Transactional Partitioning:

A New Abstraction for Main-Memory Databases

Vivek Shah, Marcos Vaz Salles

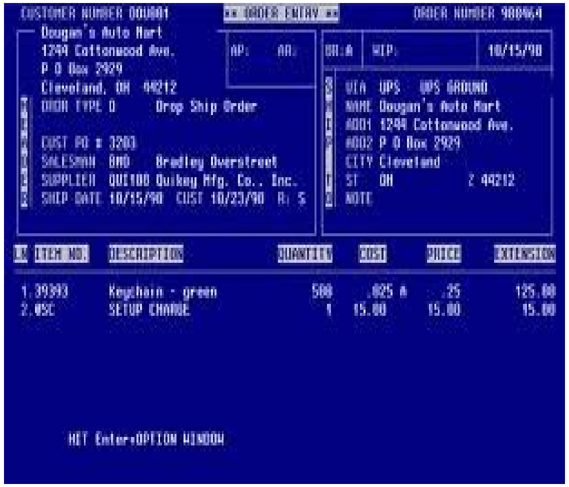
DMS Lab, Department of Computer Science (DIKU)
University of Copenhagen

HIPERFIT workshop, Skodsborg

Online Transaction Processing(OLTP) Application Goldmine

Reservation system

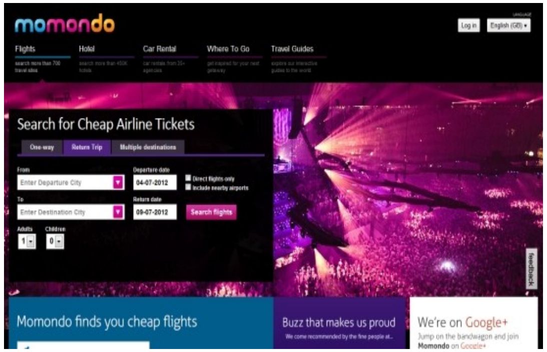
Banking system

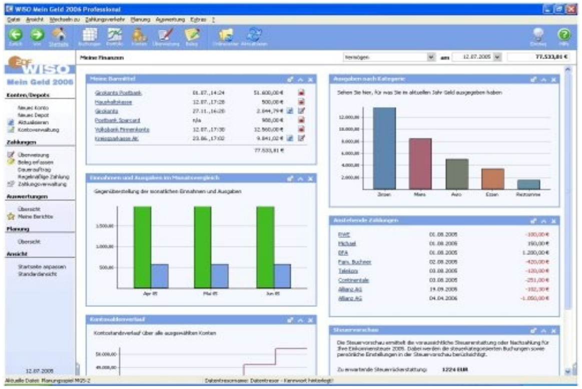


```
CUSTOMER NUMBER DOUA01        ** ORDER ENTRY **          ORDER NUMBER 900064
  Dougan's Auto Mart                 AP:    AR:     OR:A  HIP:              10/15/90
  1299 Cottonwood Ave.
  P O Box 2929                                      VIA  UPS   UPS GROUND
  Cleveland, OH  44212                              NAME Dougan's Auto Mart
  ORDR TYPE D      Drop Ship Order                  ADD1 1299 Cottonwood Ave.
                                                    ADD2 P O Box 2929
  CUST PO # 3203                                    CITY Cleveland
  SALESMAN  BND    Bradley Overstreet               ST   OH           Z 44212
  SUPPLIER  QUI100 Quikey Mfg. Co., Inc.            NOTE
  SHIP DATE 10/15/90  CUST 10/23/90  R: S

LN ITEM NO.   DESCRIPTION              QUANTITY    COST     PRICE    EXTENSION
 1.39393      Keychain - green              500   .025 A     .25       125.00
 2.USC        SETUP CHARGE                    1  15.00     15.00        15.00

      HIT Enter=OPTION WINDOW
```

