The Waggle Dance of Internet Servers (Computer Science, Biology)

The Roman scholar and writer Marcus Terentius Varro (116–27 BC) was already convinced that bees were excellent construction engineers. Inspecting their hexagonal honeycombs, he had a suspicion that this structure was the most efficient in terms of enclosing the greatest amount of space for the storage of honey with the least amount of wax. But as recently demonstrated, bees could also be good computer engineers. Two millennia after Varro, Sunil Nakrani from Oxford University and Craig Tovey from the Georgia Institute of Technology presented a paper on mathematical models of social insects at a conference. Mimicking the behavior of honeybees foraging for nectar, they arrived at an efficient way for the optimal load distribution of Internet servers.

In the 1930s the Nobel Prize–winning zoologist Karl von Frisch discovered that the so-called waggle dance of bees inside the hive provides other bees with information about the distance and quality of a flower patch. Idle bees observe the waggle dance of one of their colleagues and set out on their run. (Since it is dark in the hive, they do not “see” the dance but infer it from changes in air pressure.) They do not communicate with each other before their flights, so none of them know how many will harvest nectar from which patches. And yet honeybees maximize the rate of their nectar collection. Beggarly flower patches are foraged by few bees while profitable and close-by patches receive a plethora of nectar-collecting bees. This happens due to so-called swarm intelligence: Even though each individual bee follows only a small set of instructions, the swarm in its entirety displays near-optimal behavior.

Sunil Nakrani and Craig Tovey were interested in the problems that Internet service providers face. An Internet service provider offers several Internet services—for example, auctions, stock trading, flight reservations. Based on the predicted demand for each service the provider allocates a certain number of servers (a cluster) to each service. The two scientists designed a model of server allocation based on the bees’ behavior and ran simulations to test it. Incoming user requests are distributed into queues for the various services. For each completed request, the provider receives a payment. The number of incoming orders for each service continuously varies, and it would be profitable for underutilized servers to be allocated to overstretched clusters. This entails a cost, however, since a redistributed server must be reconfigured and loaded with the software for the new service. During that time—generally about five minutes—requests and orders cannot be met. If the waiting period (downtime) becomes too long, disappointed customers turn away and potential profits are lost. Hence, to maximize profits, providers must continuously juggle their computers between different applications and adapt to the changing levels of demand.

Traditionally, three algorithms have been used to calculate profitability. First, there is the “omniscient” algorithm, which determines at set time intervals the allocation that would have been optimal for the preceding time slice. Then there is the “greedy” algorithm, which follows the rule of thumb that the levels of demand for all services during a time slice remain unchanged during the next time slice. Finally, there is the “optimal-static” algorithm, which calculates—retrospectively—the optimal, unchanged (static) allocation of servers for the entire time period.

Nakrani and Tovey compared the honeybees’ strategy to these three algorithms. In their model the request queues represent flower patches waiting to be harvested, individual servers represent the foraging bees, and server clusters represent the group of bees harvesting a particular flower patch. The waggle dance becomes a “notice board” in their model. After having fulfilled the requests, servers will, with a certain probability, post a note with the particulars of the served queue. Other servers read the posted notes with a probability that is greater the less profitable the queue is that they presently serve. On the basis of their own recent experience and the posted note, the servers decide—like worker bees observing a waggle dance—whether to switch to a new queue. The costs of a switch from one Web application to another one are comparable to a honeybee’s time investment when observing a waggle dance and switching flower patches.

What the simulations showed is that in terms of profitability the behavior of bees collecting nectar outperforms two of the three algorithms by between 1 and 50 percent. Only the omniscient algorithm yielded higher profits. But this algorithm, which computes the absolutely highest upper limit of profitability, is not practical for two reasons. First, it is unrealistic to assume that future customer behavior is exactly known in advance. Second, the computer resources necessary to calculate the optimal allocation are enormous.

An aside: It was only in 1998 that the American mathematician Tom Hales was able to prove that the six-sided honeycomb (the hexagonal lattice) represents the most efficient partitioning of a plane into equal areas. (See Chapter 9.) But bees are not perfect, despite their apparent ability to approximate the optimal configuration in two dimensions. In three dimensions the much-lauded honeycombs are only close to optimal. The Hungarian mathematician László Fejes Toth designed honeycombs in 1964 that use 0.3 percent less wax than those built by the bees.