Lab 2

Goutham Devarayasamudram and Siddharth Udayappan Chidambaram

April 12, 2013

Introduction

As before, our world is again a grid with each position occupied by a Pig, a stone column or left empty. We have 2 programs, MainGame and Pig. MainGame is in-charge of the overall game mechanics. It handles the leader election, generates values for where and when the bird will land and sends these values to the leader. The Pig program is a Python class for representing a Pig in our system.

Leader Election

We use the bully algorithm for Leader Election. At the start of the game, The MainGame creates all the Pigs and assigns them random positions on the grid and random ports for communication. It then creates the stone columns, ensuring that no two columns are next to one another. After this, a random pig is chosen to initiate the Leader Election. The Random Pig does this by sending an election packet to all the Pigs which have a larger ID than its. The Pigs with larger IDs (if any) will send an OK message back to the Pig that started the election and will in turn send an election packet to other pigs with larger IDs than their own. The election gets over in one of 2 ways. Either a Pig receives the packet and sees that there is no Pig with a larger ID. Or one of Pigs which sent an election packet does not hear back from the other Pigs before the allotted time (and thus becomes the leader). Either way, we will have a leader at the end of this process. Once the leader has been elected, it proceeds to send an I-Won message to all the other pigs. When the Pigs receive an I-Won message, they send their current positions to the leader.

Its possible that the I-Won message of the largest id Pig is in transit but the pig that has started the election times out and elects itself leader. This is handled by our code since every time a pig receives an I-Won message, it checks if the leader id in the message is greater than its current leader ID(every pig stores the leader ID as a class variable) and updates its leader id accordingly.At the beginning of each game, a few of the pigs are randomly selected to go down. There is always a new election at the beginning of each game. If the Pig with largest ID doesn't fail, then it will always get elected as the leader.

Clock Synchronization

We use Lamport's logical clocks for synchronization. Each Pig maintains a logical clock which is initialized to zero. The clock is incremented by a random number (between 1 and 5) when an event happens at the Pig. Whenever a message is sent by a Pig, it attaches its timestamp as part of the message. At the receiver end, the timestamp in the message and the receiver's current timestamp are compared. The receiver then proceeds to update its timestamp to the maximum of these two timestamps plus 1. Note that the receiver Pig's clock always increases in this method. We aren't using clock synchronization for the leader election part since its not required. This doesn't change the correctness of our program since we're not concerned with the time during the election.

Handle Bird Approaching

The MainGame creates random values for the bird position and bird landing times and calls the leader Pig's bird approaching method with these 2 values as arguments. The leader uses the bird landing position to figure out which Pigs are affected. The algorithm the leader to get the affected Pigs goes something like this:

- A pigs-To-Be-Warned-list is maintained by the leader. The list, initially empty, will contain all the Pigs that are/maybe affected by the bird.
- First the leader checks if there is a Pig in the location where the bird is landing (Remember that all the Pigs in our system have told their location to the leader). If there is, this pig (let's call it the target Pig) is added to the list. If there is another Pig to the immediate right of the target Pig, then that Pig also is added to our list. Otherwise, if there is a stone column to the immediate right of the target Pig and there is a Pig to the right of the stone column, then that Pig is added to our list.
- If there is no Pig in the bird landing location but there is a stone column instead, then the leader checks if there is a Pig to the right of the stone column and that Pig is added to our list.
- Note that we are assuming that the Bird is launched from the left end of the Grid and a Pig or a Stone Column will always fall to their right. So Pigs to the left aren't affected.

Once the leader has the pigs-To-Be-Warned-list, it then proceeds to warn each of the Pigs (in the list) of the bird's approach. These warning messages(call it the bird approaching message) are sent with a random network delay (more on this in the next section). Then the leader will sleep for the duration of the bird time (i.e. the time given by the MainGame telling when the bird will land). Once this time has elapsed, the leader sends a bird landed message to all the Pigs in the list. This message is sent **without** any network delay. After the bird landing message has been sent, the control returns to the MainGame. The MainGame then proceeds to wait until the leader has received the status of all the affected Pigs after Bird landing.

At the affected Pigs' end, when the bird approaching message is received, the affected Pig checks if it has already received the bird landing message before this. If it hasn't, it proceeds to take evasive action by changing its position. When the bird landing message is received, it checks if the bird approaching message has already been received. If it has received the warning, then the Pig has already taken evasive action. The Pig then sends a message to the leader saying its not been hit. If the bird landing message was received first, then the Pig has been hit and it notifies the leader that its been hit.

There is a special case when the leader is one of the affected Pigs. 2 possibilities arise:

- The bird is landing exactly in the leader's position. In this case, the leader immediately changes its position. Since the leader is not affected by the bird, no other Pig is affected and hence control is returned to the MainGame with the score 0.
- The bird is landing somewhere to the left of the leader and the leader might get affected by a stone column or another Pig. In this case again, the leader immediately takes evasive action. However, the other affected Pigs are notified using messages as before.

Score Calculation

The leader keeps track of how many Pigs have been hit. As soon as the leader receives the status from all the affected Pigs, the MainGame stops waiting and queries the leader for the score (i.e. the number of pigs hit). Once the MainGame displays the score, one full iteration of our Game is done.

Message Passing

To simulate network delays, we introduce random delays when sending messages from one Pig to another (with the one exception of the bird landing message). This random delay can vary from 1 to the number of pigs in the system. The sending Pig sleeps for the duration of the random delay and then sends the message. We made the sending part threaded since a Pig will have to send a lot of these messages and if not threaded, it will get blocked until it has finished sleeping.

We use 7 message IDs in our program to make the Pigs communicate with each other. Their names and descriptions are given below:

Table 1: Message IDs

Message IDs	Name	Description
1	Bird Approaching	Sent by the leader Pig to the affected Pigs telling them they might get hit by the bird
2	Election Message	Sent by a Pig to its peers with larger IDs than its own notifying them that an election is underway
3	OK Message Receipt	Received by a Pig from one of its peers with a larger id than its own telling that it(larger ID Pig) will take over the election now.
4	I-Won message	Sent by the Pig to all the other Pigs telling them that it won the election. On receiving this message, the Pigs will update their leader variable and proceed to send their positions to the leader.
5	Position Message	Received by the leader from all the other Pigs, the message contains the id and the position of the Pig that sent the message. The leader saves these values in a dictionary.
6	Bird Landing Message	Received by the affected Pigs telling them the bird has landed at their location. The Pigs will then send their status to the leader
7	Status Message	Received by the leader from the affected Pigs telling their status after the Bird landed

Assumptions

- The stone coloumns are decided beforehand and made known to all the Pigs. Hence, when a Pig is elected leader, the other Pigs send only their positions to the leader.
- The bird landing message is sent with no delay.
- We use Direct messages . All Pigs know the ids and addresses of all other Pigs (but only theleader knows all their positions).
- A leader is elected at the start of every game iteration.
- The affected pig decides if it has been hit and sends this result to the leader.
- Logical time stamp only to the bird approaching part of the code.
- The Bird is launched from the left end of the Grid and a Pig or a Stone Column will always fall to their right. So Pigs to the left aren't affected.
- If the leader is one of the affected Pigs, it can immediately change its position.

Improvements

There are a few possible improvements that we can make. We would have liked to try out other election algorithms (like a ring algorithm) and see if they give any improvement over the bully algorithm. Another possible area of improvement is to use vector clocks instead of lamport clocks.