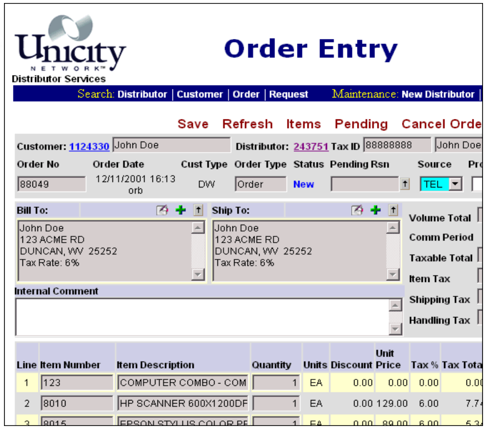
Order Entry system

OLTP Application Goldmine





TPC-B



Unicity NETWORK
Distributor Services

Order Entry

Search: Distributor | Customer | Order | Request   Maintenance: New Distributor |

Save Refresh Items Pending Cancel Orde

Customer: 1124330 John Doe Distributor: 243751 Tax ID 88888888 John Doe

Order No	Order Date	Cust Type	Order Type	Status	Pending Rsn	Source	Pr
88049	12/11/2001 16:13 orb	DW	Order	New		TEL	

Bill To:
John Doe
123 ACME RD
DUNCAN, WV 25252
Tax Rate: 6%

Ship To:
John Doe
123 ACME RD
DUNCAN, WV 25252
Tax Rate: 6%

Volume Total
Comm Period
Taxable Total
Item Tax
Shipping Tax
Handling Tax

Internal Comment

Line	Item Number	Item Description	Quantity	Units	Discount	Unit Price	Tax %	Tax Tota
1	123	COMPUTER COMBO - COM	1	EA	0.00	0.00	0.00	0.0
2	8010	HP SCANNER 600X1200DF	1	EA	0.00	129.00	6.00	7.7
3	8015	EPSON STYLUS COLOR PE	1	EA	0.00	89.00	6.00	5.3

TPC-C

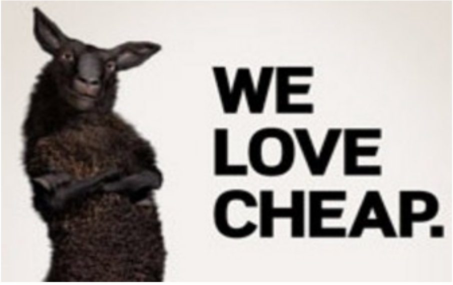
TPC-E

CollabNet.
Enterprise Cloud Development
Enabling integrated collaborative teams to enhance productivity and speed time-to-market.

