Part 3

SAS High Performance Computing with a DBMS

What will we learn?

- Understand the concepts and trend of SAS high performance computing (HPC).
- Integrate SAS with a DBMS.
- Introduce and exercise some available SAS HPC approaches.

Why do we need (SAS) HPC?

- New challenges in data processing industry.
- People's new and ever-changing expectations.
- Limitations of non-HPC computation.

The Challenges Facing Data Processing

- Increasing volumes and varieties of data (so-called “Big Data”). Exploding data volumes hinder the completion of key analytic processes in a timely manner.
- Excessive data movement and unnecessary data proliferation. Organizations struggle to determine what data should be stored where and for how long, what data should be used in analytical processing and how it should be prepared for analysis.
- Overwhelmed and poorly deployed IT resources. More requests for analytical processing mean longer waits for answers and unpredictable response times.
- Analytical processing complexities. The growing number and complexity of analytical models and frequent data updates require an on-demand pool of distributed and parallel processing resources. Otherwise, it simply takes too long to get results.

Higher People's Expectations

- Immediately capture value and gain competitive advantages by exploiting big data, including existing and new data collected from other sources.
- Achieve fast response times to identify optimal actions and make the best decisions based on a big amount of data.
- Use more granular-level data and more complex analytical algorithms to produce new insights quickly, solve the most complex problems, act confidently to seize new opportunities and better manage risks.
- Quickly meet ever-changing business demands with flexible and dynamic workload balancing and high availability.
- Ensure data quality, improve data governance and enhance resource use (by reducing data movement and redundancy).
- Grow and optimize IT infrastructures (in a cost-effective manner).

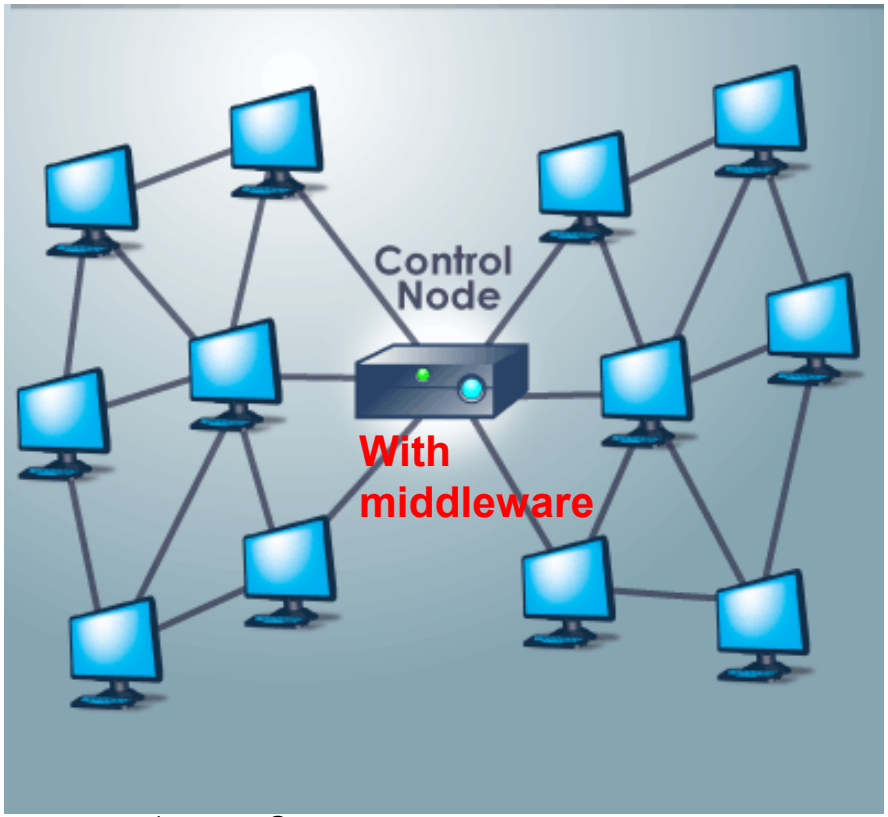
SAS HPC methods are right tools for dealing with the challenges and for achieving these expectations

