

Programmable and Formally Verified Loopy Transformations

Kedar Namjoshi¹ and Nimit Singhanian²

¹ Bell Laboratories, Nokia

² University of Pennsylvania

ExCAPE Review Meeting '16, Philadelphia

Performance!!!

```
for i in people.data.users:  
    response = client.api.statuses.user_timeline.get(screen_name=i.screen_name)  
    print 'Got', len(response.data), 'tweets from', i.screen_name  
    if len(response.data) != 0:  
        ltdate = response.data[0]['created_at']  
        ltdate2 = datetime.strptime(ltdate, '%a %b %d %H:%M:%S +0000 %Y')  
        today = datetime.now()  
        howlong = (today-ltdate2).days  
        if howlong < daywindow:  
            print i.screen_name, 'has tweeted in the past', daywindow,  
                totaltweets += len(response.data)  
            for j in response.data:  
                if j.entities.urls:  
                    for k in j.entities.urls:  
                        newurl = k['expanded_url']  
                        urlset.add((newurl, j.user.screen_name))  
        else:  
            print i.screen_name, 'has not tweeted in the past', daywindow
```

Code



Heterogeneous
Platforms

How to optimize code for heterogeneous
platforms?

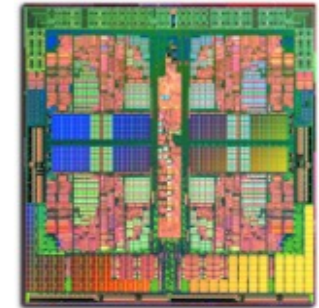
Manual optimization

```
for i in people.data.users:  
    response = client.api.statuses.user_timeline.get(screen_name=i.scre  
    print 'Got', len(response.data), 'tweets from', i.screen_name  
    if len(response.data) != 0:  
        ldate = response.data[0]['created_at']  
        ldate2 = datetime.strptime(ldate, '%a %b %d %H:%M:%S +0000 %Y')  
        today = datetime.now()  
        howlong = (today-ldate2).days  
        if howlong < daywindow:  
            print i.screen_name, 'has tweeted in the past', daywindow,  
                totaltweets += len(response.data)  
            for j in response.data:  
                if j.entities.urls:  
                    for k in j.entities.urls:  
                        newurl = k['expanded_url']  
                        urlset.add((newurl, j.user.screen_name))  
        else:  
            print i.screen_name, 'has not tweeted in the past', daywind
```

Code



OpenMP



CUDA



Scala



Programmable

Verifiable

Tedious and Error Prone

Automatic optimization



```
for i in people.data.users:  
    response = client.api.statuses.user_timeline.get(screen_name=i.scre  
    print 'Got', len(response.data), 'tweets from', i.screen_name  
    if len(response.data) != 0:  
        ldate = response.data[0]['created_at']  
        ldate2 = datetime.strptime(ldate, '%a %b %d %H:%M:%S +0000 %Y'  
        today = datetime.now()  
        howlong = (today-ldate2).days  
        if howlong < daywindow:  
            print i.screen_name, 'has tweeted in the past', daywindow,  
                totaltweets += len(response.data)  
            for j in response.data:  
                if j.entities.urls:  
                    for k in j.entities.urls:  
                        newurl = k['expanded_url']  
                        urlset.add((newurl, j.user.screen_name))  
        else:  
            print i.screen_name, 'has not tweeted in the past', daywind
```

Code





LLVM



 Programmable
 Verifiable

Limited Performance Gains

Other approaches

 Programmable
 Verifiable

OpenACC



CHiLL

POET

X Language

Orio

...

 Programmable
 Verifiable

URUK

PLUTO

...

Loopy

```
for i in people.data.users:  
    response = client.v1.statuses.user_timeline.get(screen_name=i.screen_name)  
    print 'Got', len(response.data), 'tweets from', i.screen_name  
    if len(response.data) != 0:  
        tdate = response.data[0]['created_at']  
        tdate2 = datetime.strptime(tdate, '%a %b %d %H:%M:%S +0000 %Y')  
        today = datetime.now()  
        howlong = (today - tdate2).days  
        if howlong <= daywindow:  
            print i.screen_name, 'has tweeted in the past', daywindow,  
                totaltweets = len(response.data)  
            for j in response.data:  
                if j.entities.urls:  
                    for k in j.entities.urls:  
                        newurl = k['expanded_url']  
                        urlset.add((newurl, j.user.screen_name))  
        else:  
            print i.screen_name, 'has not tweeted in the past', daywindow
```

