```bash
#!/bin/bash

# Parse commandline args
while getopts "b:e:" opt; do
  case "$opt" in
    b)  BEGIN_AT=$OPTARG
    ;;
    e)  END_AT=$OPTARG
    ;;
  esac
done

verifyResult () {
  if [ $1 -ne 0 ] ; then
    echo "!!!!!!!!!!!!!!! "$2" !!!!!!!!!!!!!!!!"
    echo "////////////// ERROR !!! FAILED to execute sendTokensFast-Concurrently //////////////////////"
    echo
      exit 1
  fi
}

chaincodeInvoke () {
  peer chaincode invoke --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/zak.codes/orderers/orderer.zak.codes/msp/tlscacerts/tlsca.zak.codes-cert.pem -n chaincode_tokens -c "${PAYLOAD}" -C channel3
    res=$?
  verifyResult $res "Sending tokens fast concurrently"
}

PAYLOAD='{"Args":["sendTokensFast","1","2","1","false"]}'

for (( i = BEGIN_AT; i < END_AT; ++i ))
do
  for (( j = 0; j < 10; ++j ))
  do
    # 1>/dev/null 2>&1
    1>/dev/null 2>&1 peer chaincode invoke --tls true --cafile \
    /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/zak.codes/orderers/orderer.zak.codes/msp/tlscacerts/tlsca.zak.codes-cert.pem \
    -n chaincode_tokens -c "${PAYLOAD}" -C channel3 &
  done
  wait
done

# peer chaincode query --tls true --cafile /opt/gopath/src/github.com/hyperledger/fabric/peer/crypto/ordererOrganizations/zak.codes/orderers/orderer.zak.codes/msp/tlscacerts/tlsca.zak.codes-cert.pem -n chaincode_tokens -c '{"Args":["getAccountTokens", "1"]}' -C channel3
```