Report 3: Loggy: a logical time logger

Jiayi Feng

September 29, 2021

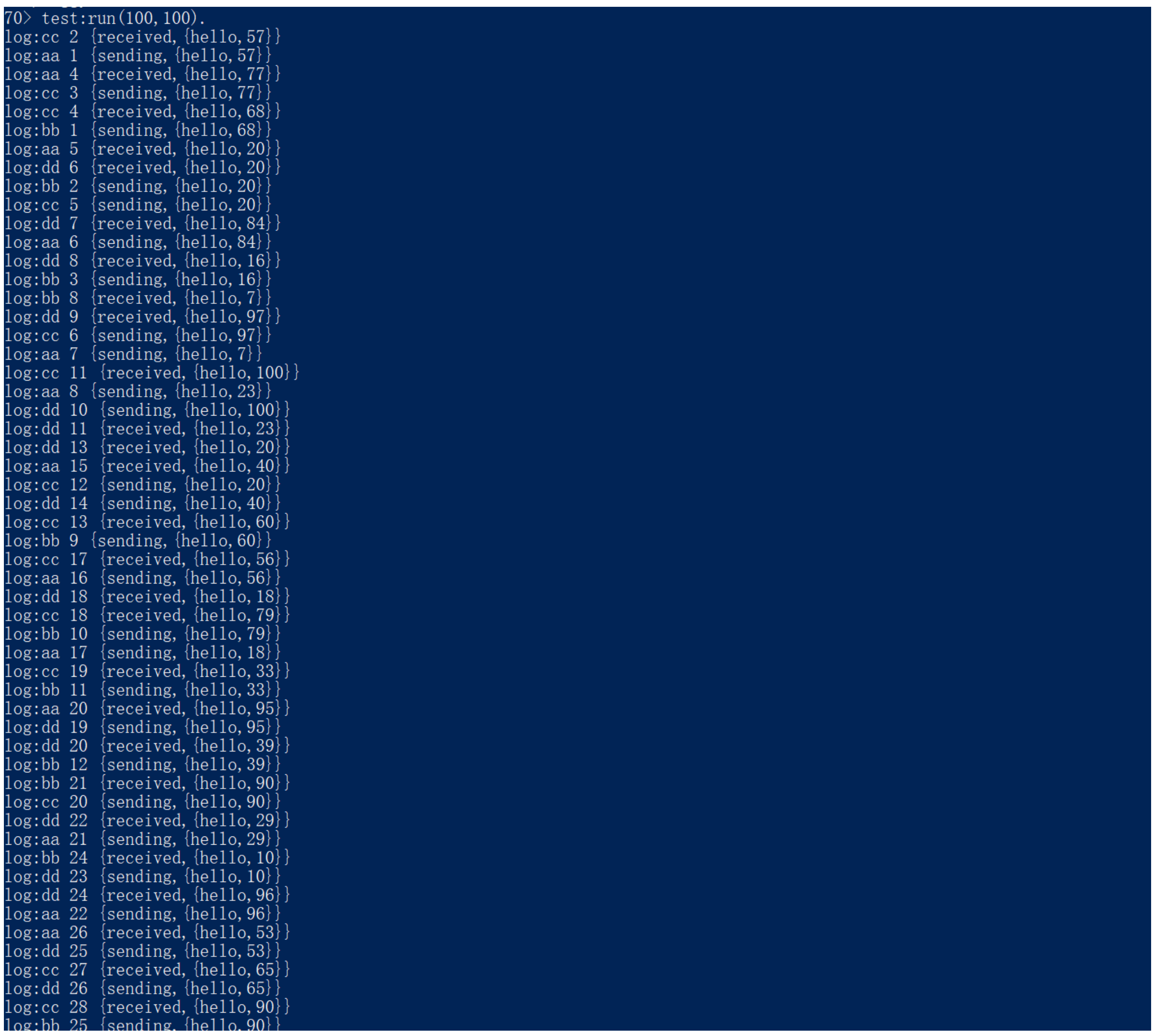
1 Introduction

The task is to implement a logging procedure that receives log events from a set of workers. Both the Lamport time stamp and vector clock will be introduced. The first point will be ordered information, and they should be ascending. The other one is that the sending message is always prior to the received one. The idea is to keep track of the time of the nodes/systems, and to log only messages which are older than the current time's.