Code

+

```
offlineRight, ( i1, j, k1 ) =>  
offlineNorm, ( i1, j, k1 ) => i1  
realignNorm, Right, 33  
realignLeft, Norm, 33  
l = liftLeft, 33  
offline1, ( i1, j, k1 ) => i1, k  
offline2, ( i1, j, k1 ) => i11,
```

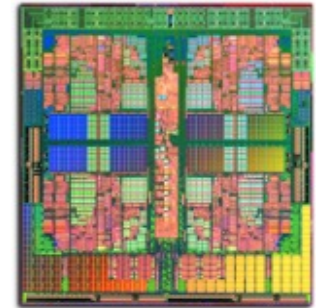
```
offlineRight, ( i1, j, k1 ) =>  
offlineNorm, ( i1, j, k1 ) => i1  
realignNorm, Right, 33  
realignLeft, Norm, 33  
l = liftLeft, 33  
offline1, ( i1, j, k1 ) => i1, k  
offline2, ( i1, j, k1 ) => i11,
```

```
offlineRight, ( i1, j, k1 ) =>  
offlineNorm, ( i1, j, k1 ) => i1  
realignNorm, Right, 33  
realignLeft, Norm, 33  
l = liftLeft, 33  
offline1, ( i1, j, k1 ) => i1, k  
offline2, ( i1, j, k1 ) => i11,
```

Trans. Scripts



Loopy



- ✓ Programmable
- ✓ Verifiable

Programmer Insight can be powerful

Loopy: design

Loop transformation framework

```
for (i = 0; i < N; i++){  
  for (j = 0; j < N; j++){  
    {  
      Init: C[i][j] = 0;  
    }  
    for (k = 0; k < N; k++){  
      Mult: C[i][j] += A[i][k]*B[k][j];  
    }  
  }  
}
```

Code

realign (Init, Mult, 0)

affine (Mult, {[i, j, k] -> [i, k, j]})

affine (Mult, {[i, j, k] -> [i1, j1, k1, i2, j2, k2]:
i1 = [i/64], j1 = [j/64], k1 = [k/64],
i2 = i%64, j2 = j%64, k2 = k%64})

Transformation Script

Basic operation

Component Label

Composition of basic operations

Loopy: design

Loop transformation framework

```
for (i = 0; i < N; i++){  
  for (j = 0; j < N; j++){  
    {  
      Init: C[i][j] = 0;  
    }  
    for (k = 0; k < N; k++){  
      Mult: C[i][j] += A[i][k]*B[k][j];  
    }  
  }  
}
```

`realign (Init, Mult, 0)`

`affine (Mult, {[i, j, k] -> [i, k, j]})`

`affine (Mult, {[i, j, k] -> [i1, j1, k1, i2, j2, k2]:
 i1 = [i/64], j1 = [j/64], k1 = [k/64],
 i2 = i%64, j2 = j%64, k2 = k%64})`

Loop Splitting

Loopy: design

Loop transformation framework

```
for (i = 0; i < N; i++){
  for (j = 0; j < N; j++){
    Init: C[i][j] = 0;
  }
}
for (i = 0; i < N; i++){
  for (j = 0; j < N; j++){
    for (k = 0; k < N; k++){
      Mult: C[i][j] += A[i][k]*B[k][j];
    }
  }
}
```

`realign (Init, Mult, 0)`

`affine (Mult, {[i, j, k] -> [i, k, j]})`

`affine (Mult, {[i, j, k] -> [i1, j1, k1, i2, j2, k2]:
i1 = [i/64], j1 = [j/64], k1 = [k/64],
i2 = i%64, j2 = j%64, k2 = k%64})`

Loop Interchange

Loopy: design

Loop transformation framework

```
for (i = 0; i < N; i++){
  for (j = 0; j < N; j++){
    Init: C[i][j] = 0;
  }
}
for (i = 0; i < N; i++){
  for (k = 0; k < N; k++){
    for (j = 0; j < N; j++){
      Mult: C[i][j] += A[i][k]*B[k][j];
    }
  }
}
```

`realign (Init, Mult, 0)`

`affine (Mult, {[i, j, k] -> [i, k, j]})`

`affine (Mult, {[i, j, k] -> [i1, j1, k1, i2, j2, k2]:
i1 = [i/64], j1 = [j/64], k1 = [k/64],
i2 = i%64, j2 = j%64, k2 = k%64})`

