
On Control Signals for Multi-dimensional time

DaeGon Kim

Gautam

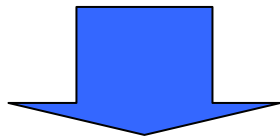
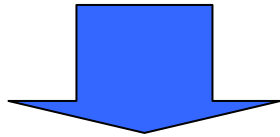
Sanjay Rajopadhye

Colorado State University

**This work is supported by National Science Foundation
under the grant EI-030614:HiPHiPECS**

Motivation & Context

Affine Control Loops



Application-specific hardware

Compute-intensive kernels
Data-intensive kernels
Scientific computation

Interconnection

Control mechanism

- when
- which computation
- distribution

High Performance - Parallelism
Low power consumption

Input specification : FGP ACLs

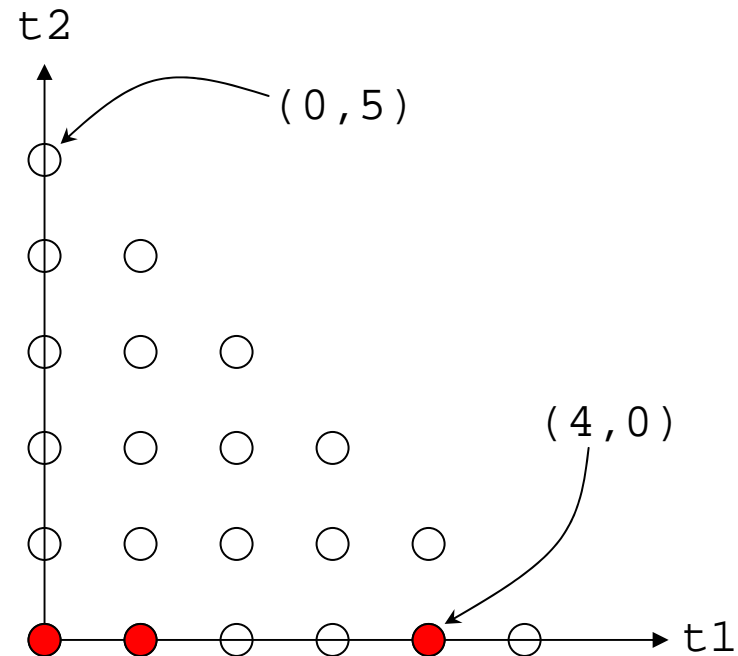
ACL

```
for i=0 to N
  for j=0 to N-i
    for k=0 to N-i-j
      A[i,j,k] = ...
```

Schedule

```
for t1=0 to N
  for t2=0 to N-t1
    for k=0 to N-t1-t2
      A[t1,t2,k] = ...
```

- Data dependence analysis
- Affine transformations



- Time is multi-dimensional !!
- Time is represented by a vector

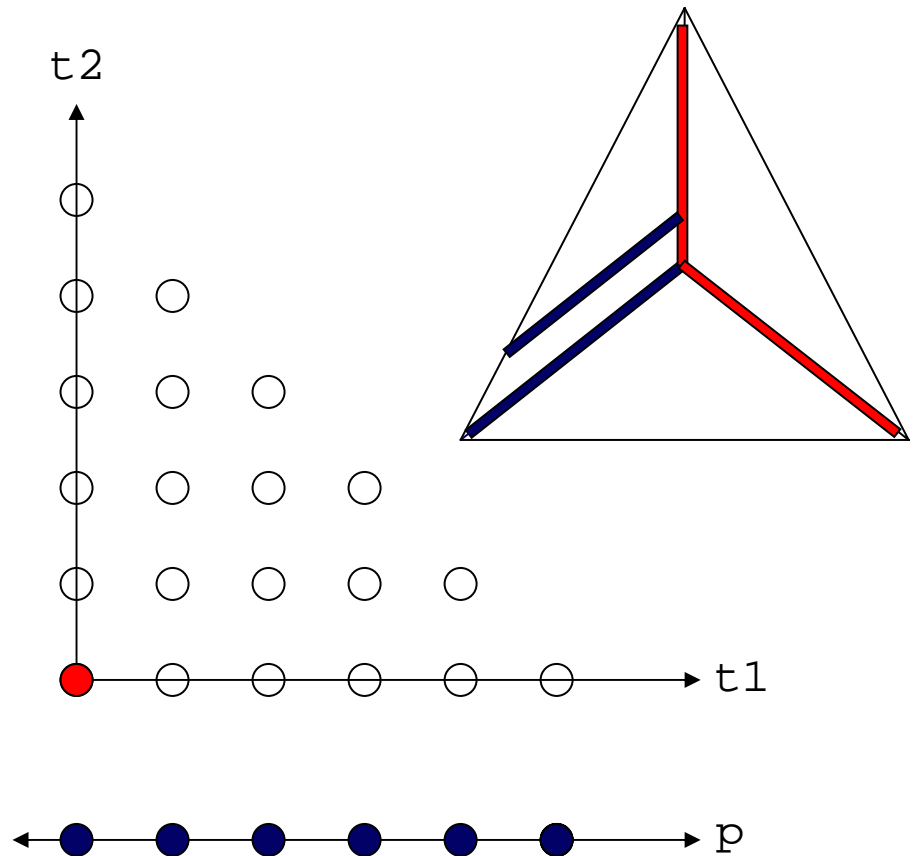
Input specification : FGP ACLs

ACL

```
for i=0 to N
  for j=0 to N-i
    for k=0 to N-i-j
      A[i,j,k] = ...
```

