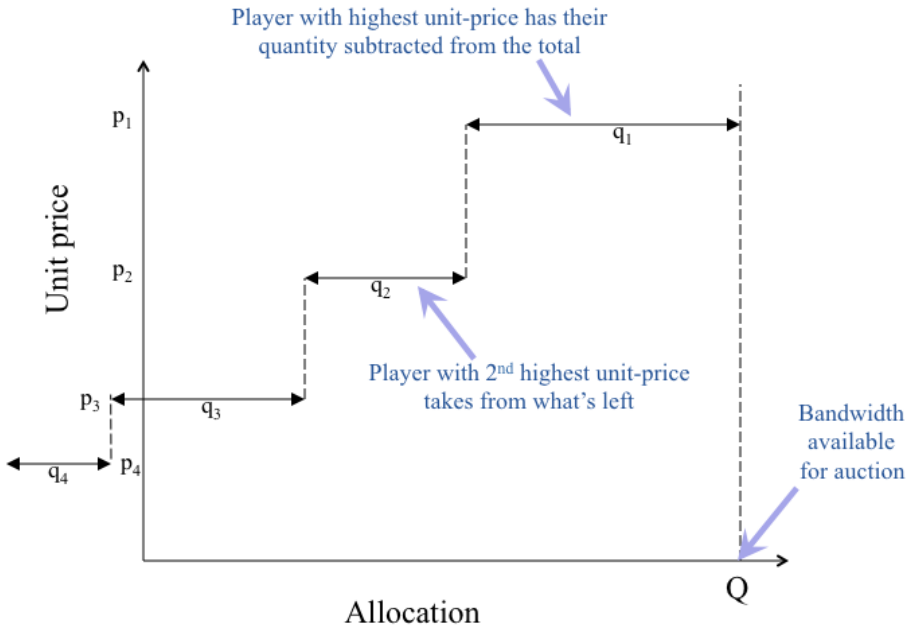

Figure 1. The allocation procedure. The highest unit-price bidder is fully served, the second-highest is served from what's left, and so on. Bidders who bid too low are allocated zero bandwidth if it is all given to higher bidders.

bid on each of the links that they are interested in in parallel. The optimal strategy for a bidder is to bid for the same bandwidth on each link-segment of the network path they are interested in, varying the price they bid at each segment in accordance with the competition for that link. Convergence to an ε-Nash equilibrium is still guaranteed for rational bidders, making the PSP a viable candidate for addressing the problem of sharing bandwidth at LHCONE.

6. Discussion

We have seen that the PSP auction can be used to calculate the allocation of network bandwidth shares in a multi-link network in a way that satisfies the requirements for LHCONE. This is clearly superior to having management-level meetings where people negotiate network shares for the next quarter, for example, but there remain a number of open questions.

The most important question is how to allocate budgets. In a true market setting the budgets are decided by the bidders themselves, the auction is then used to decide on the shares. In a setting such as LHCONE, where experiments or users are not paying with real money for the bandwidth they use, setting budgets becomes more problematic. Essentially, the budget has to represent something of real value to the bidder, so they consider it worth using rationally.

In the case of a single auction, for instance, if the budget is simply some form of fake money which has no value elsewhere, there is no incentive for the bidders to not spend their entire budget. This will lead to bidders bidding for more bandwidth than they actually need, with a consequent waste of resources. In the worst case, a bidder who has no need of bandwidth might bid on links that are valuable to a rival, essentially using their fake budget to launch a denial of service attack. Probably some mechanism is needed to penalize users who bid for resources that they do not use efficiently. If a user cannot make efficient use of their allocation they can