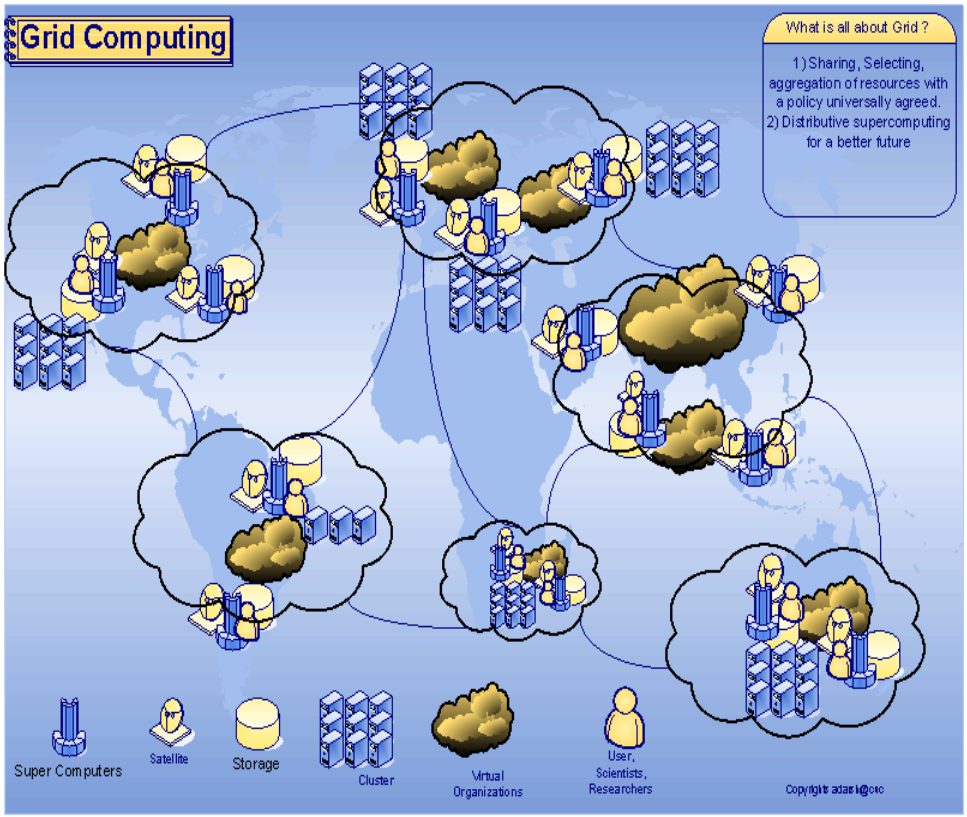
Understand the Concepts and Trend of SAS HPC