Processor Allocation

```
for t1=0 to N
  for t2=0 to N-t1
    forall p=0 to N-t1-t2
      A[t1,t2,p] = ...
```



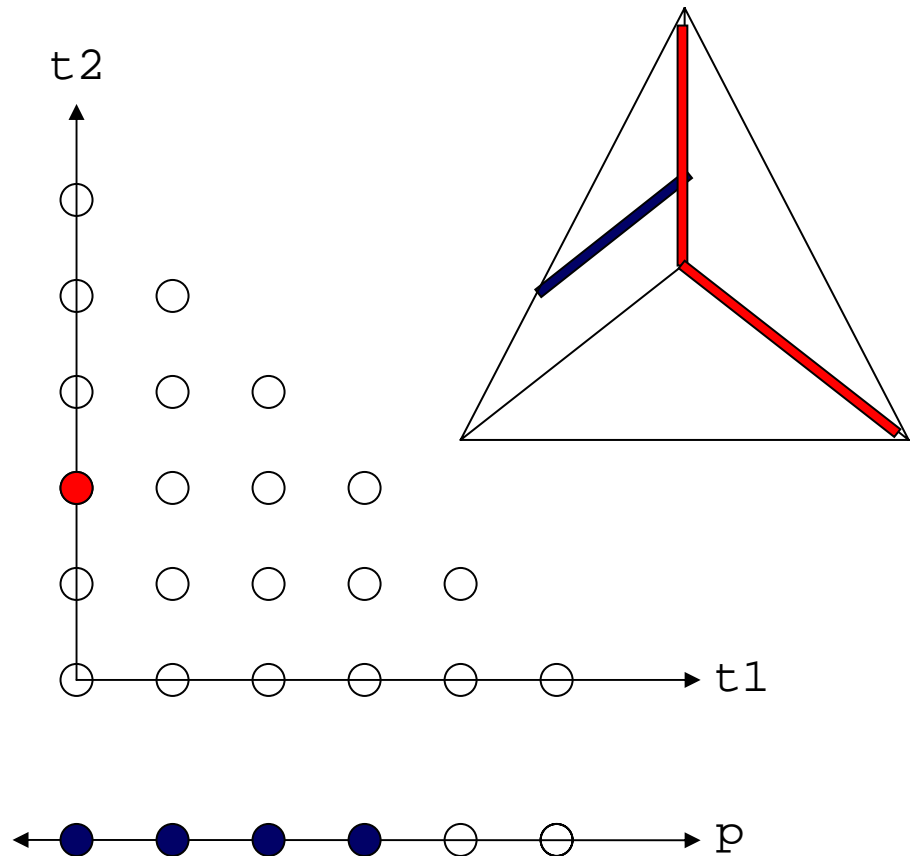
Input specification : FGP ACLs

ACL

```
for i=0 to N
  for j=0 to N-i
    for k=0 to N-i-j
      A[i,j,k] = ...
```

Processor Allocation

```
for t1=0 to N
  for t2=0 to N-t1
    forall p=0 to N-t1-t2
      A[t1,t2,p] = ...
```



