SAS High Performance Computing: The Future is Not What it Used to Be

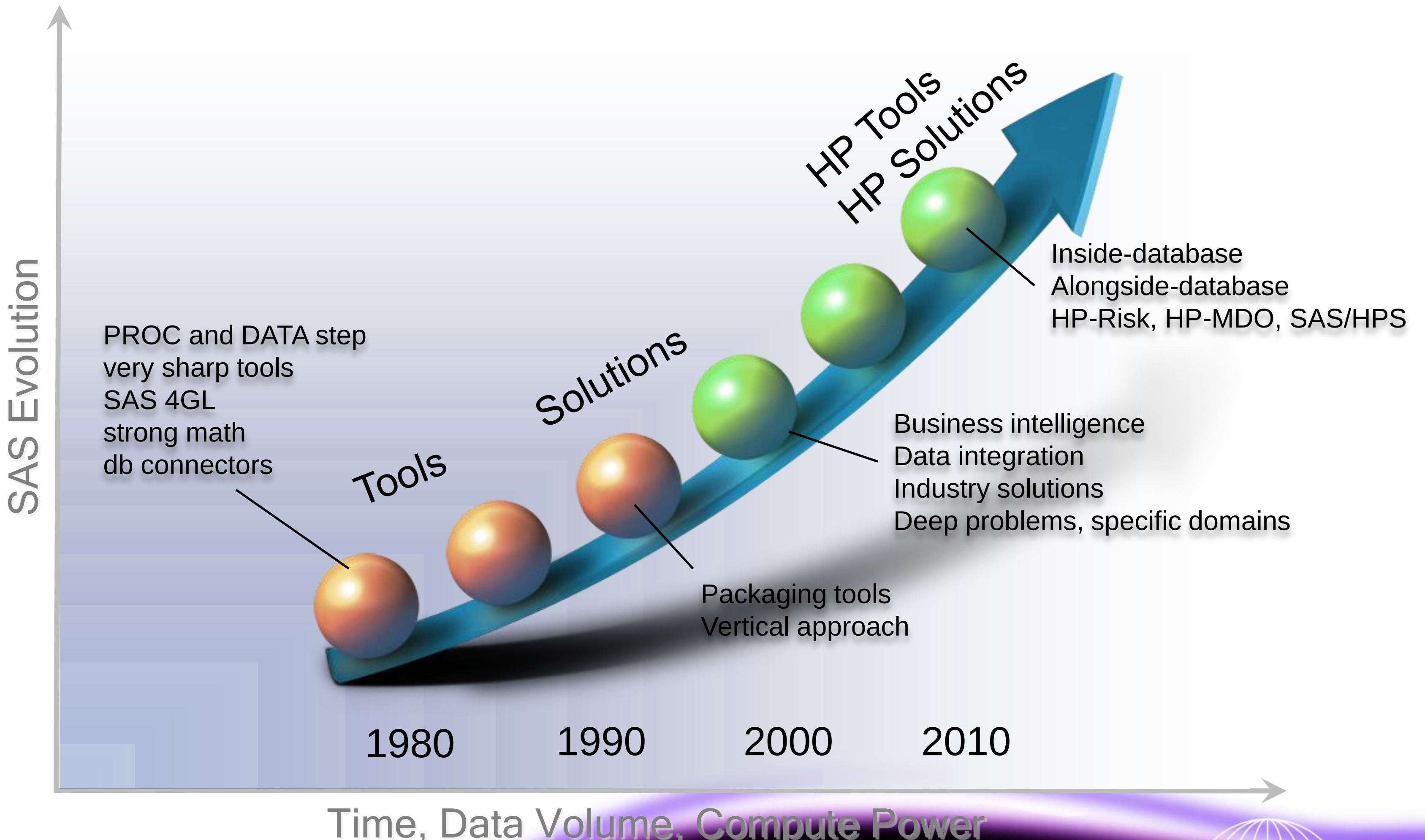
Paul Kent — VP, SAS Platform R&D
paul.kent@sas.com
twitter.com/hornpolish

Oliver Schabenberger — High Performance Analytics
oliver.schabenberger@sas.com

Premises

- Computing landscape has changed profoundly
  - Plentiful multi-core blade servers
  - Distributed computing platforms affordable
- Data landscape has changed profoundly
  - Data is much more plentiful
  - More MPP databases and file systems
- SAS High-Performance Computing
  - lives at the intersection of Big Data and Big Analytics
  - is a major area for tools and solution development
  - considers tasks of any analytic complexity

From Tools to Solutions to Tools and Solutions



Contents

Hardware Story

- Multi-socket, multi-core
- Commodity blades

Data Story

- Data explosion
- MPP databases

SAS High-Performance Computing

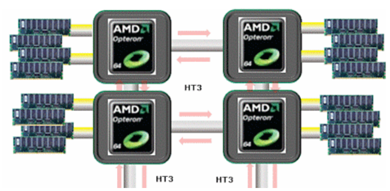
- Analytic Tiers
- Matching data and compute style

Trends in Hardware

- Multi-socket, multi-core
  - Modern platforms have one, two, or more sockets
  - Each socket houses a multi-core processor
  - e.g., dual quad core = 8 total cores in 2 sockets of 4 cores
  - 1 x 2, 1 x 4, 2 x 4: common for office PCs
  - 2 x 6, 2 x 8, 2 x 12: common for HPC blades
- Blade servers
  - A blade is a stripped-down computer
  - Blades are arranged in enclosures (chassis) that provide power, cooling, networking, interconnects
  - Chassis are arranged in racks

Trends in Hardware

Multi-socket,
multi-core platform



Commodity
blade

Chassis of blades

- Systems can be made CPU-heavy or disk-heavy
- 48 or 96 GB of RAM per blade not uncommon
  (golden rule: 4 GB RAM per core for HPC apps)
- 64-bit x86 Linux becoming standard

Big Analytics—Big Multiplier

- 2011 middle-of-the-road system
  - *hp* Proliant BL465c blades
  - 2 x 12 core 2.2 GHz processor (AMD Opteron 6174)
  - 64 GB of RAM
  - 2 x 300 GB drives
  - standard 10 GbE switches
- Rack of 48 blades (3 chassis of 16 blades each)
  - 1,152 cores
  - 3 TB of memory
  - 28 TB of storage
  - List price ~ $600 K

Possibility for HPC?