2 Main problems and solutions

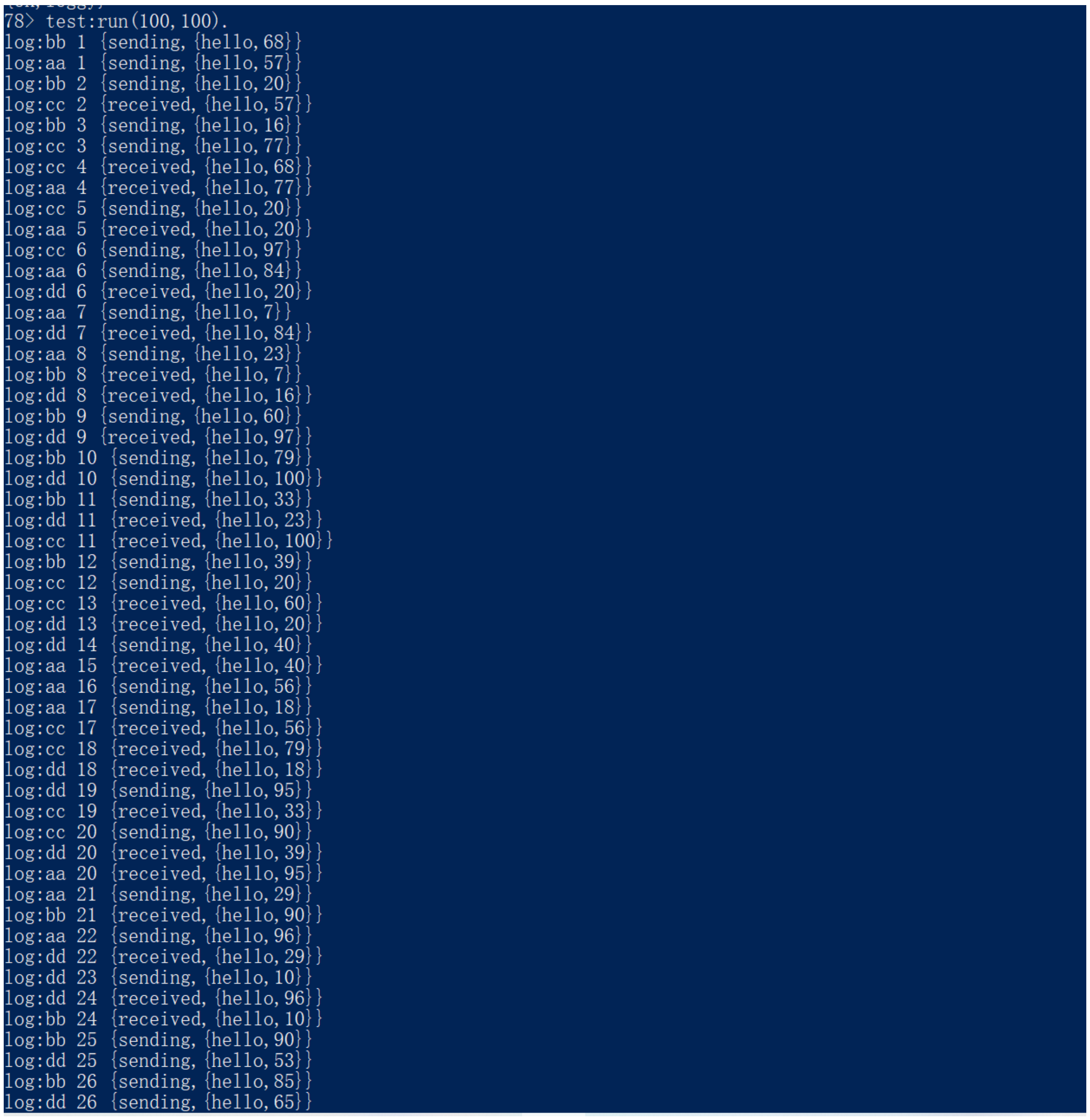
The difference between terms between time and clock.
The log is given with a zero in the tuple, which is given in zero() function. Increment is given by keyfind and keyreplace, where the tuple with targeted name is taken and then increase its time. merge function gives the maximum value of two. leq gives the comparison.
Original module name logger receives errors in compiling, so change it into loggy.