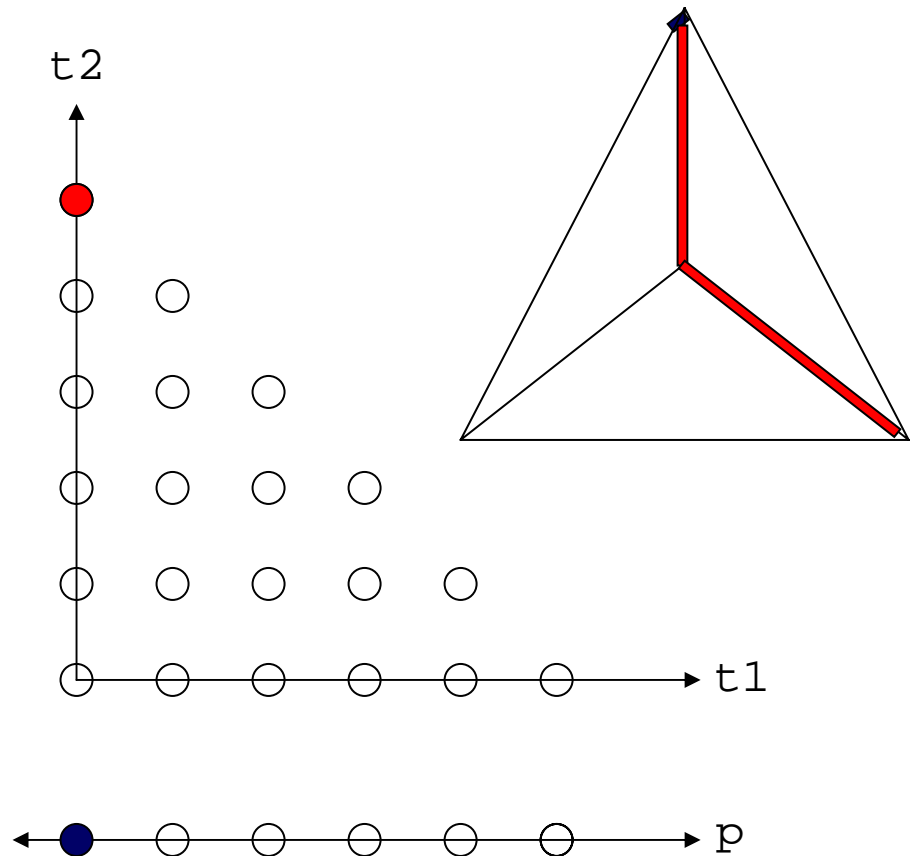
Input specification : FGP ACLs

ACL

```
for i=0 to N
  for j=0 to N-i
    for k=0 to N-i-j
      A[i,j,k] = ...
```

Processor Allocation

```
for t1=0 to N
  for t2=0 to N-t1
    forall p=0 to N-t1-t2
      A[t1,t2,p] = ...
```



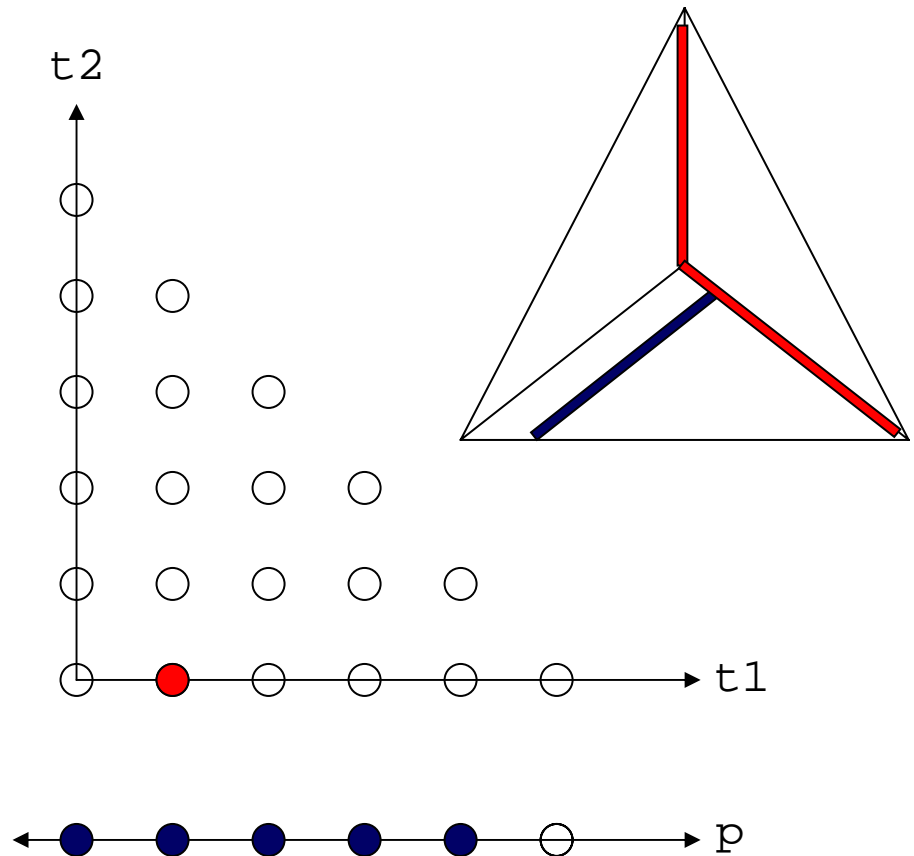
Input specification : FGP ACLs

ACL

```
for i=0 to N
  for j=0 to N-i
    for k=0 to N-i-j
      A[i,j,k] = ...
```

Processor Allocation

```
for t1=0 to N
  for t2=0 to N-t1
    forall p=0 to N-t1-t2
      A[t1,t2,p] = ...
```