Online Transaction Processing(OLTP) Application Trends

OLTP Application Evolution

- Internet
- Computer Hardware



- Computer Software
  Open Source
- Cloud Computing

OLTP Application Trends

=

Throughput

Latency

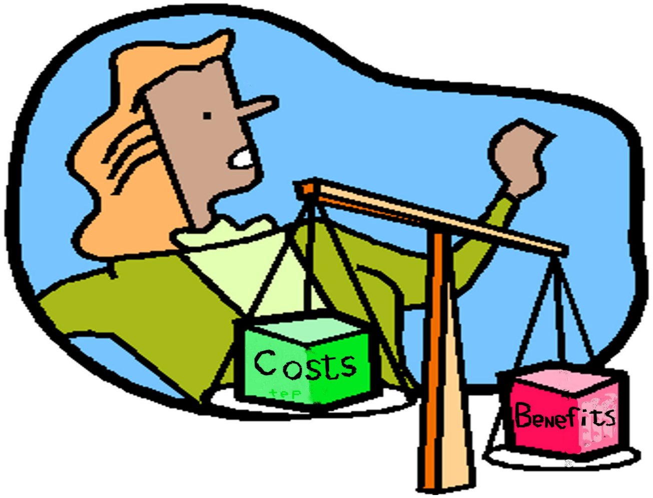
Resource Utilization

OLTP Application Trends

=

Maintenance

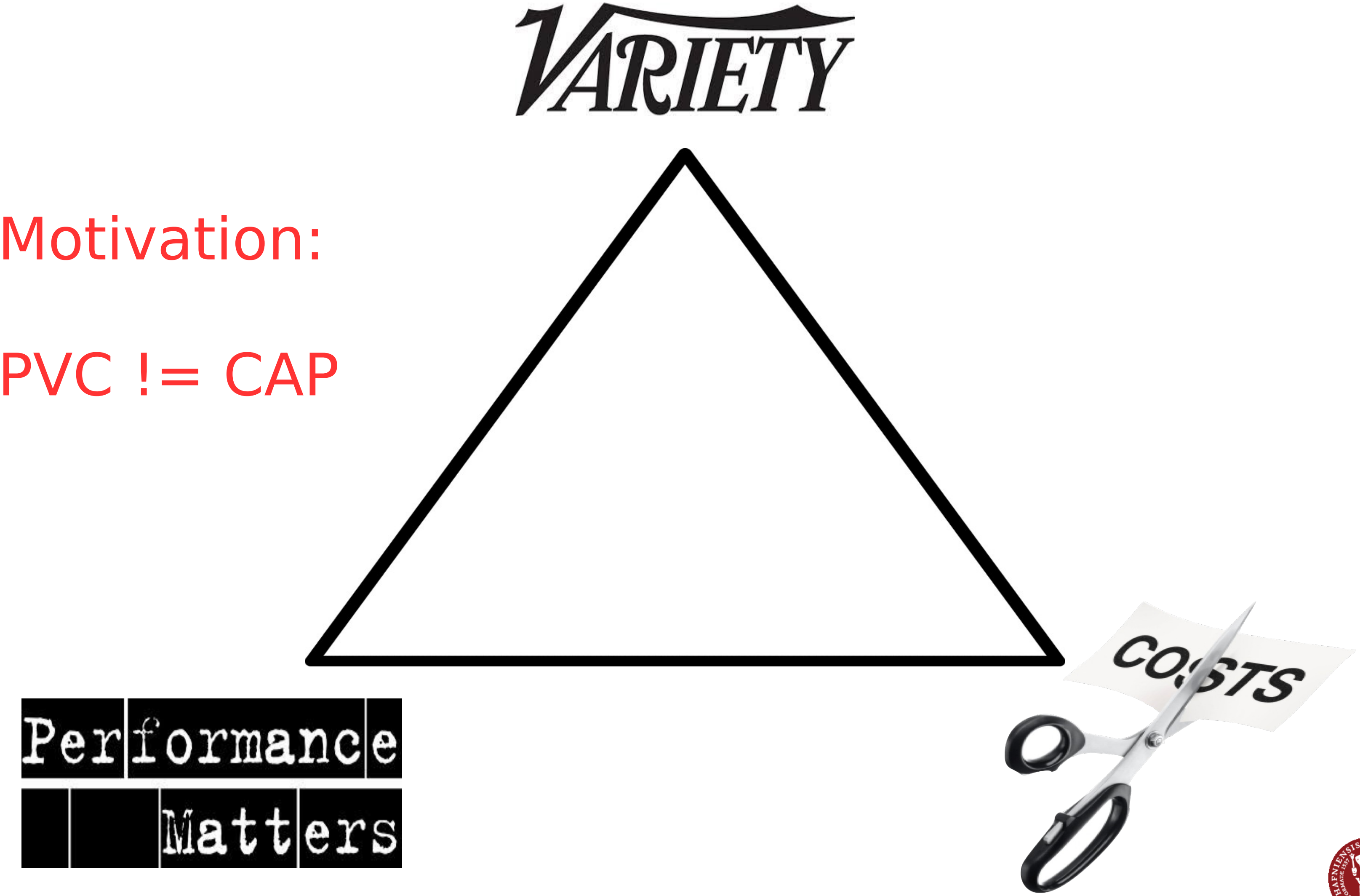
Development



PVC Triangle

Motivation:

PVC != CAP



Outline

- Motivation ✓
- PVC Problem in Existing Solutions
- Logical Partition Solution
- Challenges
- Conclusion

