Filter placement on a pipelined architecture

Anne Benoit, Fanny Dufossé, Yves Robert
GRAAL team, LIP, ENS Lyon, France

APDCM'09 in Roma, Italy
May 25, 2009

Introduction and motivation

- Schedule an application onto a computational platform, with some criteria to optimize
- Target application
  - Streaming application: several data sets are processed by a set of filtering query services
  - Ordered or free ordering of the services
- Target platform
  - Linear chain of servers: hierarchical network
  - Different service/communication cost models
- Optimization criteria
  - Period: inverse of throughput; time between two data sets
  - Latency: response time for a single data set

Mapping *filtering streaming applications* on a *pipelined architecture*

Introduction and motivation

- Schedule an application onto a computational platform, with some criteria to optimize
- Target application
  - Streaming application: several data sets are processed by a set of filtering query services
  - Ordered or free ordering of the services
- Target platform
  - Linear chain of servers: hierarchical network
  - Different service/communication cost models
- Optimization criteria
  - Period: inverse of throughput; time between two data sets
  - Latency: response time for a single data set

Mapping *filtering streaming applications* on a *pipelined architecture*

Introduction and motivation

- Schedule an application onto a computational platform, with some criteria to optimize
- Target application
  - Streaming application: several data sets are processed by a set of filtering query services
  - Ordered or free ordering of the services
- Target platform
  - Linear chain of servers: hierarchical network
  - Different service/communication cost models
- Optimization criteria
  - Period: inverse of throughput; time between two data sets
  - Latency: response time for a single data set

Mapping *filtering streaming applications* on a *pipelined architecture*

Introduction and motivation

- Schedule an application onto a computational platform, with some criteria to optimize
- Target application
  - Streaming application: several data sets are processed by a set of filtering query services
  - Ordered or free ordering of the services
- Target platform
  - Linear chain of servers: hierarchical network
  - Different service/communication cost models
- Optimization criteria
  - Period: inverse of throughput; time between two data sets
  - Latency: response time for a single data set

Mapping *filtering streaming applications* on a *pipelined architecture*

Introduction and motivation

- Schedule an application onto a computational platform, with some criteria to optimize
- Target application
  - Streaming application: several data sets are processed by a set of filtering query services
  - Ordered or free ordering of the services
- Target platform
  - Linear chain of servers: hierarchical network
  - Different service/communication cost models
- Optimization criteria
  - Period: inverse of throughput; time between two data sets
  - Latency: response time for a single data set

Mapping *filtering streaming applications* on a *pipelined architecture*

Related work

- Filtering query services: resemble classical pipelined workflow graphs, extensively studied in the literature [DataCutter project, Wu et al, Benoit et al, ...].
- Filtering property: query optimization over web services [Srivastava et al], general data streams [Babu et al], database predicate processing [Chaudhuri et al, Hellerstein].
- Scheduling unreliable jobs on parallel machines: service selectivities correspond to job failure probabilities [Detti et al]
- Recent paper by Srivastava, Munagala and Widom: independent filtering services, linear array of servers, latency minimization. Problem left open with arbitrary service costs.

Related work

- Filtering query services: resemble classical pipelined workflow graphs, extensively studied in the literature [DataCutter project, Wu et al, Benoit et al, ...].
- Filtering property: query optimization over web services [Srivastava et al], general data streams [Babu et al], database predicate processing [Chaudhuri et al, Hellerstein].
- Scheduling unreliable jobs on parallel machines: service selectivities correspond to job failure probabilities [Detti et al]
- Recent paper by Srivastava, Munagala and Widom: independent filtering services, linear array of servers, latency minimization. Problem left open with arbitrary service costs.