3 Evaluation



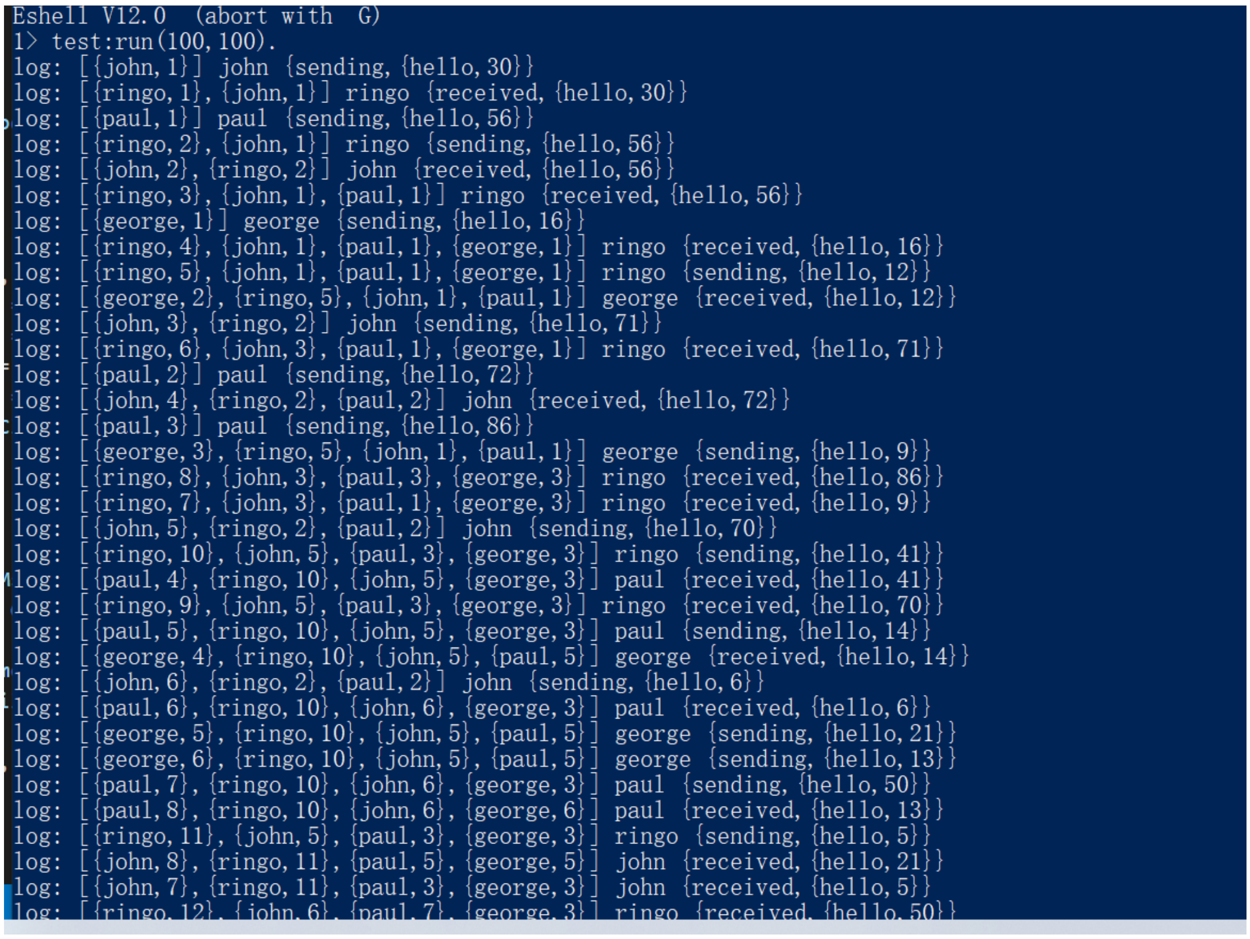
```
70> test:run(100,100).
log:cc 2 {received, {hello,57}}
log:aa 1 {sending, {hello,57}}
log:aa 4 {received, {hello,77}}
log:cc 3 {sending, {hello,77}}
log:cc 4 {received, {hello,68}}
log:bb 1 {sending, {hello,68}}
log:aa 5 {received, {hello,20}}
log:dd 6 {received, {hello,20}}
log:bb 2 {sending, {hello,20}}
log:cc 5 {sending, {hello,20}}
log:dd 7 {received, {hello,84}}
log:aa 6 {sending, {hello,84}}
log:dd 8 {received, {hello,16}}
log:bb 3 {sending, {hello,16}}
log:bb 8 {received, {hello,7}}
log:dd 9 {received, {hello,97}}
log:cc 6 {sending, {hello,97}}
log:aa 7 {sending, {hello,7}}
log:cc 11 {received, {hello,100}}
log:aa 8 {sending, {hello,23}}
log:dd 10 {sending, {hello,100}}
log:dd 11 {received, {hello,23}}
log:dd 13 {received, {hello,20}}
log:aa 15 {received, {hello,40}}
log:cc 12 {sending, {hello,20}}
log:dd 14 {sending, {hello,40}}
log:cc 13 {received, {hello,60}}
log:bb 9 {sending, {hello,60}}
log:cc 17 {received, {hello,56}}
log:aa 16 {sending, {hello,56}}
log:dd 18 {received, {hello,18}}
log:cc 18 {received, {hello,79}}
log:bb 10 {sending, {hello,79}}
log:aa 17 {sending, {hello,18}}
log:cc 19 {received, {hello,33}}
log:bb 11 {sending, {hello,33}}
log:aa 20 {received, {hello,95}}
log:dd 19 {sending, {hello,95}}
log:dd 20 {received, {hello,39}}
log:bb 12 {sending, {hello,39}}
log:bb 21 {received, {hello,90}}
log:cc 20 {sending, {hello,90}}
log:dd 22 {received, {hello,29}}
log:aa 21 {sending, {hello,29}}
log:bb 24 {received, {hello,10}}
log:dd 23 {sending, {hello,10}}
log:dd 24 {received, {hello,96}}
log:aa 22 {sending, {hello,96}}
log:aa 26 {received, {hello,53}}
log:dd 25 {sending, {hello,53}}
log:cc 27 {received, {hello,65}}
log:dd 26 {sending, {hello,65}}
log:cc 28 {received, {hello,90}}
log:bb 25 {sending, {hello,90}}
```