Online Transaction Processing Application(OLTP) Properties

Interactive

Update heavy

Update consistency

Highly Interactive Commodity hardware Consistent on Update

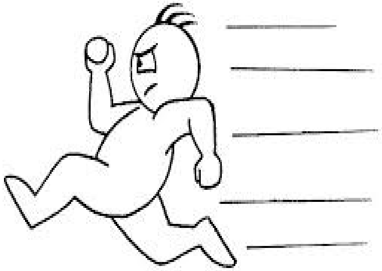
HICCUP

Commodity hardware

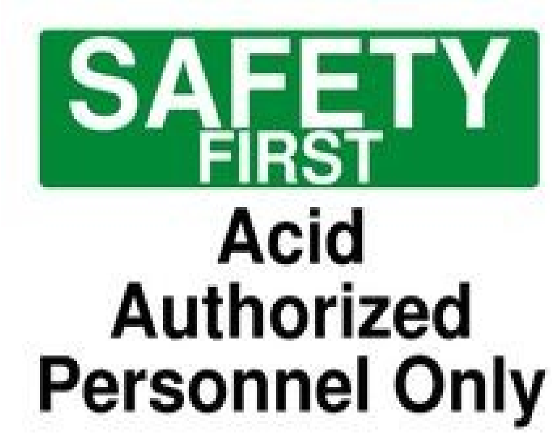
OLTP Application Properties

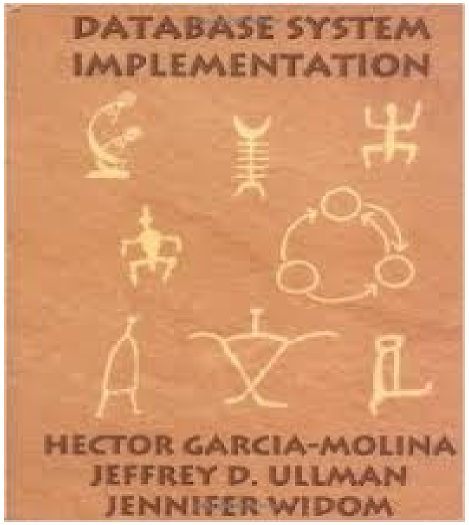
Fast

Short



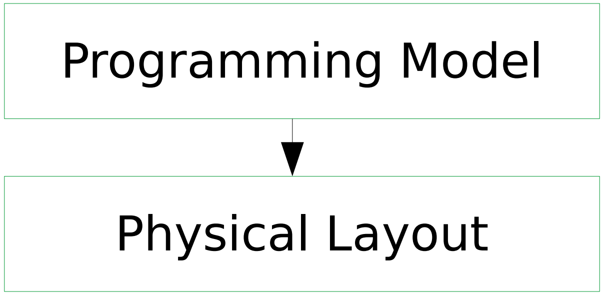
THE OLTP System for OLTP Applications





OLTP Programming Models

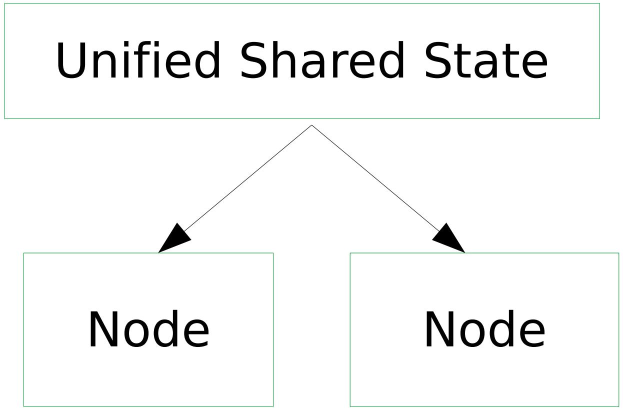
- Logical Programming model
  - Not Data model
  - Abstraction of the physical layout
- Physical Implementation



