Software Speculation

Cody Glover
Pankaj Channe
Software Speculation

• What is it?

• Why is it important?
Speculation Program

speculation

{ parallel code }

verification

{ sequential code }

endSpeculation(check variable)
Speculation Program

- Fork
  ```c
  pid_t childPid = fork();
  if (childPid == 0) {
    // Child code
  } else {
    // Parent code
  }
  ```

- Pipes
  ```c
  #define speculation
  if (childPid == 0) {
    // code
  } else {
    // code
  }
  ```

- Macros
Speculation Program

speculation
  for(int i=0;i<10;i++){
    arrTest[i] = parentValue;
  }
verification
  id="child ";
  for(int i=0;i<10;i++){
    arrTest[i] = childValue;
  }
  printf("finished child\n");
endSpeculation(int,arrTest)
while(true){
  printf("In %s \n", id);
  sleep(1);
}
return(0);
Speculation Program

Speculator (Parent)

speculation
{
   parallel code
}

verification
{
   sequential code
}

dendSpeculation(check variable)

Verifier (Child)

speculation
{
   parallel code
}

verification
{
   sequential code
}

dendSpeculation(check variable)
Speculation Program

```
[johglove@silol ~]$ cd Desktop/
[johglove@silol Desktop]$ ./example.out 2 1
In parent
finished child
done waiting
done reading
checking variable
got check variable: -1
Killing parent
In child
Killed
[johglove@silol Desktop]$ In child
In child
In child
In child
In child
In child
In child
In child
In child
In child
In child
In child
```
Testing Speculation Program

• Two Test Scenarios
  – Scenario 1: Graph Theory Problems
  – Scenario 2: Binary Search Algorithms
Scenario 1

- Test the library for graph theory problem
- Measure no. of times the speculator is killed
- Analyze Effect of graph sizes on the results.
Scenario 2

- Approach
  - Chance of Infeasible Value
- Dynamic Calculation Times

- Measuring
  - Effects of Chance Values
- The Effects of the Calculation Times