- Blade systems offer tremendous compute and storage power at low cost
- Software needs to take advantage of
  - Multi-core processors on each blade
  - Distributed shared nothing environment across blades
  - Local storage to read/write data and to reduce data movement
- Obvious choice for high-performance solutions with relatively small input data
- How do we cope with TB or 100's of GB of input/output?

Contents

Hardware Story

Multi-socket, multi-core

Commodity blades

Data Story

Data explosion

MPP databases

SAS High-Performance Computing

Analytic Tiers

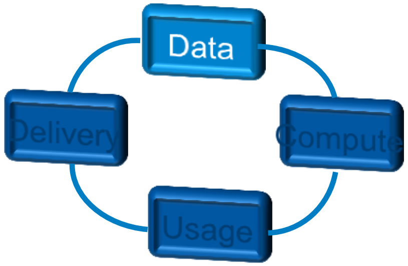
Matching data and compute style

Data Environments

- Buzzword Bingo. Mine is Bigger than Yours

- 90's Mainframe Hosted Analytics
  - Decision Support. Information Center.   …
- 00's Unix. Large SMP. Big Shared Storage.
- 05's Unix. Grid. Rack. Big Shared Storage.
- 05's Unix. Rack. Shared Nothing Storage
- 2010   Cluster. Still Shared Nothing

Data Explosion



161 EB captured and replicated in 2006

281 EB captured and replicated in 2007

264 EB of available storage in 2007

1800 EB captured and replicated in 2011

IDC
Analyse the Future

Data at Discover Financial Services

After

- Central Database / EDW
  - Per Analyst Sandboxes
- Grid for Compute
  - Multiple Analysts
  - Restart / Sequencing
  - Performance Groups

Large SMP box
at capacity

Expensive SAN Storage
hard to manage

Before

Source: *Teradata Partners Conference*, 2010

Data at eBay

- Enterprise Data Warehouse
  - 500 – 5000 users
  - High concurrency (1000s active queries per minute slice)
- Singularity
  - 50 – 500 users
  - Modest concurrency (100s active queries per minute slice)
  - ~2x more data than EDW
- Hadoop Cluster
  - 5 – 10 users
  - Low concurrency
  - CPU “rich”
  - ?x more data than Singularity

Source: O Ratzesberger.
Teradata Partners Conference, 2010

Hard Projection – to – Soft Projection

Platform	Projection	Description
EDW	Hard	• Traditional Data Models • “Managed Data Warehouse” • Optimized Placement • Optimized Indexes
Singularity	In Between	• Some Traditional Columns • A “blob” organized as name-value pairs
Hadoop Cluster	Soft	• Data stored as text/binary files “as you received them” • Interesting Fields projected from files during initial map phase

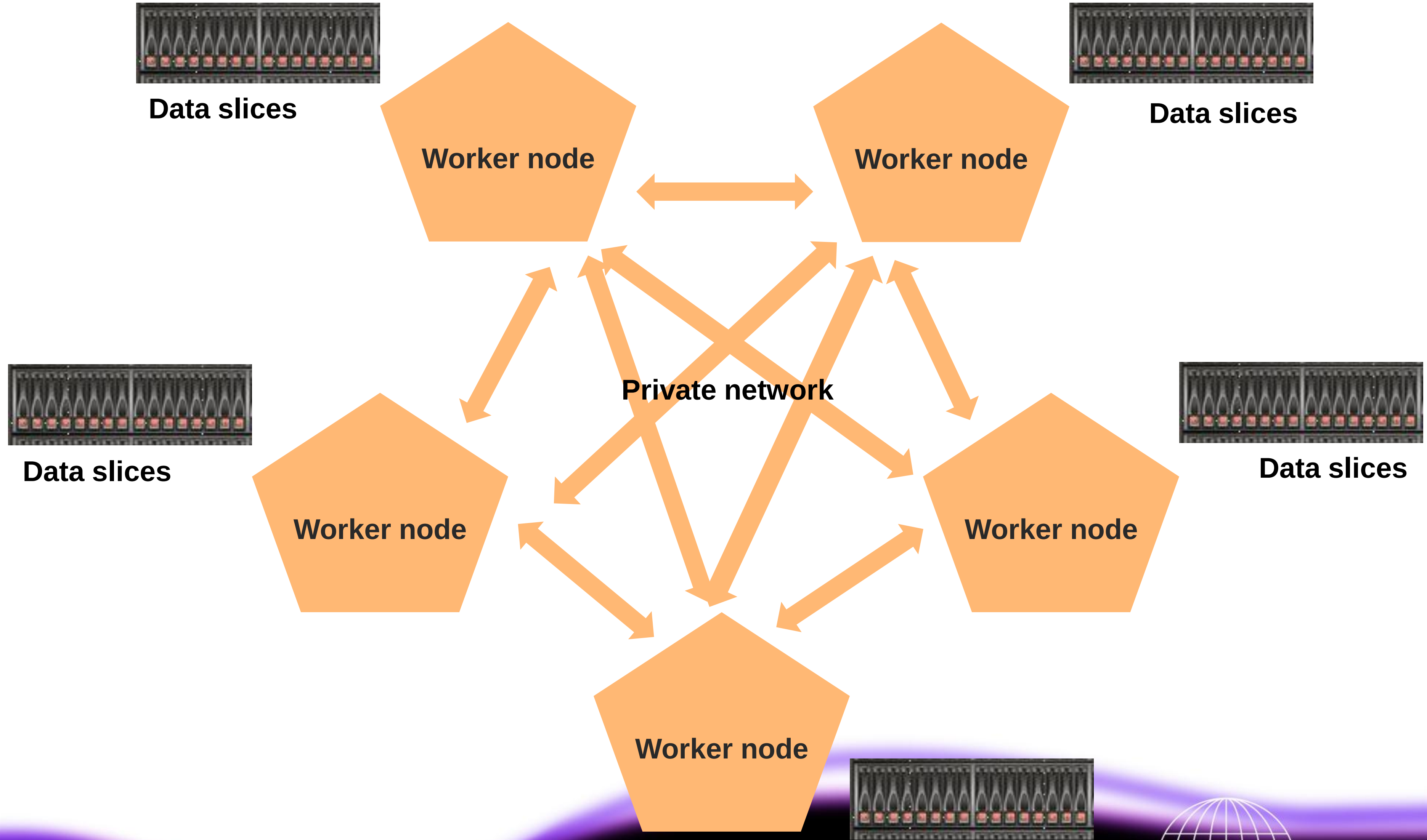
Source: O Ratzesberger.
Teradata Partners Conference, 2010