Current OLTP Programming Models

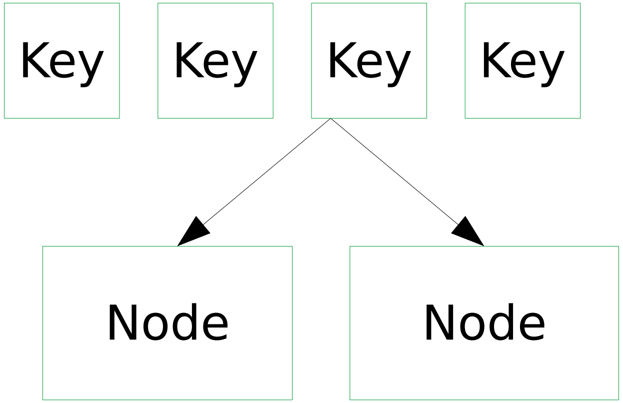
- Unified view of shared state
  - Classic relational model
- Distributed storage oriented partitioned by key
  - Key-value store model

Unified View of Shared State Model



- Strong consistency, high level data model
- Hiding partitioning → Hard to reason about performance
- Partitioning is key → Houdini systems
- PVC or PVC or PVC

Distributed Storage Oriented Partitioned by Key Model



- Weak/no consistency, low level storage oriented data model
- Hard to reason about locality
- Exposing partitioning → Reason about performance
- Control performance (build yourself), variety
- PVC or PVC or PVC

THE PROBLEM

How to build an OLTP system that

- Maintains ACID guarantees
- Exposes partitioning in the programming model
  - Exposes program costs

Write good programs

Programming Model

- Maps the programming model to the commodity-hardware cluster
- Guarantees high resource utilization

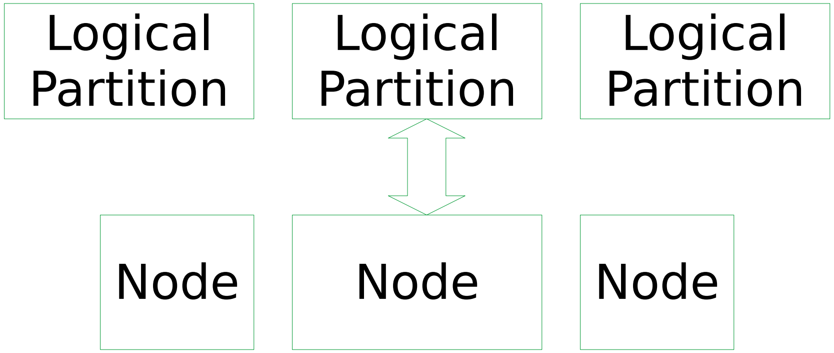
Run programs efficiently

Implementation

Outline

- Motivation ✓
- PVC Problem in Existing Solutions ✓
- Logical Partition Solution
- Challenges
- Conclusion

Our Solution (Logical Partitioning)



- Logical Partition = Logical unit of execution and associated storage (e.g., warehouse in TPC-C)
- Accessible through function calls → Transactions
- Transactions are local, invoked with logical partition
- Transactions can invoke other transactions

```
txn T1 (...) {
  ......
  return res;
}
```

Local execution

```
EXEC T1 (x_input) ON PARTITION (L0)
```

Invoke txn with input parameters and logical partition identifier

```
txn T2 (...) {
   ......
   input' = f(..);

 res = EXEC T1(input') ON
     PARTITION (L1);

}
```

Txn Invocation

Our Solution (Logical Partitioning)

```
txn T1 (input) {
   ……
   ……
   return res;
}
```

Make it better

```
EXEC T1 (input) on PARTITION L0;
```

```
PARTITIONING FUNCTION map(input) {
    …..
    return logical_id;
}

T1 PARTITION MAPPER map;
```

Mapping function
for partition id

```
EXEC T1(input) on PARTITION (input);
```

New Order

Skodsborg

Order:
2 Longjing tea – Beijing

3 Liquorice - Copenhagen

Stock:
Longjing tea - 5

Item:
Longjing tea - $4
Liquorice - $3

Total cost: $17

Stock:
Liquorice - 3

Beijing

Copenhagen

New Order in Unified View of Shared State Model

```
txn new_order (w_id, d_id, c_id, order) {

  <wh,dist,cust> = gen_order_id(w_id, d_id, c_id, order);
                                                    Generate order id
  total = 0;
  for(ord_item in order.items) {
    amount = get_amount(ord_item);
    total += amount;                             Compute order item cost

    update_stock(ord_item, amount);
                                                         Update stock

    stock_info = get_dist_info_stock(ord_item);
    add("order_line", dist.order_id, w_id, d_id, stock_info,
    amount, ...);                                         Add order line
  }

  total_pay = (1 + wh.tax + dist.tax)*total* (1 - cust.discount);
  return total_pay;
                                                 Compute total order cost
}
```

New Order (How to use the new model?)