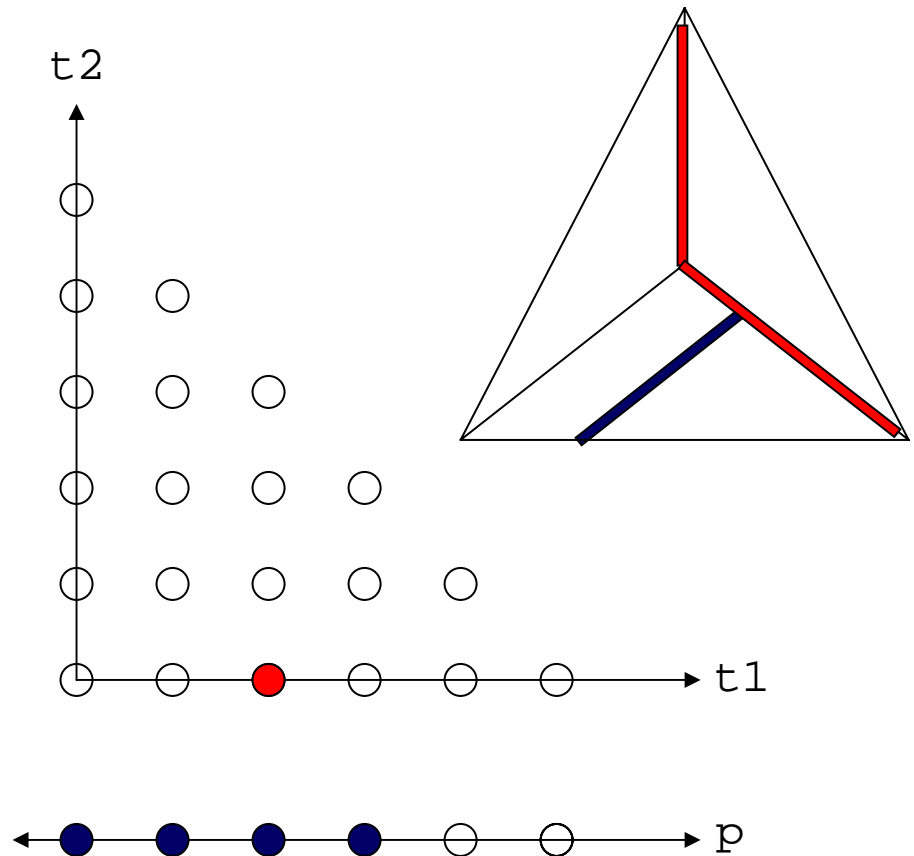
Input specification : FGP ACLs

ACL

```
for i=0 to N
  for j=0 to N-i
    for k=0 to N-i-j
      A[i,j,k] = ...
```

Processor Allocation

```
for t1=0 to N
  for t2=0 to N-t1
    forall p=0 to N-t1-t2
      A[t1,t2,p] = ...
```



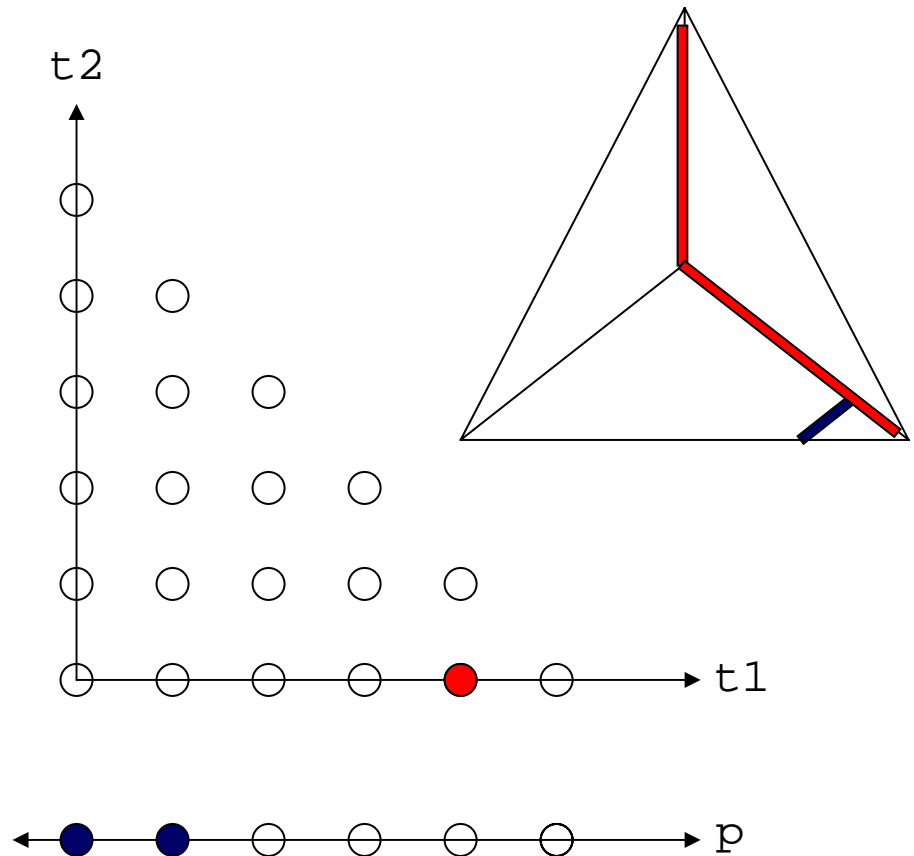
Input specification : FGP ACLs

ACL

```
for i=0 to N
  for j=0 to N-i
    for k=0 to N-i-j
      A[i,j,k] = ...
```