How Do You Respond to the Data Explosion

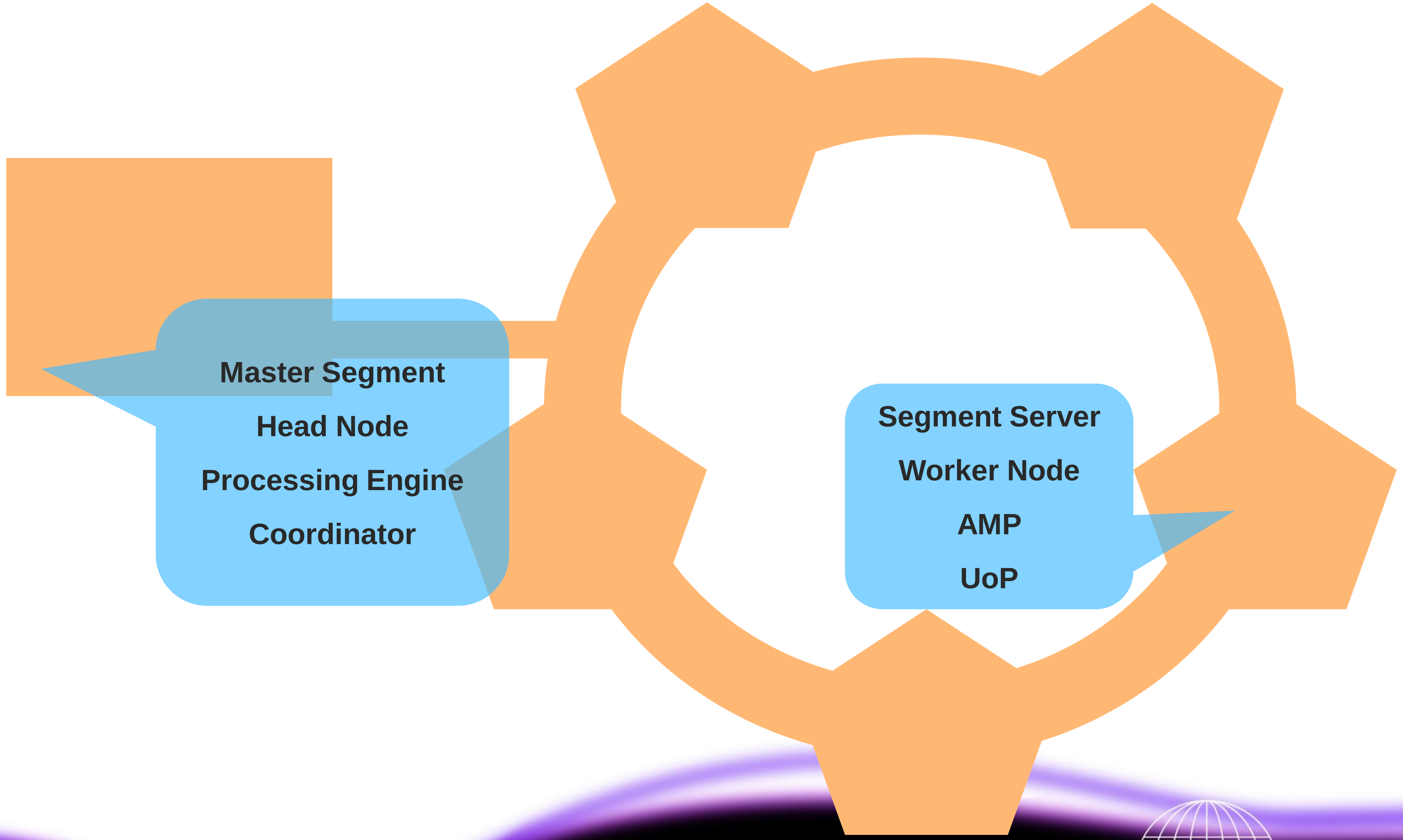
- ~~Get more storage~~
- Store in multiple places
  - Distributed databases
  - MPP architecture allows parallel processing
- Classic shared nothing MPP for DBMS
  - A set of CPU bricks (blades!!!)
  - A matching set of disks
  - A private network

The Usual Suspects

Database MPP – Simplified View



Database MPP – Simplified View



Database MPP – Simplified View

Partitioned Data

"Divide and Conquer"

Redundancy :: Resiliency

Comparison: Blade Server HPC and MPP DBMS

- Homogeneous hardware
  - At least at the worker node level
  - Nodes are multi-socket, multi-core platforms
  - 64-bit x86 Linux operating systems
- Central point of contact
  - Head node, master segment, queen, etc. for DBMS
  - Dedicated or arbitrarily chosen root node for HPC
- Focus slightly different
  - MPP DBMS: more disk, less CPU
  - HPC blades: less disk, more CPU

Look Ahead

- How can we engage a MPP DBMS environment for SAS high-performance analytics?
- Which analytic problems can be solved and how?
- How do we resolve tension between demand for CPU and disk

Contents

Hardware Story

- Multi-socket, multi-core
- Commodity blades

Data Story

- Data explosion
- MPP databases

SAS High-Performance Computing

- Analytic Tiers
- Matching data and compute style

Analytics—Latest Victim of Buzzword Craze

- Hal Varian, Google's Chief Economist:
  "I keep saying that the sexy job in the next 10 years will be statisticians. And I'm not kidding."

- 4 out of 3 people using the word "Analytics" are bad with fractions.
- Deep Analytics! (as compared to what, shallow analytics?)
  - "In practical terms, it [Deep Analytics] is the process of creating an analysis application for a power user specifically for them to do Deep Analytics."