Main contributions

- Proof that Srivastava's problem is NP-hard
- Extension of the problem when services are no longer independent: fixed prescribed order
- Extension of the problem for period minimization
- Impact of communication costs on the problem complexity

Outline

Framework: application and platform

- Target application $\{C_1, C_2, \ldots, C_n\}$: set of n filtering services
- Streaming application: several data sets, each processed by every services
- Data communicated from one service to another

- Linear chain of m servers $S_1, \ldots, S_m$
- Server S_u can only send data to S_{u+1} $(1 \leq u \leq m-1)$
- Hierarchical network: S_1 acquires data and S_m outputs results

- Service C_i: selectivity σ_i, basic cost $c_{i,u}$ on server S_u
- *Proportional* costs $c_{i,u} = \frac{w_i}{s_u}$ versus *Arbitrary* costs
- pred(C_i): predecessors in the mapping. Execution cost: $\left(\prod_{C_j \in \text{pred}(C_i)} \sigma_j\right) \times c_{i,u}$

Framework: application and platform

- Target application $\{C_1, C_2, \ldots, C_n\}$: set of n filtering services
- Streaming application: several data sets, each processed by every services
- Data communicated from one service to another

- Linear chain of m servers $S_1, \ldots, S_m$
- Server S_u can only send data to S_{u+1} $(1 \leq u \leq m-1)$
- Hierarchical network: S_1 acquires data and S_m outputs results

- Service C_i: selectivity σ_i, basic cost $c_{i,u}$ on server S_u
- *Proportional* costs $c_{i,u} = \frac{w_i}{s_u}$ versus *Arbitrary* costs
- pred(C_i): predecessors in the mapping. Execution cost: $\left(\prod_{C_j \in \text{pred}(C_i)} \sigma_j\right) \times c_{i,u}$

Framework: application and platform

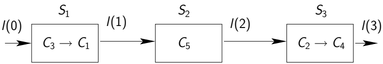
- Target application $\{C_1, C_2, \ldots, C_n\}$: set of n filtering services
- Streaming application: several data sets, each processed by every services
- Data communicated from one service to another

- Linear chain of m servers $S_1, \ldots, S_m$
- Server S_u can only send data to S_{u+1} $(1 \leq u \leq m-1)$
- Hierarchical network: S_1 acquires data and S_m outputs results

- Service C_i: selectivity σ_i, basic cost $c_{i,u}$ on server S_u
- *Proportional* costs $c_{i,u} = \frac{w_i}{s_u}$ versus *Arbitrary* costs
- pred(C_i): predecessors in the mapping. Execution cost: $\left(\prod_{C_j \in \text{pred}(C_i)} \sigma_j\right) \times c_{i,u}$

Framework: rules and mapping

- Independent services (*Free*).

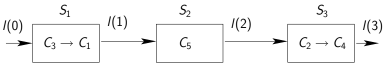


- Mapping: permutation π of services + allocation function a
- $\pi = [3, 1, 5, 2, 4]$, $a(1) = a(3) = 1$, $a(5) = 2$ and $a(2) = a(4) = 3$
- Execution cost of C_5: $\sigma_3 \sigma_1 c_{5,2}$

- Fixed ordering of services (*Ordered*): identical but π is fixed to $[1, 2, \ldots, n]$ (no permutation of services)
- Execution cost of C_i: $\left(\prod_{j<i} \sigma_j\right) c_{i,a(i)}$

Framework: rules and mapping

- Independent services (*Free*).



- Mapping: permutation π of services + allocation function a
- $\pi = [3, 1, 5, 2, 4]$, $a(1) = a(3) = 1$, $a(5) = 2$ and $a(2) = a(4) = 3$
- Execution cost of C_5: $\sigma_3 \sigma_1 c_{5,2}$
- Fixed ordering of services (*Ordered*): identical but π is fixed to $[1, 2, \dots, n]$ (no permutation of services)
- Execution cost of C_i: $\left(\prod_{j<i} \sigma_j\right) c_{i,a(i)}$

Framework: communication model

- Without communication costs (*NoCost*)
- With communication costs (*Cost*): no computation and communication overlap, model of Srivastava et al
- ALLOC_v: set of services allocated to server S_v
  $\text{PRED}_u = \bigcup_{v=1}^{u-1} \text{ALLOC}_v$, $\text{UPTO}_u = \bigcup_{v=1}^{u} \text{ALLOC}_v$
- Communication cost between S_u and S_{u+1}:
  $C_{comm}(u) = l(u) \times \prod_{C_j \in \text{UPTO}_u} \sigma_j$
- $l(u)$: inverse of bandwidth of link $S_u \rightarrow S_{u+1}$
- Input for server S_1: cost $C_{comm}(0)$, bandwidth $1/l(0)$
- Output for server S_m: cost $C_{comm}(m)$, bandwidth $1/l(m)$