Processor Allocation

```
for t1=0 to N
  for t2=0 to N-t1
    forall p=0 to N-t1-t2
      A[t1,t2,p] = ...
```



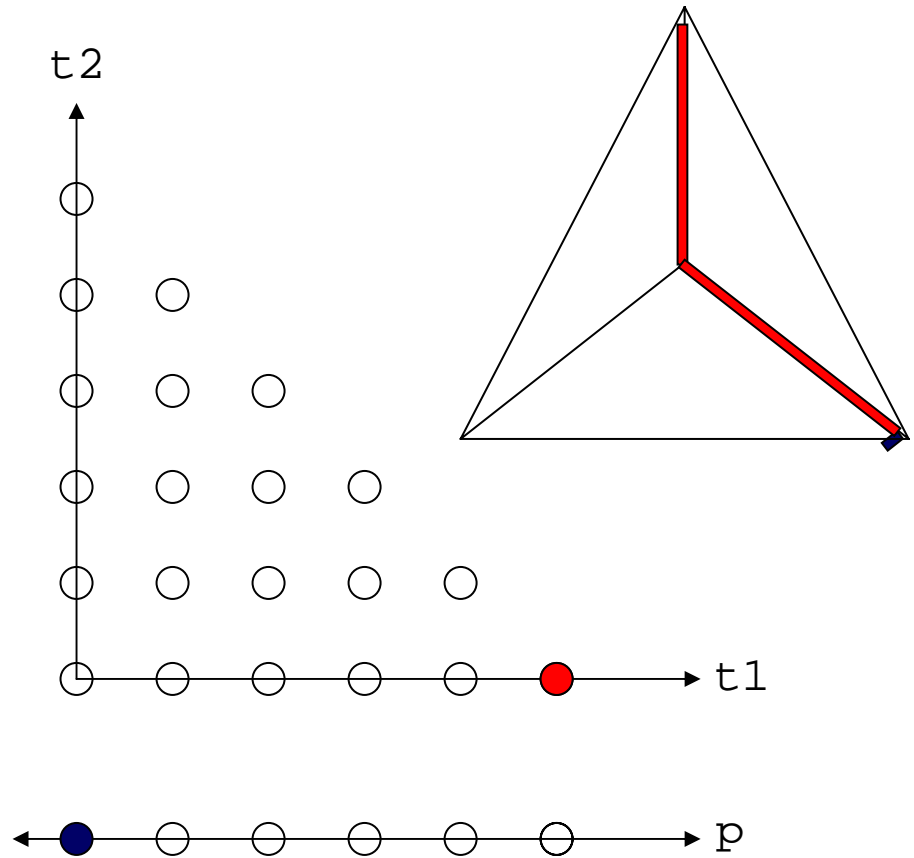
Input specification : FGP ACLs

ACL

```
for i=0 to N
  for j=0 to N-i
    for k=0 to N-i-j
      A[i,j,k] = ...
```

Processor Allocation

```
for t1=0 to N
  for t2=0 to N-t1
    forall p=0 to N-t1-t2
      A[t1,t2,p] = ...
```