Loop Tiling

Loopy: design

Loop transformation framework

```
for (i = 0; i < N; i++){
  for (j = 0; j < N; j++){
    Init: C[i][j] = 0;
  }
}
for (i1 = 0; i1 < N/64; i1++){
  for (k1 = 0; k1 < N/64; k1++){
    for (j1 = 0; j1 < N/64; j1++){
      for (i2 = 0; i2 < min(64, N-i1*64); i2++){
        for (k2 = 0; k2 < min(64, N-k1*64); k2++){
          for (j2 = 0; j2 < min(64, N-j1*64); j2++){
            Mult: C[i1*64 + i2][j1*64 + j2] += A[i1*64
+ i2][k1*64 + k2]*B[k1*64 + k2][j1*64 + j2];
          }
        }
      }
    }
  }
}
```

`realign (Init, Mult, 0)`

`affine (Mult, {[i, j, k] -> [i, k, j]})`

`affine (Mult, {[i, j, k] -> [i1, j1, k1, i2, j2, k2]:
i1 = [i/64], j1 = [j/64], k1 = [k/64],
i2 = i%64, j2 = j%64, k2 = k%64})`

Loopy: design

Loop transformation framework

```
for (i = 0; i < N; i++){
  for (j = 0; j < N; j++){
    Init: C[i][j] = 0;
  }
}
for (i1 = 0; i1 < N/64; i1++){
  for (k1 = 0; k1 < N/64; k1++){
    for (j1 = 0; j1 < N/64; j1++){
      for (i2 = 0; i2 < min(64, N-i1*64); i2++){
        for (k2 = 0; k2 < min(64, N-k1*64); k2++){
          for (j2 = 0; j2 < min(64, N-j1*64); j2++){
            Mult: C[i1*64 + i2][j1*64 + j2] += A[i1*64
+ i2][k1*64 + k2]*B[k1*64 + k2][j1*64 + j2];
          }
        }
      }
    }
  }
}
```

`realign (Init, Mult, 0)`

`affine (Mult, {[i, j, k] -> [i, k, j]})`

`affine (Mult, {[i, j, k] -> [i1, j1, k1, i2, j2, k2]:
i1 = [i/64], j1 = [j/64], k1 = [k/64],
i2 = i%64, j2 = j%64, k2 = k%64})`



Check correctness

Loopy: architecture

```
for i in people.data.users:  
    response = client.api.statuses.user_timeline.get(screen_name=i.scre  
    print 'Got', len(response.data), 'tweets from', i.screen_name  
    if len(response.data) == 0:  
        tdate = response.data[0]['created_at']  
        tdate2 = datetime.strptime(tdate, '%a %b %d %H:%M:%S +0000 %Y')  
        today = datetime.now()  
        howlong = (today - tdate2).days  
        if howlong > daywindow:  
            print i.screen_name, 'has tweeted in the past', daywindow,  
            totaltweets = len(response.data)  
            for j in response.data:  
                if j.entities.urls:  
                    for k in j.entities.urls:  
                        newurl = k['expanded_url']  
                        urlset.add(newurl, j.user.screen_name)  
            else:  
                print i.screen_name, 'has not tweeted in the past', daywind
```