Framework: communication model

- Without communication costs (*NoCost*)
- With communication costs (*Cost*): no computation and communication overlap, model of Srivastava et al
- ALLOC_v: set of services allocated to server S_v
  $\text{PRED}_u = \bigcup_{v=1}^{u-1} \text{ALLOC}_v$, $\text{UPTO}_u = \bigcup_{v=1}^{u} \text{ALLOC}_v$
- Communication cost between S_u and S_{u+1}:
  $C_{comm}(u) = l(u) \times \prod_{C_j \in \text{UPTO}_u} \sigma_j$
- $l(u)$: inverse of bandwidth of link $S_u \rightarrow S_{u+1}$
- Input for server S_1: cost $C_{comm}(0)$, bandwidth $1/l(0)$
- Output for server S_m: cost $C_{comm}(m)$, bandwidth $1/l(m)$

Framework: communication model

- Without communication costs (*NoCost*)
- With communication costs (*Cost*): no computation and communication overlap, model of Srivastava et al
- ALLOC_v: set of services allocated to server S_v
  $\text{PRED}_u = \bigcup_{v=1}^{u-1} \text{ALLOC}_v$, $\text{UPTO}_u = \bigcup_{v=1}^{u} \text{ALLOC}_v$
- Communication cost between S_u and S_{u+1}:
  $C_{comm}(u) = l(u) \times \prod_{C_j \in \text{UPTO}_u} \sigma_j$
- $l(u)$: inverse of bandwidth of link $S_u \rightarrow S_{u+1}$
- Input for server S_1: cost $C_{comm}(0)$, bandwidth $1/l(0)$
- Output for server S_m: cost $C_{comm}(m)$, bandwidth $1/l(m)$

Framework: communication model

- Without communication costs (*NoCost*)
- With communication costs (*Cost*): no computation and communication overlap, model of Srivastava et al
- ALLOC_v: set of services allocated to server S_v
  $\mathrm{PRED}_u = \bigcup_{v=1}^{u-1} \mathrm{ALLOC}_v$, $\mathrm{UPTO}_u = \bigcup_{v=1}^{u} \mathrm{ALLOC}_v$
- Communication cost between S_u and S_{u+1}:
  $C_{comm}(u) = l(u) \times \prod_{C_j \in \mathrm{UPTO}_u} \sigma_j$
- $l(u)$: inverse of bandwidth of link $S_u \rightarrow S_{u+1}$
- Input for server S_1: cost $C_{comm}(0)$, bandwidth $1/l(0)$
- Output for server S_m: cost $C_{comm}(m)$, bandwidth $1/l(m)$

Framework: objective functions

- Period (PER): limited by the slowest (bottleneck) server. Time interval between the processing of two data sets.
- Latency (LAT): sum of the costs incurred by all services in the mapping. Time required for one data set to be processed by all the services.

- Server S_u: computation cost $C_{comp}(u)$
- Let the services in ALLOC_u be $C_1 \rightarrow C_2 \rightarrow \ldots \rightarrow C_k$
- $C_{comp}(u) = \left(\prod_{C_j \in \text{PRED}_u} \sigma_j\right) \sum_{i=1}^{k} \left(\prod_{q=1}^{i-1} \sigma_q\right) \times c_{i,u}$
- *NoCost*: $\mathcal{P} = \max_{1 \le u \le m}\{C_{comp}(u)\}$, $\mathcal{L} = \sum_{u=1}^{m} C_{comp}(u)$
- *Cost*: $\mathcal{P} = \max_{1 \le u \le m}\{C_{comm}(u-1) + C_{comp}(u) + C_{comm}(u)\}$, $\mathcal{L} = C_{comm}(0) + \sum_{u=1}^{m} \left(C_{comp}(u) + C_{comm}(u)\right)$

Framework: objective functions

- Period (PER): limited by the slowest (bottleneck) server. Time interval between the processing of two data sets.
- Latency (LAT): sum of the costs incurred by all services in the mapping. Time required for one data set to be processed by all the services.