Analytical Tiers

Tier	Examples	Class	SAS Procedures
Hindsight	Descriptive statistics, summarization		SUMMARY, MEANS, RANK, UNIVARIATE
	Cross-tabulation		FREQ
	Reporting		REPORT, TABULATE

Analytical Tiers

Tier	Examples	Class	SAS Procedures
Hindsight	Descriptive statistics, summarization		SUMMARY, MEANS, RANK, UNIVARIATE
	Cross-tabulation		FREQ
	Reporting		REPORT, TABULATE
Insight—descriptive modeling	Correlation analysis Variable clustering Factor analysis Principal component analysis	Relationships among variables	REG, CORR, VARCLUS FACTOR PRINCOMP

Analytical Tiers

Tier	Examples	Class	SAS Procedures
Hindsight	Descriptive statistics, summarization		SUMMARY, MEANS, RANK, UNIVARIATE
	Cross-tabulation		FREQ
	Reporting		REPORT, TABULATE
Insight—descriptive modeling	Correlation analysis Variable clustering Factor analysis Principal component analysis	Relationships among variables	REG, CORR, VARCLUS FACTOR PRINCOMP
Foresight—predictive modeling	Linear models Generalized linear models	Linear elements	REG, GLM, GLMSELECT LOGISTIC, GENMOD
	Nonlinear least-squares and maximum likelihood	Nonlinear elements	NLIN, NLP, MODEL
	Neural networks		NEURAL
	Linear mixed models Generalized linear mxed models Nonlinear mixed models	Random effects	MIXED GLIMMIX NLMIXED
	Decision methods		ARBOR, ARBORETUM
Optimization	Optimization		OPTMODEL, MO, MILP

High-End Analytics

- Much more than slice + dice reporting
- Often requires
  - access to all the data to train quality models
  - multiple passes through the data, even for non-iterative techniques
- Analytical techniques are varied
  - Customers resort to less appropriate methods to get "some results"/"any results" with big data
  - Real cases:
    - » 300 GB of input data for predictive modeling
    - » 1 billion records in a logistic regression
    - » 57 GB of scores from a single proc run

Contents

The Most Important Acceleration Strategies

- Co-location (of data and analytics)
- Co-location (of data and analytics)
- Co-location (of data and analytics)
- Avoid the disk, use memory
- Parallelize
- But, co-location
  - has many technological solutions
  - has to be done right
  - has to adjust to the complexity of the analytic task

Acceleration Strategies With Distributed DBMS

- Customers want to improve response times to SAS workload that accesses data inside DBMS

- What are the options
  - Re-state the work as SQL, let DBMS parallelize — SQL-PassThru
  - Extend SQL with UDFs — Inside-DB
  - Go beyond the simple (obvious) transforms — Inside-DB
  - Put SAS CPUs closer to DBMS CPUs — Alongside-DB

Analytical Tiers and Acceleration

Tier	Examples	Acceleration	SAS Procedures
Hindsight	Descriptive statistics, summarization	SQL-Passthru Inside-DB	SUMMARY, MEANS, RANK, UNIVARIATE
	Cross-tabulation	SQL-Passthru Inside-DB	FREQ
	Reporting	SQL-Passthru Inside-DB	REPORT, TABULATE
Insight—descriptive modeling	Correlation analysis Variable clustering Factor analysis Principal component analysis	Inside-DB possible, but not convenient	REG, CORR, VARCLUS FACTOR PRINCOMP
Foresight—predictive modeling	Linear models Generalized linear models	?	REG, GLM, GLMSELECT LOGISTIC, GENMOD
	Nonlinear least-squares and maximum likelihood	?	NLIN, NLP, MODEL
	Neural networks	?	NEURAL
	Linear mixed models Generalized linear mxed models Nonlinear mixed models	?	MIXED GLIMMIX NLMIXED
	Decision methods	?	ARBOR, ARBORETUM
Optimization	Optimization	?	OPTMODEL, MO, MILP

Inside-DB Barriers in High-End Tier

- Multi-pass methods
  - Only first pass should hit disk, keep data memory resident afterwards
  - Even ostensibly simple problems might require more than one pass (interaction effects with classification variables)
- Iterative methods
- Classification factors
- Dynamically generated code
- State preservation between passes

Inside-DB and MapReduce

- “What we’ve got here is a failure to communicate”
- DBMS communicates among nodes
  - If this requires node-to-node com

```
CREATE TABLE example(
    id        varchar(3),
    name      varchar(2),
    value1    float8    ,
    value2    float8    ) DISTRIBUTED BY (id);
INSERT INTO example SELECT * from ... ;
```

  - Why would **we** try to do **this** without?

```
proc hplogistic data=GPLib.MyTable;
   class A B C D ;
   model y = a b c b*d x1-x100;
   performance details;
   id a b;
   output out=gplib.logout pred=p xbeta=xb;
run;
```

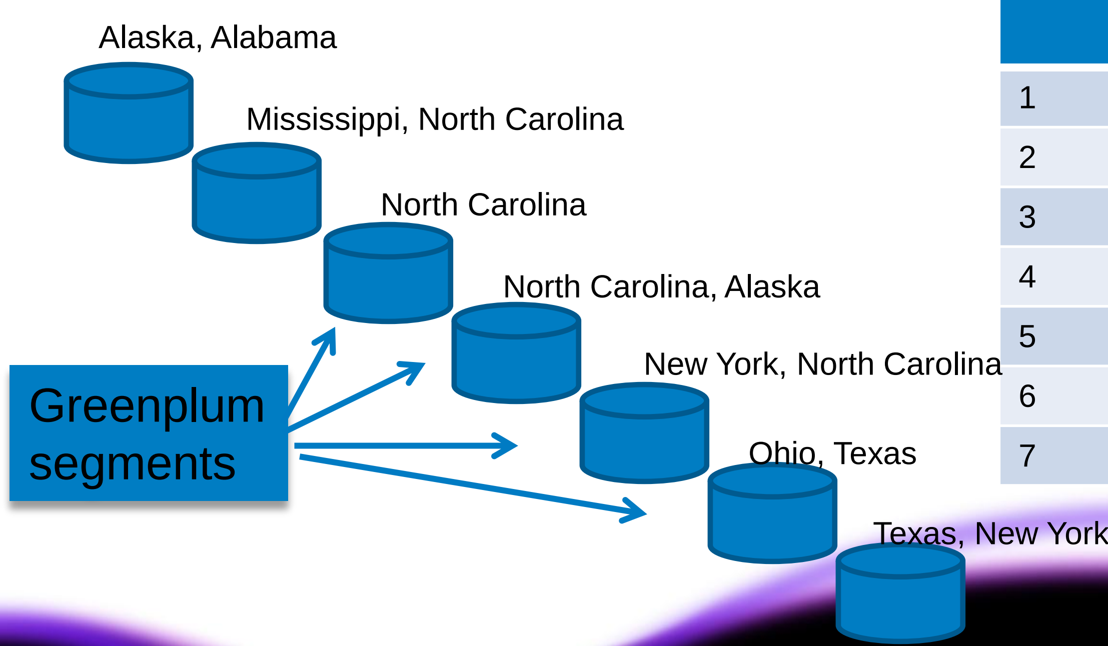
Architecture Details (why does it matter)

```
proc hpreg data=Greenplum.MyTable;
   class state age_group;
   model y = A B state*age_group x1-x100;
   selection stepwise;
run;
```

```
    Class Level Information

   Class      Levels      Values
   state           7      Alabama  Alaska
                          Mississippi  New York
                          North Carolina
                          Ohio  Texas
```