- Element of distribution
- Affinity of programs
- Increase in data and compute
- Warehouses (Intuitively from application)

New Order (How to use the new model?)

```
txn new_order(w_id, d_id, c_id, order) {

   <wh,dist,cust> = gen_order_id(w_id, d_id, c_id, order);

  total = 0;
  for(ord_item in order.items) {
     amount = get_amount(ord_item);                    Remote warehouse
     total += amount;
                                                                         Separate
     update_stock(ord_item, amount);                                     in a txn

                                                       Remote warehouse
     stock_info = get_dist_info_stock(ord_item);


    add("order_line", dist.order_id, w_id, d_id, stock_info,
    amount,...);
  }

  total_pay = (1 + wh.tax + dist.tax)*total*(1 - cust.discount);
  return total_pay;

}
```

New Order Stock Update using Logical Partitioning

```
txn new_order_update_stock(order) {
  Result = <>;

  for(ord_item in order.items) {
    amount = get_amount(ord_item);

    update_stock(ord_item, amount);

    stock_info = get_dist_info_stock(ord_item);

    append(result,<stock_info,amount>);
  }

  return result;
}
```

Compute order item cost

Update stock

Gather stock information
for order line

Use warehouse id as logical partition id

```
PARTITIONING FUNCTION map(w_id) {return w_id;};
new_order PARTITION MAPPER map;
new_order_update_stock PARTITION MAPPER map;
```

- Can I optimize more ?
- Details in the paper

New Order using Logical Partitioning

Generate order id

```
txn new_order (w_id, d_id, c_id, order) {
  <wh,dist,cust> = gen_order_id(w_id, d_id, c_id, order);
   results = <>;

  for(s_id in order.supplier_w_id) {
    temp_res = EXEC new_order_update_stock
                        (subset(order, s_id)) ON PARTITION (s_id);
    append(results,temp_res);
  }
```

Invoke stock update txn on supplier warehouses

```
  total = 0;
  for(result in results) {
    for(item_result in result) {
       total += item_result.amount;
       add("order_line", dist.order_id, w_id, d_id, item_result, ...);
    }
  }
```

Use results to compute order cost and add order line

```
  total_pay = (1+wh.tax+dist.tax)*total*(1-cust.discount);
  return total_pay;
}
```

Compute total order cost

What has changed ?

- Exposed partitioning
  - Cost of communication
  - Cost of co-ordination
  - Performance is visible, controllable
- Maintained ACID
  - Isolation is good
  - No need to reason about inter-leavings

Logical Partitioning Model

- Split the programming model into logical units of storage and execution
- Application Developer does splitting → WYSWYG
- Maintain ACID guarantees
- Transactions → Code Isolation → Partitioning Element
- Programs → Produce Data
- Separation of concerns → Honesty about cleverness → One man does not fix all

Challenges

- Implementation (ongoing work)
  - Mapping logical to physical partitions
    - Reuse main-memory shared-everything engine (Silo)
    - Cost model, workload variance, skew, scheduling
  - Local Concurrency Control & Global Commit
    - Optimistic concurrency control → Global commit
    - Less is more
- Evaluation
  - TPC-C (Varied configurations of physical partitions, workload parameters)
  - Oltpbench ?
- Cloud Integration
  - Programs, performance requirements, resources

Conclusion

Thank You

- Performance, Variance, Cost (PVC) → OLTP Trends
- Existing programming models do not meet PVC goals
- Logical Programming model
  - Expose partitioning → Use transactions
  - Provide global ACID guarantees
- Write Good Programs → Good abstraction
- Run good programs efficiently → Resource Utilization
- Logical Partitioning → PVC Goals → GET Ph.D.