- Server S_u: computation cost $C_{comp}(u)$
- Let the services in ALLOC_u be $C_1 \rightarrow C_2 \rightarrow \ldots \rightarrow C_k$
- $C_{comp}(u) = \left(\prod_{C_j \in \text{PRED}_u} \sigma_j\right) \sum_{i=1}^{k} \left(\prod_{q=1}^{i-1} \sigma_q\right) \times c_{i,u}$

- *NoCost*: $\mathcal{P} = \max_{1 \le u \le m}\{C_{comp}(u)\}$, $\mathcal{L} = \sum_{u=1}^{m} C_{comp}(u)$
- *Cost*: $\mathcal{P} = \max_{1 \le u \le m}\{C_{comm}(u-1) + C_{comp}(u) + C_{comm}(u)\}$, $\mathcal{L} = C_{comm}(0) + \sum_{u=1}^{m} (C_{comp}(u) + C_{comm}(u))$

Framework: objective functions

- Period ($\textsc{Per}$): limited by the slowest (bottleneck) server. Time interval between the processing of two data sets.
- Latency ($\textsc{Lat}$): sum of the costs incurred by all services in the mapping. Time required for one data set to be processed by all the services.

- Server S_u: computation cost $C_{comp}(u)$
- Let the services in $\textsc{Alloc}_u$ be $C_1 \to C_2 \to \ldots \to C_k$
- $C_{comp}(u) = \left(\prod_{C_j \in \textsc{Pred}_u} \sigma_j\right) \sum_{i=1}^{k} \left(\prod_{q=1}^{i-1} \sigma_q\right) \times c_{i,u}$

- *NoCost*: $\mathcal{P} = \max_{1 \leq u \leq m}\{C_{comp}(u)\}$, $\mathcal{L} = \sum_{u=1}^{m} C_{comp}(u)$
- *Cost*: $\mathcal{P} = \max_{1 \leq u \leq m}\{C_{comm}(u-1) + C_{comp}(u) + C_{comm}(u)\}$, $\mathcal{L} = C_{comm}(0) + \sum_{u=1}^{m} (C_{comp}(u) + C_{comm}(u))$

Framework: objective functions

- Period ($\textsc{Per}$): limited by the slowest (bottleneck) server. Time interval between the processing of two data sets.
- Latency ($\textsc{Lat}$): sum of the costs incurred by all services in the mapping. Time required for one data set to be processed by all the services.

- Server S_u: computation cost $C_{comp}(u)$
- Let the services in $\textsc{Alloc}_u$ be $C_1 \rightarrow C_2 \rightarrow \dots \rightarrow C_k$
- $C_{comp}(u) = \left(\prod_{C_j \in \textsc{Pred}_u} \sigma_j\right) \sum_{i=1}^{k} \left(\prod_{q=1}^{i-1} \sigma_q\right) \times c_{i,u}$

- *NoCost*: $\mathcal{P} = \max_{1 \le u \le m}\{C_{comp}(u)\}$, $\mathcal{L} = \sum_{u=1}^{m} C_{comp}(u)$
- *Cost*: $\mathcal{P} = \max_{1 \le u \le m}\{C_{comm}(u-1) + C_{comp}(u) + C_{comm}(u)\}$, $\mathcal{L} = C_{comm}(0) + \sum_{u=1}^{m} \left(C_{comp}(u) + C_{comm}(u)\right)$

Framework: summary

- Problem denoted by *XYZ-Obj*, where:
  - $X = O|F$: service ordering (*Ordered* or *Free*);
  - $Y = P|A$: service costs (*Proportional* or *Arbitrary*);
  - $Z = C|N$: communication costs (*Cost* or *NoCost*);
  - $Obj = \text{PER}|\text{LAT}$: objective function.
- *: any instance of the problem
- Examples: FAC-LAT, O**-PER, ...
- 16 problems to solve