Table partitioned by zip code

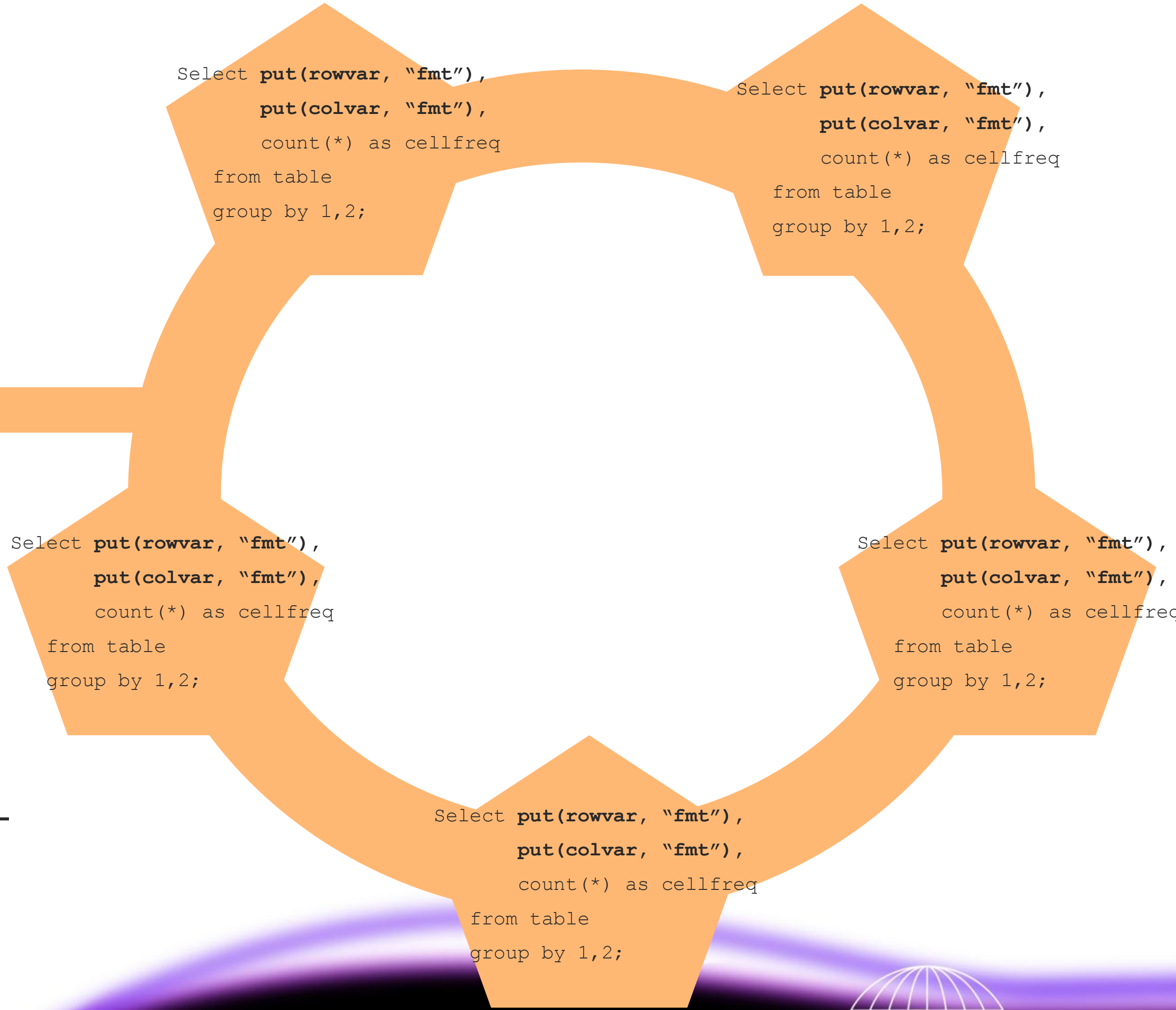


Segment	AL (1)	AK (2)	MS (3)	NY (4)	NC (5)	OH (6)	TX (7)
1	2	1					
2			1		2		
3					1		
4		2			1		
5				1	2		
6						1	2
7				2			1

Order-dependent local encoding must be mapped to correct global encoding; requires communication b/w segments

Strategy #1: Send SQL to Worker Nodes

```
Select put(rowvar, "fmt"),
       put(colvar, "fmt"),
       count(*) as cellfreq
   from table
   group by 1,2;
```