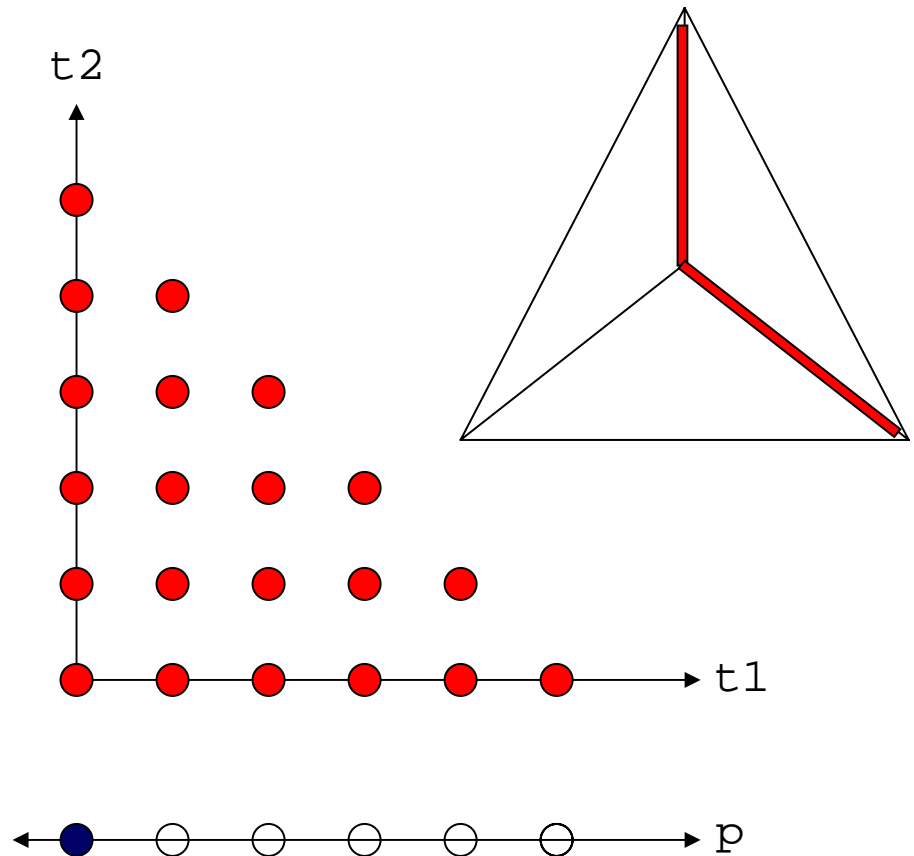


- Active processors vary depending on time !!

Local time domain - control signals

FGP ACLs

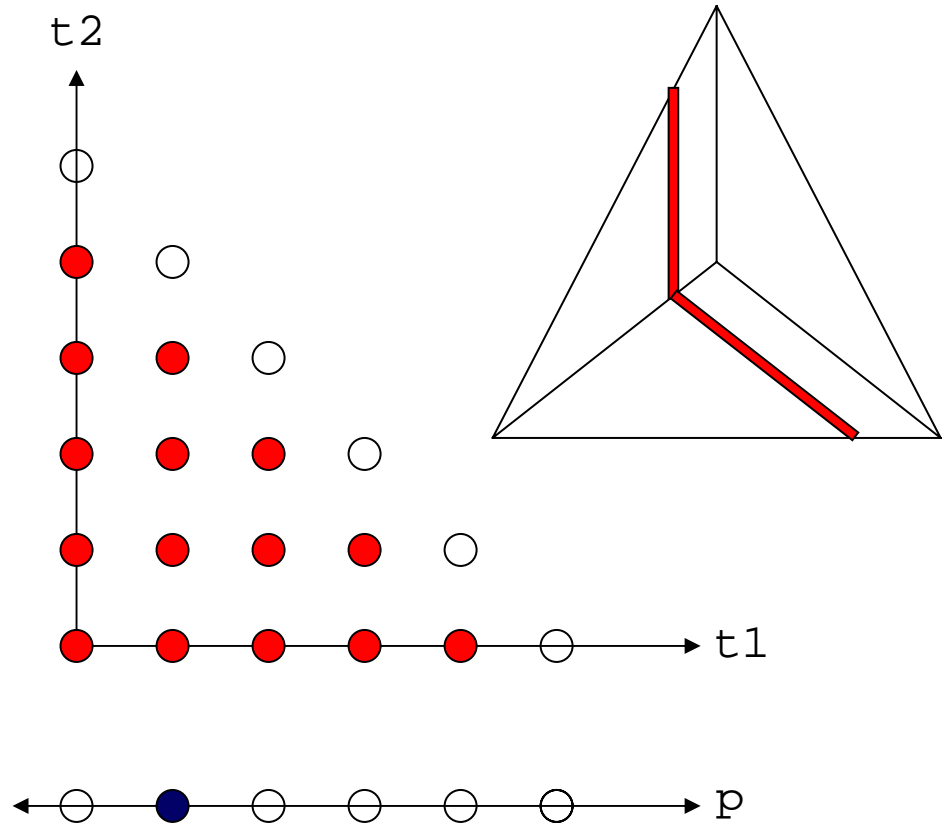
```
for t1=0 to N
  for t2=0 to N-t1
    forall p=0 to N-t1-t2
      A[t1,t2,p] = ...
```



Local time domain - control signals

FGP ACLs

```
for t1=0 to N
  for t2=0 to N-t1
    forall p=0 to N-t1-t2
      A[t1,t2,p] = ...
```



Problem & Main Contribution

- Find sets of time when control signals are needed.
 - Multidimensional time/processors
 - Goal : simple processors (no ISA)
- Characterize time instants when control signals are needed
- Algorithms for computing the set of such time instants

Problem & Main Contribution

- Find sets of time when control signals are needed.
 - Multidimensional time/processors
 - Goal : simple processors (no ISA)
- Characterize time instants when control signals are needed
- Algorithms for computing the set of such time instants