Framework: summary

- Problem denoted by *XYZ-Obj*, where:
  - $X = O|F$: service ordering (*Ordered* or *Free*);
  - $Y = P|A$: service costs (*Proportional* or *Arbitrary*);
  - $Z = C|N$: communication costs (*Cost* or *NoCost*);
  - $Obj = \textsc{Per}|\textsc{Lat}$: objective function.
- *: any instance of the problem
- Examples: FAC-LAT, O**-PER, ...
- 16 problems to solve

Framework: summary

- Problem denoted by *XYZ-Obj*, where:
  - $X = O|F$: service ordering (*Ordered* or *Free*);
  - $Y = P|A$: service costs (*Proportional* or *Arbitrary*);
  - $Z = C|N$: communication costs (*Cost* or *NoCost*);
  - $Obj = \text{PER}|\text{LAT}$: objective function.
- *: any instance of the problem
- Examples: FAC-LAT, O**-PER, ...
- 16 problems to solve

Framework: summary

- Problem denoted by *XYZ-Obj*, where:
  - $X = O|F$: service ordering (*Ordered* or *Free*);
  - $Y = P|A$: service costs (*Proportional* or *Arbitrary*);
  - $Z = C|N$: communication costs (*Cost* or *NoCost*);
  - $Obj = \textsc{Per}|\textsc{Lat}$: objective function.
- *: any instance of the problem
- Examples: FAC-$\textsc{Lat}$, O**-$\textsc{Per}$, ...
- 16 problems to solve

Outline

1. Framework
2. Complexity results
3. Conclusion

Period minimization

Theorem

*All problems F**-PER are NP-hard (free ordering of services).*
*All problems O**-PER have polynomial complexity.*

- NP-hardness: easy reduction from 2-Partition: instance of FPN-PER with n services and 2 identical servers, $\sigma_i = 1$, cost $c_{i,u} = a_i$.
- Algorithm which computes the optimal mapping for problem OAC-PER in time $O(m \times n^3)$:
  - Input – n services of selectivities $\sigma_1, ..., \sigma_n$, m servers with a matrix of costs c, and a vector of communication costs l
  - Result – mapping function a optimizing the period
  - $P(i,j)$: optimal period with last i services and last j servers. $a(i,j,.)$: corresponding allocation function.

Period minimization

Theorem

*All problems F**-PER are NP-hard (free ordering of services).*
*All problems O**-PER have polynomial complexity.*

- NP-hardness: easy reduction from 2-Partition: instance of FPN-PER with n services and 2 identical servers, $\sigma_i = 1$, cost $c_{i,u} = a_i$.
- Algorithm which computes the optimal mapping for problem OAC-PER in time $O(m \times n^3)$:
  - Input – n services of selectivities $\sigma_1, ..., \sigma_n$, m servers with a matrix of costs c, and a vector of communication costs l
  - Result – mapping function a optimizing the period
  - $P(i,j)$: optimal period with last i services and last j servers. $a(i,j,.)$: corresponding allocation function.

Period minimization

Theorem

*All problems F**-*PER *are NP-hard (free ordering of services).*
*All problems O**-*PER *have polynomial complexity.*

- NP-hardness: easy reduction from 2-Partition: instance of FPN-PER with n services and 2 identical servers, $\sigma_i = 1$, cost $c_{i,u} = a_i$.
- Algorithm which computes the optimal mapping for problem OAC-PER in time $O(m \times n^3)$:
  - Input – n services of selectivities $\sigma_1, ..., \sigma_n$, m servers with a matrix of costs c, and a vector of communication costs l
  - Result – mapping function a optimizing the period
  - $P(i,j)$: optimal period with last i services and last j servers. $a(i,j,.)$: corresponding allocation function.