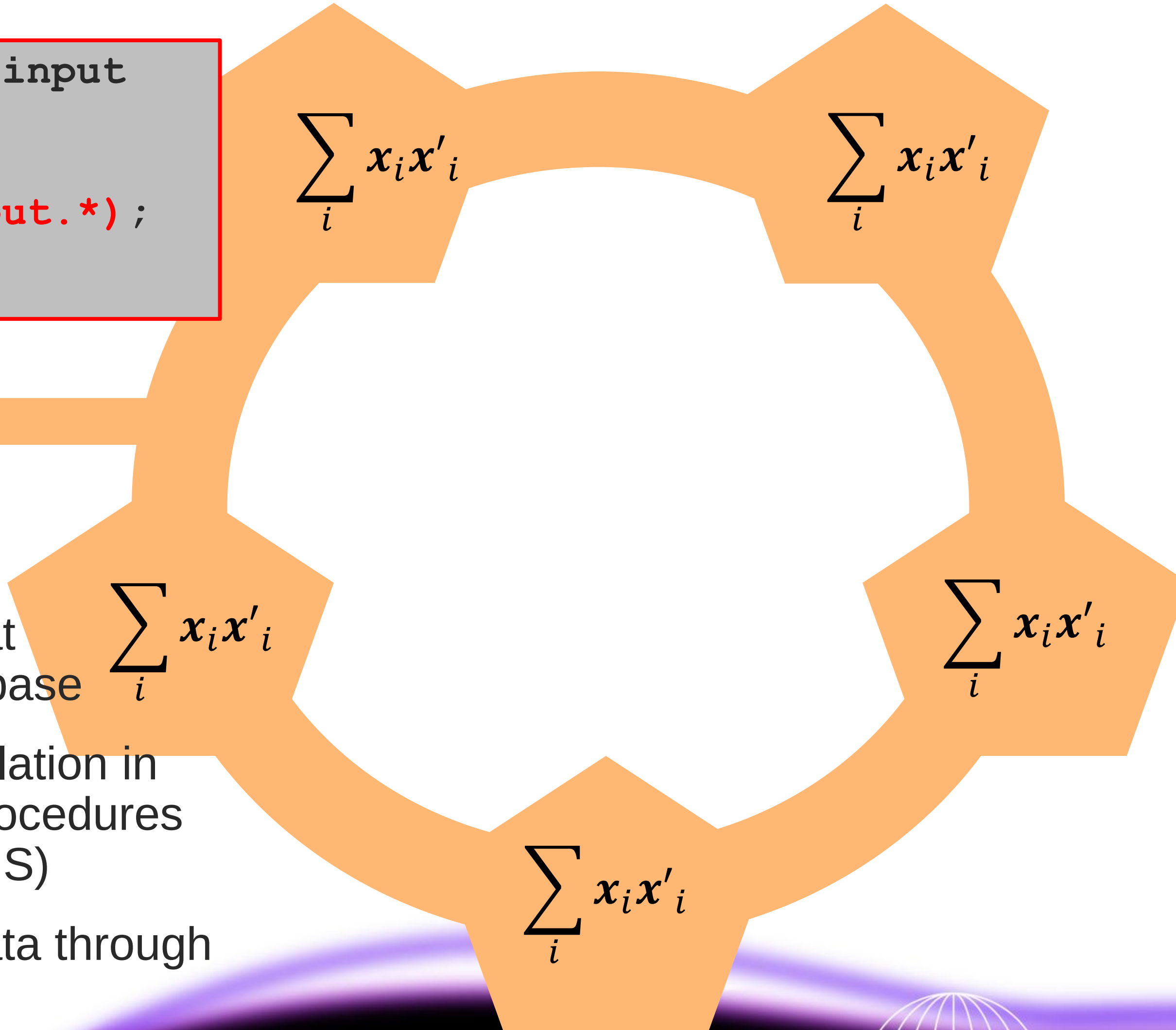


Co-location strategy:

- Math expressed through SQL
- SQL executes in parallel

Strategy #2: Send Math to worker nodes

```
With select * from T as input
Select *
  from SAS_SSCP_PREP(input.*);
```



Move up-front calculation that range over all data into database

Early success with **X'X** calculation in many classical regression procedures (REG, PRINCOMP, VARCLUS)

Co-location: bring math to data through UDF

Strategy #2: Disadvantages

- Local Partition Data Scanned only once.
  - No “do until tolerance < xxx”
- Local Results returned without any look at other Local Results
  - Similar to Hadoop – Map stages have no access to other instances of Map running else where
- Table UDF has constraints on
  - Number of columns
  - Result-set signature (column definitions)
- Math Logic ran under DBMS “terms & conditions”

New Strategy: Alongside-the-Database

- Math processes running as peers of DBMS
- Uses same hardware as the DBMS
- Pops on/off dynamically (on demand)
- Co-location model: ~~move~~ pass data to analytic process
- Math processes can communicate
  (MPI, Message Passing Interface)

Alongside-dB: Benefits

- Database HPC Appliance provides
  - World-class MPP database
  - High-performance ETL/ELT/ETLT
  - Replication and failover
  - High-performance analytic environment
- SAS in-Memory Analytics provides
  - High-end high-performance analytics
  - Familiar user interface (MVA SAS)
  - Distributed read/write into/from local memory

Alongside-dB: Talking Points

- Getting data to the analytic process is only half of the story. You also need
  - analytic software that can make use of partial data
  - SMP/MPP software that takes advantage of multi-core platforms
  - software that knows when and how much to communicate between units to enable high-end analytics
- The MPP database minimizes communications and uses known points in the query process
- High-end distributed analytics do the same

SAS High-Performance Analytics SAS/HPA

▪ HPREG	linear regression and variable selection
▪ HPLOGISTIC	logistic regression and variable selection
▪ HPLMIXED	linear mixed models
▪ HPNEURAL	neural nets
▪ HPNLIN	nonlinear regression and maximum likelihood
▪ HPREDUCE	covariance/correlation analysis, variable reduction
▪ HPDMDB	summarization
▪ HPSUMMARY	descriptive statistics
▪ HPFOREST	predictive modeling based on decision trees
▪ HPDS2	next-generation data step

SAS Procedures Then and Now

```
proc logistic data=TD.mydata;
   class A B C;
   model y(event='1') = A B B*C;
run;
```

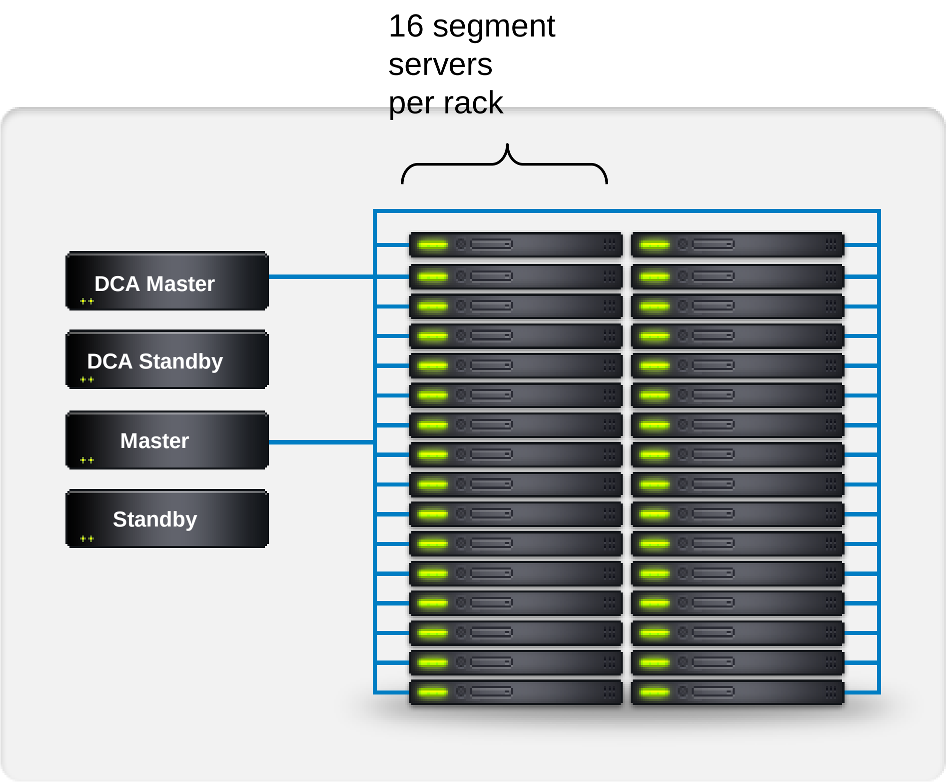
```
proc hplogistic data=TD.mydata;
   class A B C;
   model y(event='1') = A B B*C;
run;
```