Resume/Suspend signals

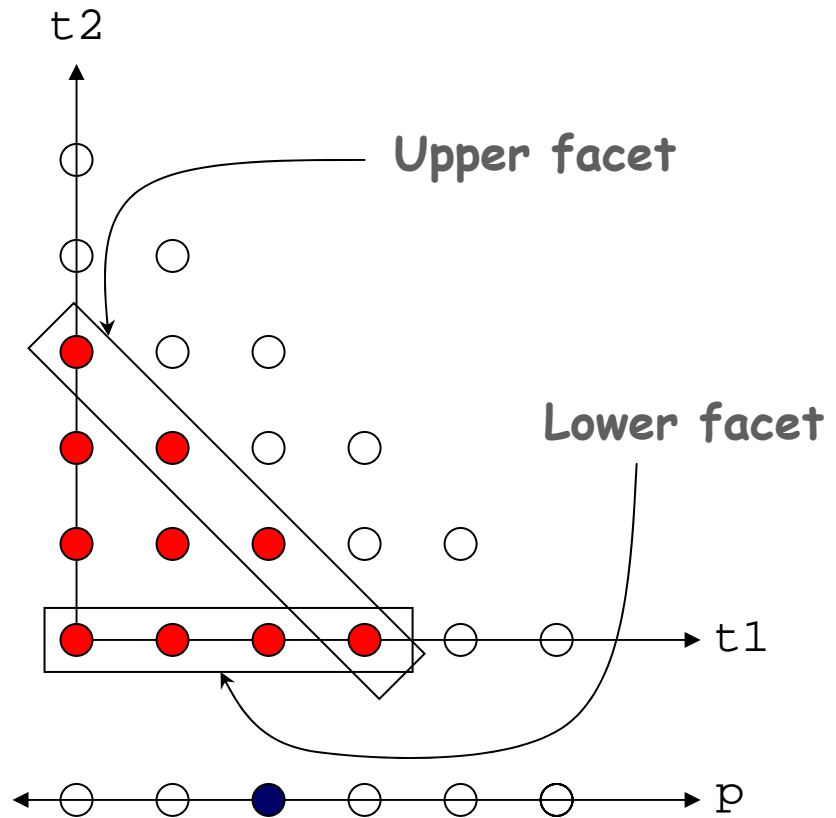
FGP ACLs

```
for t1=0 to N
  for t2=0 to N-t1
    forall p=0 to N-t1-t2
      A[t1,t2,p] = ...
```

Control signals

Resume signal: active time instant
whose previous time is not active

Suspend signal: active time instants
whose next time is not active



First Characteristics

- Conditions for control signals
 - A time instant requiring a resume signal
 - Lower facet of local time domain
 - A time instant requiring a suspend signal
 - Upper facet of local time domain

However, this is not necessary condition.

Can we optimize further?

Another example

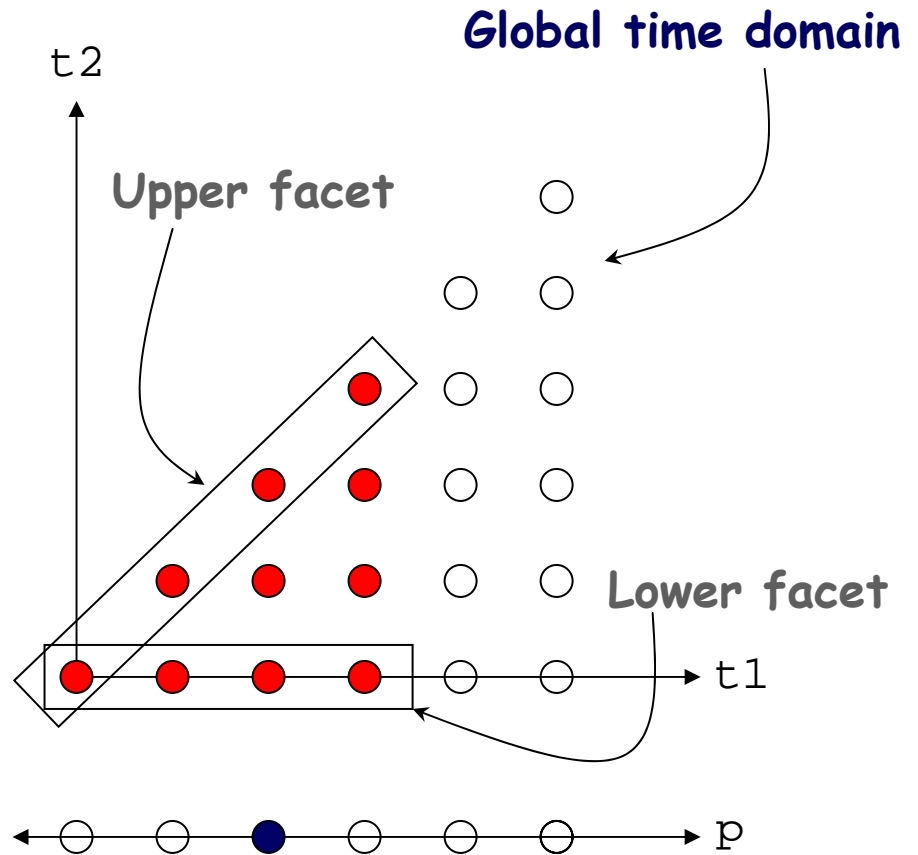
FGP ACLs

```
for t1=0 to N
  for t2=0 to t1
    forall p=0 to N-t1
      A[t1,t2,p] = ...
```