Algorithm for OAC-PER

$P(0,1) = l(m-1) + l(m)$;
for $j = 2$ **to** m *(No services)* **do**
$P(0,j) = \max\{l(m-j) + l(m-j+1), P(0,j-1)\}$;
end
for $i = 1$ **to** n *(One server)* **do**
$P(i,1) = l(m-1) + c_{n-i+1,m} + \sigma_{n-i+1}(P(i-1,1) - l(m-1))$;
$\forall 1 \leq k \leq i,\ a(i,1,n-k+1) = m$;
end
for $j = 2$ **to** m *(Increase server nb)* **do**
for $i = 1$ **to** n *(Increase service nb)* **do**
$\forall\ 0 \leq r \leq i,\ f(r) = \max\{l(m-j) + \sum_{q=1}^{r} \prod_{p=1}^{q-1} \sigma_{n-i+p} c_{n-i+q,m-j+1} + \prod_{p=1}^{r} \sigma_{n-i+p} l(m-j+1),\ \ \prod_{p=1}^{r} \sigma_{n-i+p} P(i-r,j-1)\}$;
$k = argmin_{0 \leq r \leq i}\{f(r)\}$; $P(i,j) = f(k)$;
$\forall 1 \leq q \leq k,\quad a(i,j,n-i+q) = m-j+1$;
$\forall k < q \leq i,\quad a(i,j,n-i+q) = a(i-k,j-1,n-i+q)$;
end
end

Latency minimization

Theorem

All problems FA-LAT are NP-hard (free ordering of services, arbitrary costs).*
*All problems O**-LAT have polynomial complexity.*

- FP*-LAT: Srivastava, polynomial complexity
- With arbitrary costs, even without communication costs, the problem becomes NP-hard: involved reduction from 2-Partition
- Polynomial algorithm for OAC-LAT, in $O(n^3m)$

Latency minimization

Theorem

All problems FA-LAT are NP-hard (free ordering of services, arbitrary costs).*
*All problems O**-LAT have polynomial complexity.*

- FP*-LAT: Srivastava, polynomial complexity
- With arbitrary costs, even without communication costs, the problem becomes NP-hard: involved reduction from 2-Partition
- Polynomial algorithm for OAC-LAT, in $O(n^3 m)$

Latency minimization

Theorem

All problems FA-LAT are NP-hard (free ordering of services, arbitrary costs).*
*All problems O**-LAT have polynomial complexity.*

- FP*-LAT: Srivastava, polynomial complexity
- With arbitrary costs, even without communication costs, the problem becomes NP-hard: involved reduction from 2-Partition
- Polynomial algorithm for OAC-LAT, in $O(n^3 m)$

Algorithm for OAN-LAT

for $j = 1$ **to** m **do**
  $L(0, j) = 0;$
end
for $i = 1$ **to** n **do**
  $L(i, 1) = c_{n-i+1,m} + \sigma_{n-i+1} L(i-1, 1);$
  $\forall 1 \leq k \leq i,\ a(i, 1, n-k+1) = m;$
end
for $j = 2$ **to** m **do**
  for $i = 1$ **to** n **do**
    $\forall\ 0 \leq l \leq i,\ f(l) = \sum_{i'=1}^{l} \left(\prod_{q=1}^{i'-1} \sigma_{n-i+q}\right) c_{n-i+i',m-j+1} + \left(\prod_{q=1}^{l} \sigma_{n-i+q}\right) L(i-l, j-1);$
    $k = argmin_{0 \leq l \leq i}\{f(l)\}; \quad L(i, j) = f(k);$
    $\forall 1 \leq q \leq k, \quad a(i, j, n-i+q) = m-j+1;$
    $\forall k < q \leq i, \quad a(i, j, n-i+q) = a(i-k, j-1, n-i+q);$
  end
end

Outline

Conclusion and future work

- Mapping filtering streaming applications onto a linear array of heterogeneous servers, two optimization criteria
- Complexity of all optimization problems; note that there is no impact from communication costs:

	PER	LAT
O**	Polynomial	Polynomial
FP*	NP-complete	Polynomial [Srivastava]
FA*	NP-complete	NP-complete

- Future work: Approximation algorithms and lower bounds for NP-hard instances of the problem

Conclusion and future work

- Mapping filtering streaming applications onto a linear array of heterogeneous servers, two optimization criteria
- Complexity of all optimization problems; note that there is no impact from communication costs:

	PER	LAT
O**	Polynomial	Polynomial
FP*	NP-complete	Polynomial [Srivastava]
FA*	NP-complete	NP-complete

- Future work: Approximation algorithms and lower bounds for NP-hard instances of the problem