Single-threaded	Multi-threaded
Not aware of distributed computing environment	Aware of distributed computing environment
SAS/ACCESS for data read	SAS/ACCESS for parsing support
Runs on client	Runs on client or DBMS appliance
Brings distributed data to client	Runs alongside distributed data source
Large I/O	In-Memory Analytics

Platform

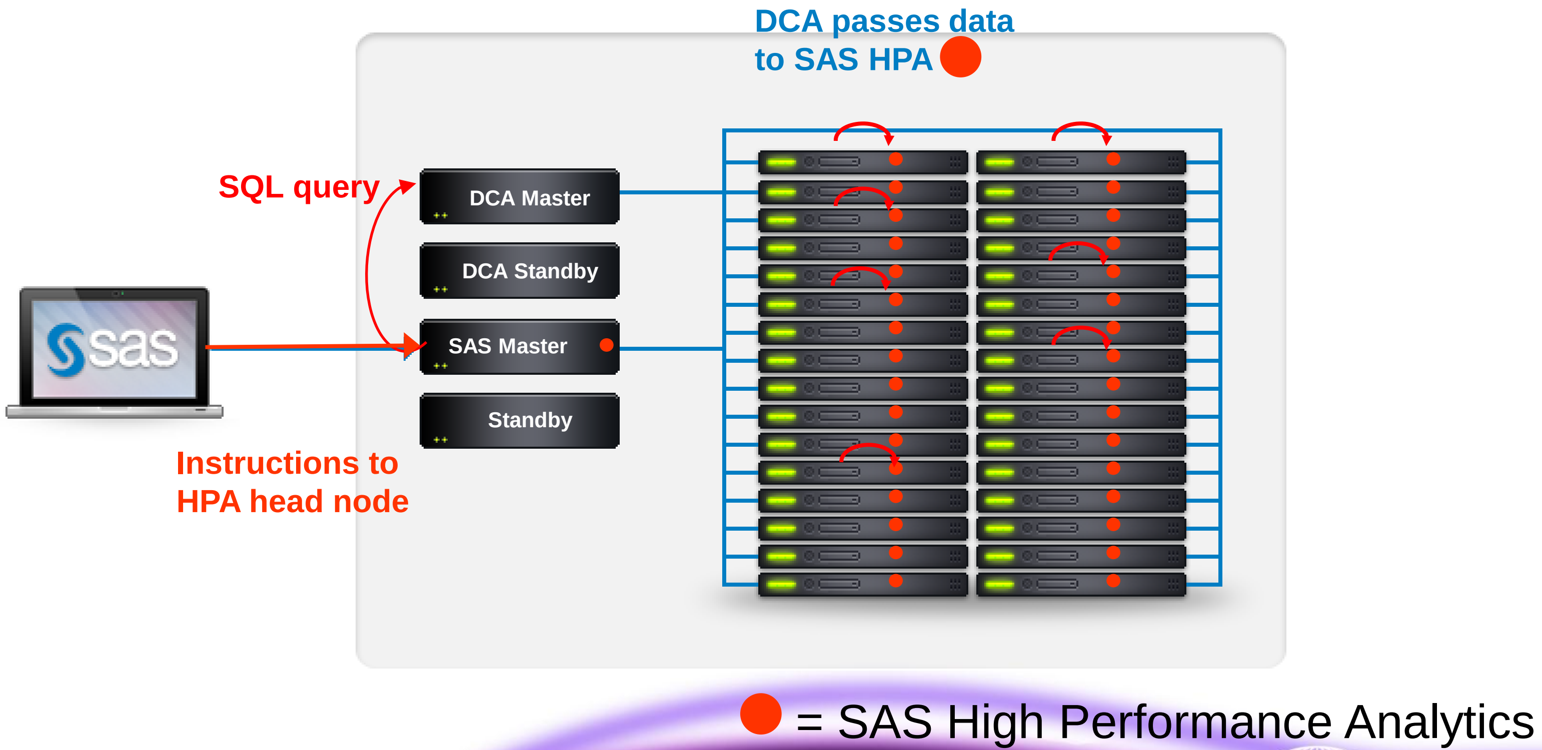
- EMC Greenplum and Teradata analytic appliances
- Provides
  - MPP database
  - MPP computing environment
- Client-side operation from standard SAS session