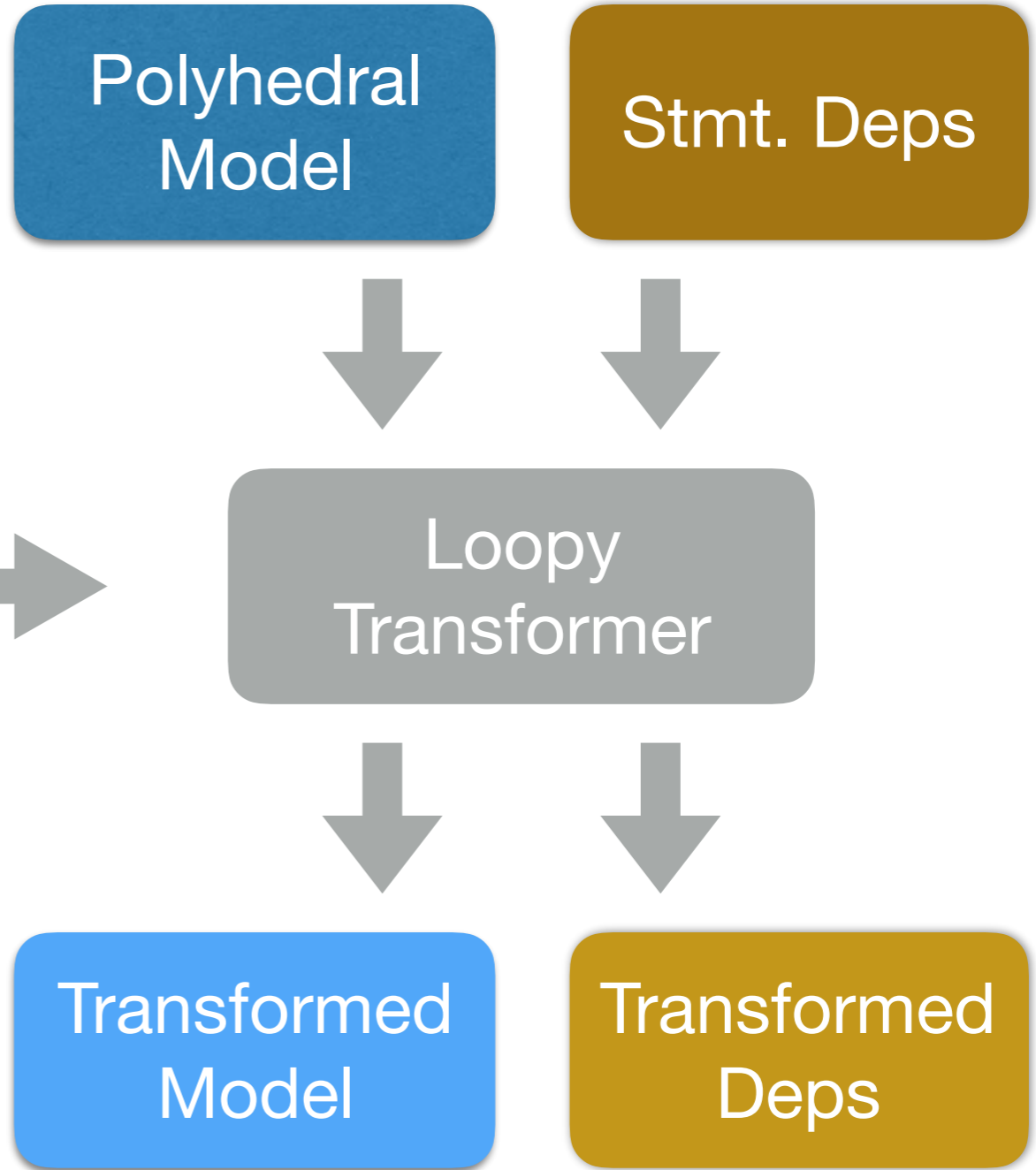
Code

```
offsetRight, ( i i, j, k) ->  
offsetNorm, ( i i, j, k) -> i  
realignNorm, Right, 33  
realignLeft, Norm, 33  
i = littleLeft, 33  
offsetLeft, ( i i, j, k) -> i, k  
offsetLeft, ( i i, j, k) -> i i,
```

Trans. Script

```
for i in people.data.users:  
    response = client.api.statuses.user_timeline.get(screen_name=i.scre  
    print 'Got', len(response.data), 'tweets from', i.screen_name  
    if len(response.data) == 0:  
        tdate = response.data[0]['created_at']  
        tdate2 = datetime.strptime(tdate, '%a %b %d %H:%M:%S +0000 %Y')  
        today = datetime.now()  
        howlong = (today - tdate2).days  
        if howlong > daywindow:  
            print i.screen_name, 'has tweeted in the past', daywindow,  
            totaltweets = len(response.data)  
            for j in response.data:  
                if j.entities.urls:  
                    for k in j.entities.urls:  
                        newurl = k['expanded_url']  
                        urlset.add(newurl, j.user.screen_name)  
            else:  
                print i.screen_name, 'has not tweeted in the past', daywind
```

Opt. Code



Evaluation

- Implemented in LLVM using Polly
- Evaluated on Polybench benchmark suite
 - A suite of 30 diverse programs
- Compared with ICC and PLUTO on single-threaded performance
- Average speed up: **1.95x** (ICC), **2.18x** (PLUTO)
 - Avg # of basic operations: **2.4**

Conclusion

- Loopy balances Programmability and Verifiability
- Clean programmer abstraction to sophisticated technology
- Trust but verify!

Thank you!