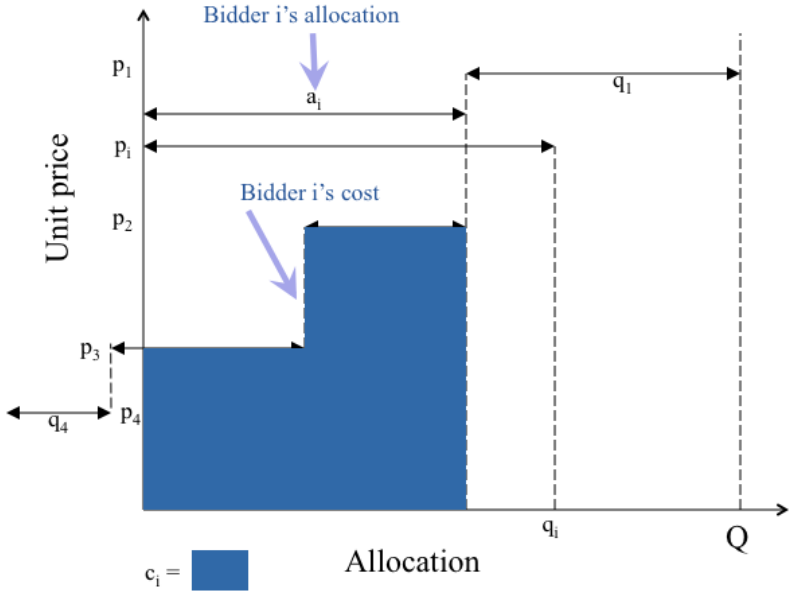

Figure 2. Calculating the cost for a new bid from bidder *i*. Bidder *i*'s allocation means bidders 2 and 3 lose their allocations, so their allocation is multiplied by their unit price, summed, and charged to bidder *i*.

simply trigger a new auction, and reduce their bids accordingly.

A related problem is that of multiple auctions, to handle renegotiation of allocations when a network-users' requirements change. Technically, there's no reason not to re-run the PSP auction whenever requirements change, or on some fixed schedule such as every hour. However, each auction will cost the successful bidders some of their fake budget, eventually leading to budget-depletion. This means budgets need to be replenished at intervals, but how this should happen in a world of fake money is not clear.

One possibility is to simply pay the bidders on a regular basis, according to some policy. However, this can lead to 'market distortions' if a bidder doesn't need their allocation for some period of time (perhaps due to experiment down-time). When the bidder re-enters the market they may have saved enough budget to dominate the bidding for some time, which can starve other bidders. IMposing a cap on the maximum budget may address this. In fact, this regard, allocating budgets bears a lot of resemblance to sharing CPU on a batch farm, and there are probably algorithms used there that can be adapted for this purpose.

Multiple auctions bring another consideration, that of learned strategies. In a single auction the bidders' best strategy is to tell the truth, but when auctions are repeated regularly it is possible that a bidder may learn the true valuation that other bidders place on a given resource. They may then adapt their strategies accordingly, to their advantage and to the detriment of other bidders.

Collaboration is also possible, with users deciding to take turns to bid for a given resource. This is less problematic by definition, since it means that both parties are satisfied with the arrangement, but since it may still disadvantage other users of the network it needs to be considered.

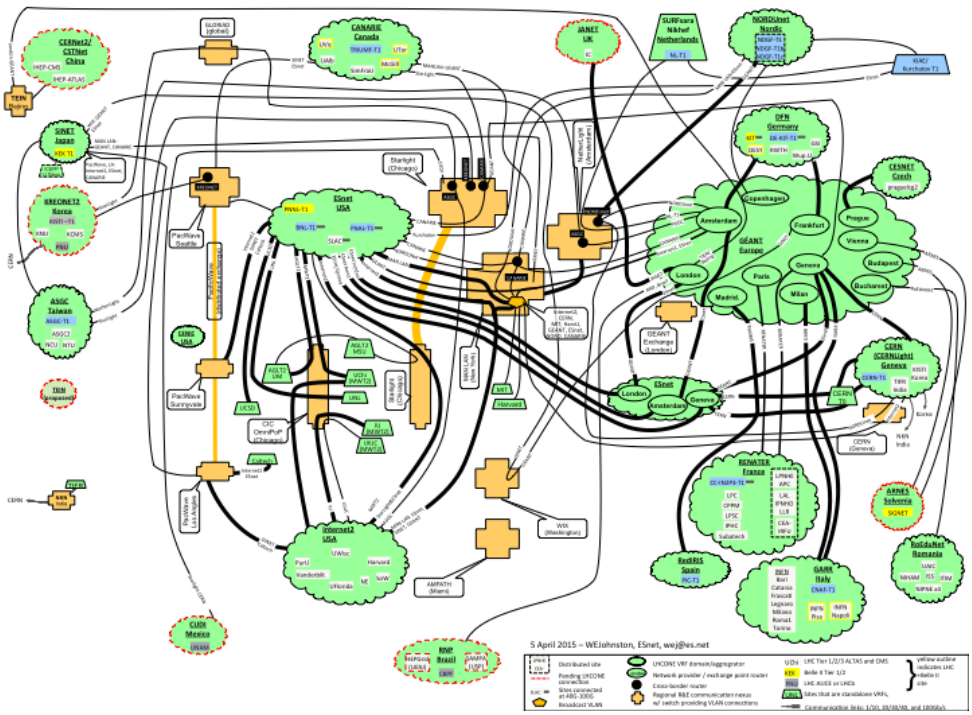

Figure 3. An approximate schematic of the LHCONE network links, showing the complexity of the LHCONE layout.