Two Rack EMC/Greenplum DCA



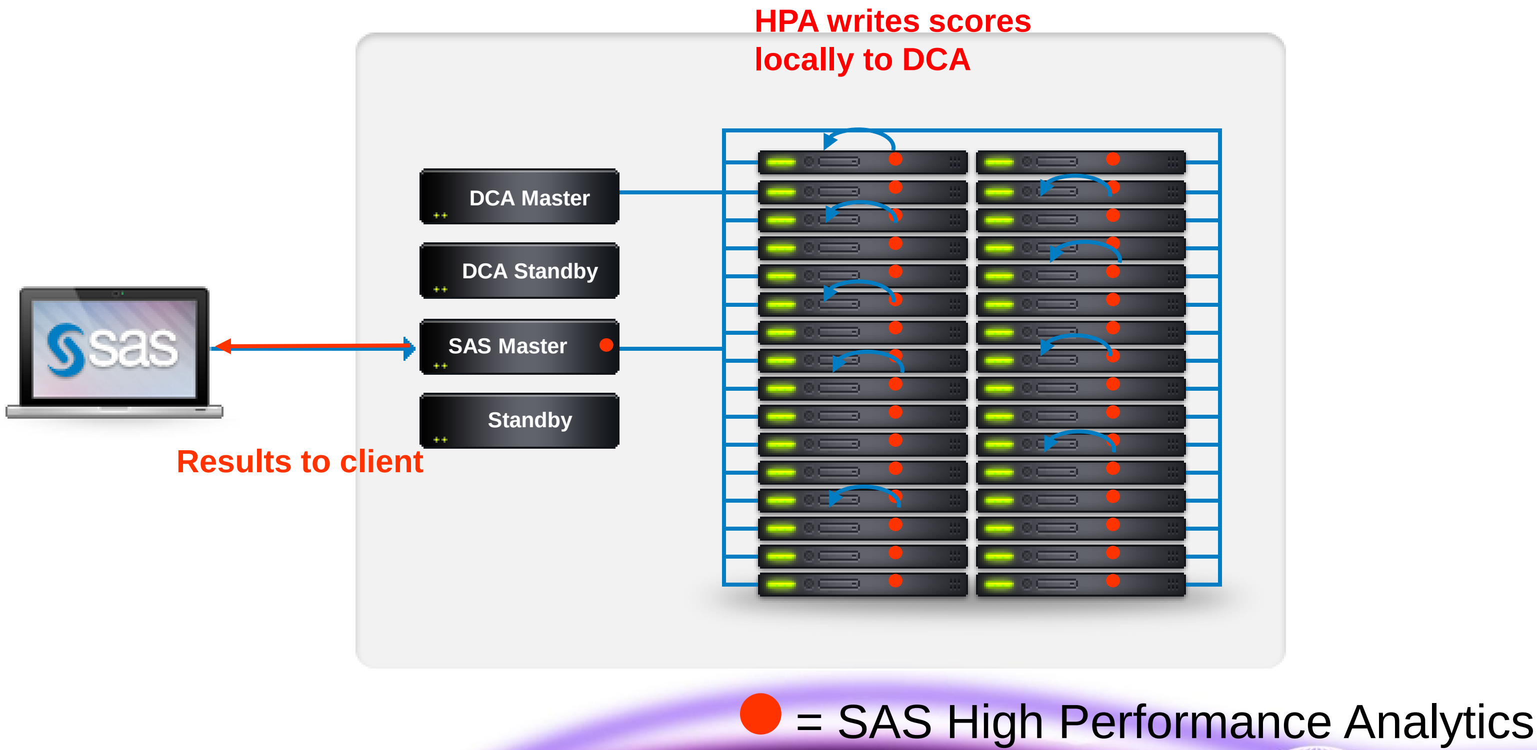
SAS/HPA Alongside-Greenplum

```
proc hplogistic data=GPLib.MyTable;
   class A B C D ;
   model y = a b c b*d x1-x100;
   output out=GPlib.logout pred=p;
run;
```

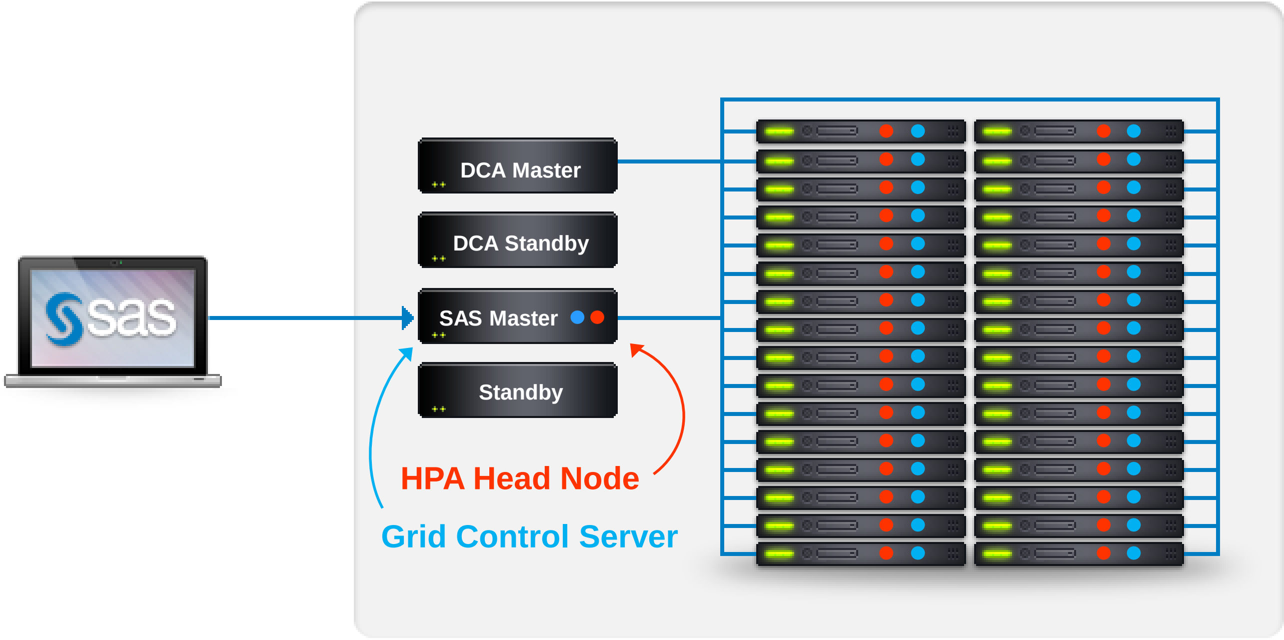


SAS/HPA Alongside-Greenplum

```
proc hplogistic data=GPLib.MyTable;
   class A B C D ;
   model y = a b c b*d x1-x100;
   output out=GPlib.logout pred=p;
run;
```



SAS Grid Manager and SAS/HPA Alongside-Greenplum



= SAS High Performance Analytics
= SAS Grid Manager

Contents

Hardware Story

- Multi-socket, multi-core
- Commodity blades

Data Story

- Data explosion
- MPP databases

SAS High-Performance Computing

- Analytic Tiers
- Matching data and compute style

Where do we go from here?

Where Do We Go From Here

- SAS High-Performance Analytics
  - major platform for proc-based high-performance computing
  - alongside-the-database model
  - more solutions built on top of SAS/HPA
- More SAS high-performance solutions
- Integration of in-database, alongside-db approaches
- DS2 as next-generation data step language
  - Executes inside and alongside the database
  - "DS2, the last language you will ever learn"
- Asymmetric architectures have potential

Strategy #3: Math Processes as Peers to Other Storage

Map phase extracts fields

send to "localhost"

```
Top:

Extract fields

Send to
    localhost

goto top;
```

```
proc hplogistic data=_ds_;
   class A B C;
   model Y = A B B*C x;
run;
```

Strategy #3 :: Math Processes as Peers to Other Storage

```
With select * from T as
   inp

select *

   from ECHO_PEER(inp.*);
```

SAS® GLOBAL FORUM

2011 Las Vegas Nevada

Thank You

#SASGF11