Figure 1: Some random results



```
78> test:run(100,100).
log:bb 1  {sending, {hello,68}}
log:aa 1  {sending, {hello,57}}
log:bb 2  {sending, {hello,20}}
log:cc 2  {received, {hello,57}}
log:bb 3  {sending, {hello,16}}
log:cc 3  {sending, {hello,77}}
log:cc 4  {received, {hello,68}}
log:aa 4  {received, {hello,77}}
log:cc 5  {sending, {hello,20}}
log:aa 5  {received, {hello,20}}
log:cc 6  {sending, {hello,97}}
log:aa 6  {sending, {hello,84}}
log:dd 6  {received, {hello,20}}
log:aa 7  {sending, {hello,7}}
log:dd 7  {received, {hello,84}}
log:aa 8  {sending, {hello,23}}
log:bb 8  {received, {hello,7}}
log:dd 8  {received, {hello,16}}
log:bb 9  {sending, {hello,60}}
log:dd 9  {received, {hello,97}}
log:bb 10  {sending, {hello,79}}
log:dd 10  {sending, {hello,100}}
log:bb 11  {sending, {hello,33}}
log:dd 11  {received, {hello,23}}
log:cc 11  {received, {hello,100}}
log:bb 12  {sending, {hello,39}}
log:cc 12  {sending, {hello,20}}
log:cc 13  {received, {hello,60}}
log:dd 13  {received, {hello,20}}
log:dd 14  {sending, {hello,40}}
log:aa 15  {received, {hello,40}}
log:aa 16  {sending, {hello,56}}
log:aa 17  {sending, {hello,18}}
log:cc 17  {received, {hello,56}}
log:cc 18  {received, {hello,79}}
log:dd 18  {received, {hello,18}}
log:dd 19  {sending, {hello,95}}
log:cc 19  {received, {hello,33}}
log:cc 20  {sending, {hello,90}}
log:dd 20  {received, {hello,39}}
log:aa 20  {received, {hello,95}}
log:aa 21  {sending, {hello,29}}
log:bb 21  {received, {hello,90}}
log:aa 22  {sending, {hello,96}}
log:dd 22  {received, {hello,29}}
log:dd 23  {sending, {hello,10}}
log:dd 24  {received, {hello,96}}
log:bb 24  {received, {hello,10}}
log:bb 25  {sending, {hello,90}}
log:dd 25  {sending, {hello,53}}
log:bb 26  {sending, {hello,85}}
log:dd 26  {sending, {hello,65}}
```