Control signals

Resume signal: active time instant
whose **previous** time is not active

Suspend signal: active time instants
whose **next** time is not active



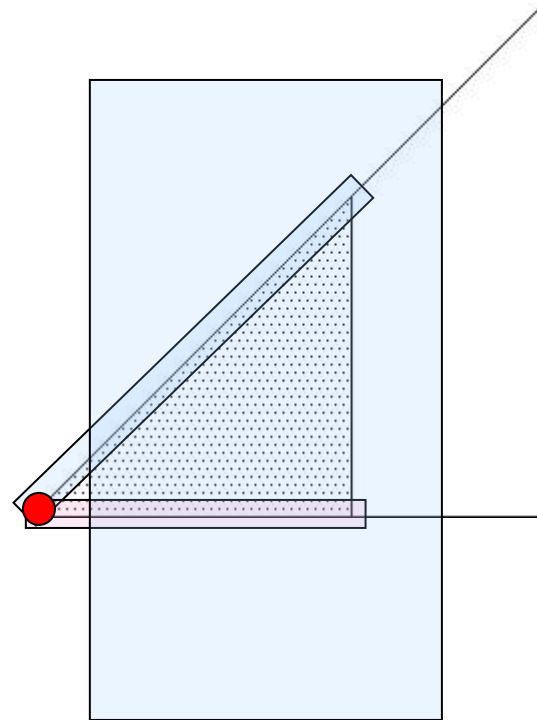
Algorithm

Resume Signals

- Lower facet of LT, not GT
- Lower facet of LT and GT

Suspend Signals

- Upper facet of LT, not GT
- Upper facet of LT and GT



Algorithm

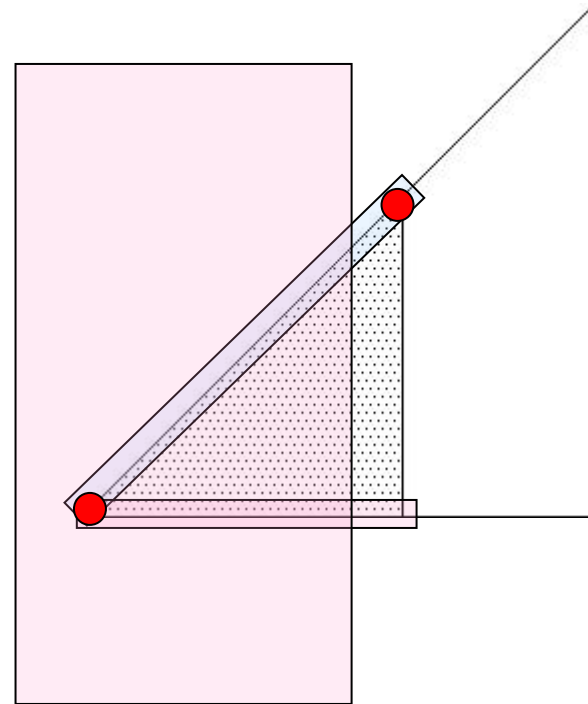
Resume Signals

- Lower facet of LT, not GT
- Lower facet of LT and GT

Suspend Signals

- Upper facet of LT, not GT
- Upper facet of LT and GT

The number of control signals reduces from $O(n)$ to $O(1)$.



Generalization (details in the paper)

- Concepts and algorithm
 - Multi-dimensional time/arrays
- Algorithm
 - Recursion on dimensions
 - Minimal set of time instants for control signals
 - Local time domain can be a union of polyhedra
 - Global time domain must be a single polyhedron

Contributions

- **Problem formulation & Analysis**
 - Good understanding on when and which computations need to be done by each processor
- **Characteristics of time instants for control signals**
- **Algorithms for exact solution**

Future work

■ Distribution

□ Linear arrays

- Distribution scheme for linear arrays

□ Multidimensional arrays

■ Implementation

- Algorithm for computing the set of time instants of control signals.

Related work

	Approach
Guillou et. al.	Each processor scans the global time domain
Bowden et. al.	When control signals can be expressed as a quadratic function of processor index, distribution is embedded into architecture.

Questions

On Control Signals for Multidimensional Time



DaeGon Kim
Gautam
Sanjay V. Rajopadhye



kim@cs.colostate.edu
gautam@cs.colostate.edu
Sanjay.Rajopadhye@colostate.edu