Another open question is that of how to represent LHCONE. In principle the network can be represented literally, with each link in its proper place. Auctions can be held by each network provider for the links they maintain, with no need of coordination among different network providers. In practice, certain simplifications may be possible, based on how data actually flows through the network and on the actual needs of the users.

Finally, it is unreasonable to expect all network users to bid in an auction every time they interact with the network. There may be low-level users whose needs do not justify such complexity in their computing systems, and the overhead of managing many small circuits is not welcome. One simple solution is to have the network providers auction only a fraction of the available bandwidth, say maybe 80%, leaving the remaining 20% for other users. This will ensure that non-participants in the auction are not starved for resources by those who bid for bandwidth. However, it also means that if the reserved 20% is not actually used at some point then the bidders who do want bandwidth may get less than they require unless they can overflow into the remaining bandwidth. A more effective, but more challenging solution, is to vary the reserved bandwidth fraction and trigger a new auction whenever non-participants present traffic to the network link. In this sense, soft bandwidth guarantees for circuits are more helpful than hard bandwidth guarantees, since they enable the flow of other traffic which can be monitored to detect a significant change in conditions.

7. Conclusions

Market mechanisms, such as the auctions, offer a convenient way of allocating limited resources to users with different and time-dependent needs. A well-designed auction will have desirable properties such as rapid convergence and a best-strategy of truthful bidding for rational bidders.

The PSP in particular is suitable for sharing network bandwidth on non-trivial network

topologies without the need for fixed quotas or negotiations between network providers and network users. Providers simply state their offer, and users then negotiate, by participating in the auction, to decide their shares among themselves. When the auction concludes the network provider has a clear statement of the current state of the users needs across the entire system.

The key to making auctions work is that the monetary unit must have real value to the bidders in order to preserve rational behaviour. A real market situation clearly satisfies that demand, but an environment like LHCONE does not. Other ways must therefore be found to manage budgets and budget allocations in order to preserve the desirable properties of the auction.

The situation is further complicated by the need for repeated auctions to track the changing needs for bandwidth. Fake-money budgets need to be replenished by some suitable algorithm, perhaps inspired by quota-management mechanisms for shared CPU farms. Repeat auctions can also lead to learned strategies, which further complicates the picture.

Finally, the question of how the network is presented for auction needs consideration, especially in an environment where there may be casual or minor users whose needs do not justify the use of circuits and the allocation of budgets.

The two main questions can then be summarised as how to manage budgets and how to present the network for auction. When solved, they can be used with the PSP auction to provide a lightweight, responsive, fair and efficient tool for calculating the needs of the network users at any time. Network providers can then use this information as the basis of a system for providing schedulable use of the network for the data-management systems of present and future experiments.

References

[1] Grandi C, Stickland D and Taylor L 2005 The CMS Computing Model *CERN-LHCC-2004-35/G-083, CMS note 2004-031*

[2] Knoblock J *et al.* 2005 LHC Computing Grid Technical Design Report *CERN-LHCC-2005-024*

[3] XROOTD, http://xrootd.slac.stanford.edu/

[4] Bonacorsi D and Wildish T 2013 Challenging CMS Computing with Network-Aware Systems *J.Phys.Conf.Ser. 513 (2014)*

[5] McKee S et.al 2013 Application Performance Evaluation and Recommendations for the DYNES *J.Phys.Conf.Ser. 513 (2014)*

[6] OGF Network Service Interface *https://www.terena.org/activities/e2e/ws2/slides2/11_NSI_Eduard.pdf*

[7] Integrating Network-Awareness and Network-Management into PhEDEx, *International Symposium on Grids and Clouds 2015*

[8] Egeland R, Wildish T and Metson S 2008 Data transfer infrastructure for CMS data taking, *XII Advanced Computing and Analysis Techniques in Physics Research (Erice, Italy: Proceedings of Science)*

[9] A.A.Lazar and N.Semret. Design, analysis and simulation of the progressive second price auction for network bandwidth sharing. *Technical Report CU/CTR/TR 497-98-21, Columbia University, 1998.*

[10] A.A.Lazar and N.Semret. The Progressive Second Price Auction Mechanism for Network Resource Sharing. *8th International Symposium on Dynamic Games and Applications, Maastricht, The Netherlands, 1998*