Figure 2: Results with Lamport time stamp



```
Eshell V12.0  (abort with  G)
1> test:run(100,100).
log: [{john,1}] john {sending,{hello,30}}
log: [{ringo,1},{john,1}] ringo {received,{hello,30}}
log: [{paul,1}] paul {sending,{hello,56}}
log: [{ringo,2},{john,1}] ringo {sending,{hello,56}}
log: [{john,2},{ringo,2}] john {received,{hello,56}}
log: [{ringo,3},{john,1},{paul,1}] ringo {received,{hello,56}}
log: [{george,1}] george {sending,{hello,16}}
log: [{ringo,4},{john,1},{paul,1},{george,1}] ringo {received,{hello,16}}
log: [{ringo,5},{john,1},{paul,1},{george,1}] ringo {sending,{hello,12}}
log: [{george,2},{ringo,5},{john,1},{paul,1}] george {received,{hello,12}}
log: [{john,3},{ringo,2}] john {sending,{hello,71}}
log: [{ringo,6},{john,3},{paul,1},{george,1}] ringo {received,{hello,71}}
log: [{paul,2}] paul {sending,{hello,72}}
log: [{john,4},{ringo,2},{paul,2}] john {received,{hello,72}}
log: [{paul,3}] paul {sending,{hello,86}}
log: [{george,3},{ringo,5},{john,1},{paul,1}] george {sending,{hello,9}}
log: [{ringo,8},{john,3},{paul,3},{george,3}] ringo {received,{hello,86}}
log: [{ringo,7},{john,3},{paul,1},{george,3}] ringo {received,{hello,9}}
log: [{john,5},{ringo,2},{paul,2}] john {sending,{hello,70}}
log: [{ringo,10},{john,5},{paul,3},{george,3}] ringo {sending,{hello,41}}
log: [{paul,4},{ringo,10},{john,5},{george,3}] paul {received,{hello,41}}
log: [{ringo,9},{john,5},{paul,3},{george,3}] ringo {received,{hello,70}}
log: [{paul,5},{ringo,10},{john,5},{george,3}] paul {sending,{hello,14}}
log: [{george,4},{ringo,10},{john,5},{paul,5}] george {received,{hello,14}}
log: [{john,6},{ringo,2},{paul,2}] john {sending,{hello,6}}
log: [{paul,6},{ringo,10},{john,6},{george,3}] paul {received,{hello,6}}
log: [{george,5},{ringo,10},{john,5},{paul,5}] george {sending,{hello,21}}
log: [{george,6},{ringo,10},{john,5},{paul,5}] george {sending,{hello,13}}
log: [{paul,7},{ringo,10},{john,6},{george,3}] paul {sending,{hello,50}}
log: [{paul,8},{ringo,10},{john,6},{george,6}] paul {received,{hello,13}}
log: [{ringo,11},{john,5},{paul,3},{george,3}] ringo {sending,{hello,5}}
log: [{john,8},{ringo,11},{paul,5},{george,5}] john {received,{hello,21}}
log: [{john,7},{ringo,11},{paul,3},{george,3}] john {received,{hello,5}}
log: [{ringo,12},{john,6},{paul,7},{george,3}] ringo {received,{hello,50}}
```

Figure 3: Results with vector

4 Conclusions

Both the lamport and vector algorithms are successful. The send information is prior to the receive ones and they are well-ordered.