- SAS grid computing
- SAS in-database processing
- SAS in-memory analytics

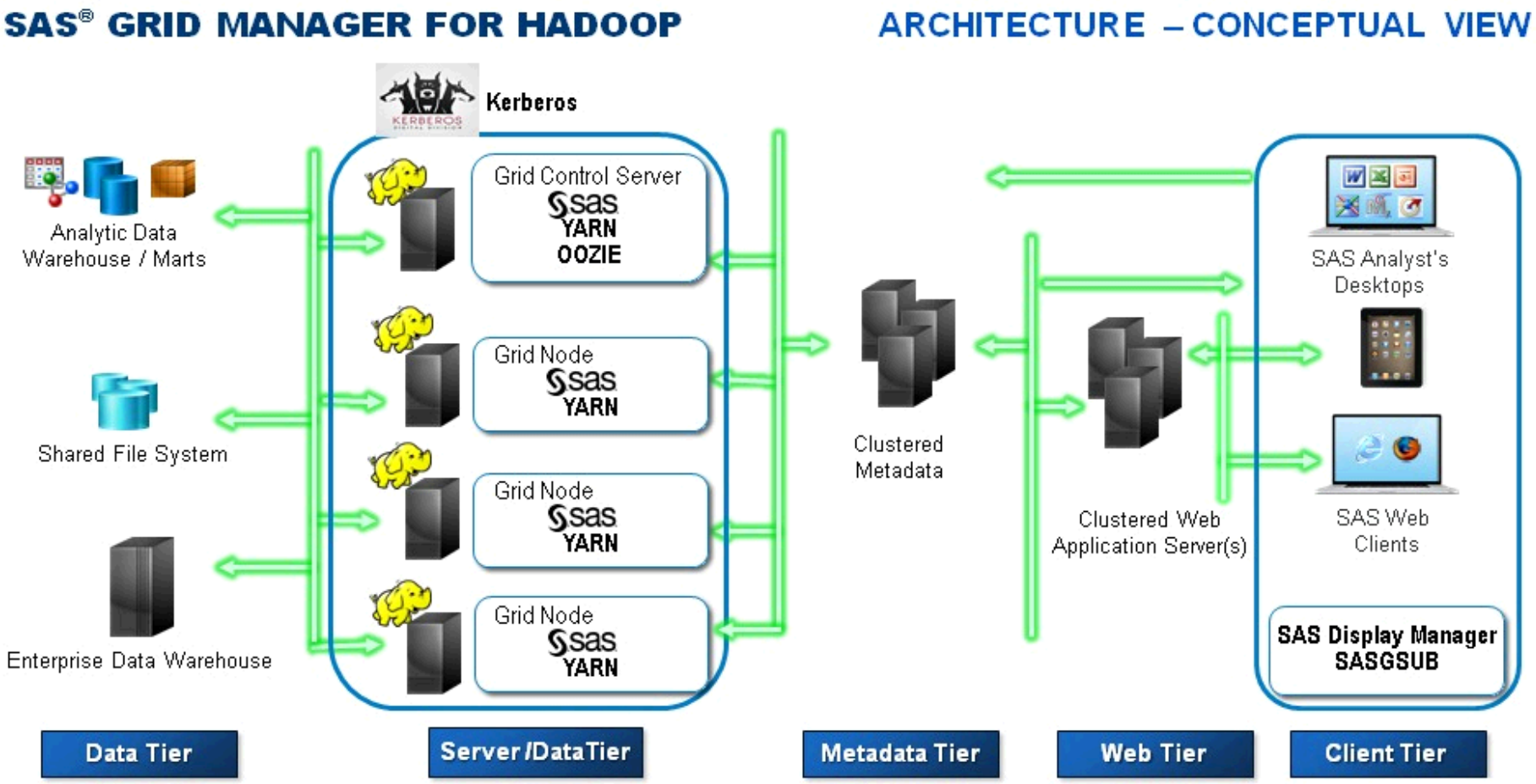
SAS Grid Computing: Basic Concepts

Grid computing is based on a parallel and distributed computer system in which resources are shared and integrated by common rules. At its most basic level, grid computing is a computer network in which each computer's resources are shared with every other computer in the system. Processing power, memory and data storage are all community resources that authorized users can tap into and leverage for specific tasks. A grid computing system can be as simple as a collection of similar computers running on the same operating system or as complex as internetworked systems comprised of every computer platform you can think of.



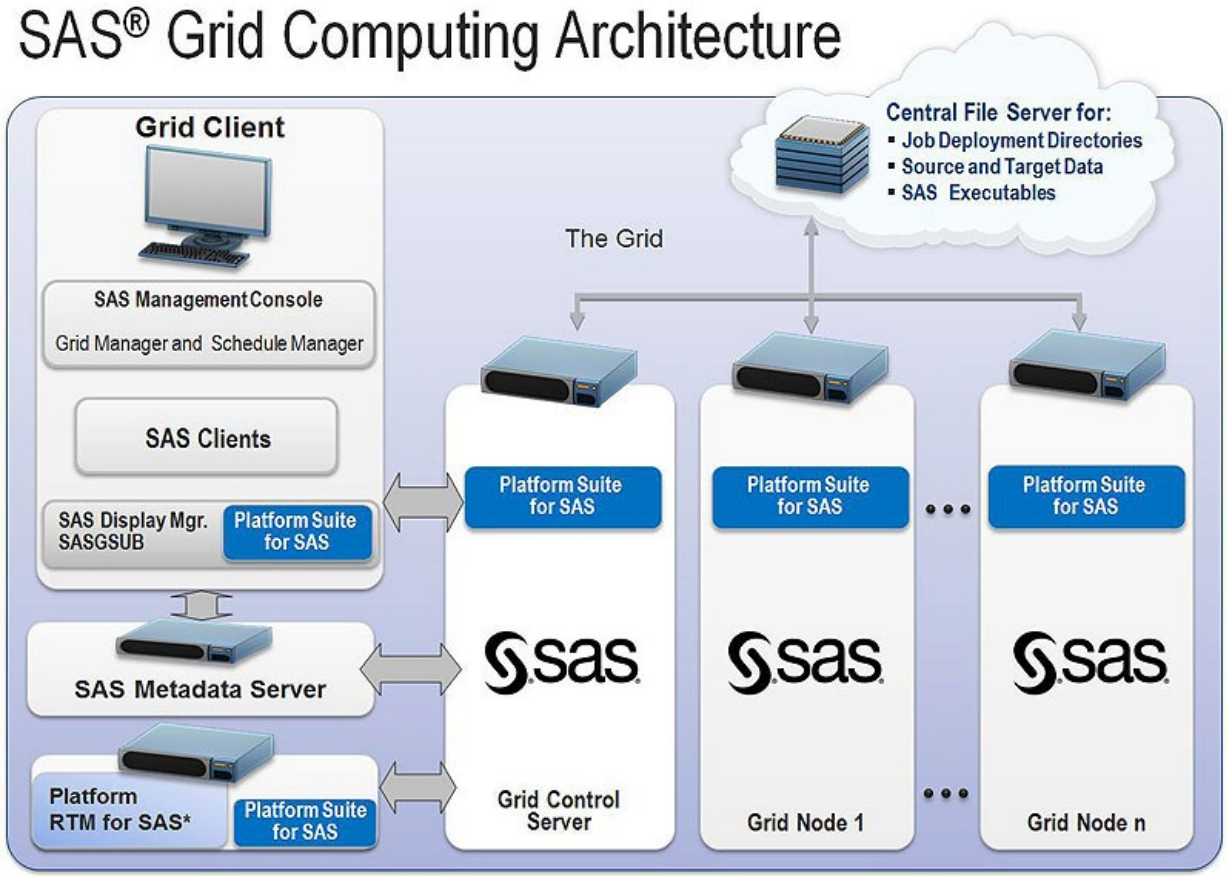


SAS Grid Manager for Hadoop



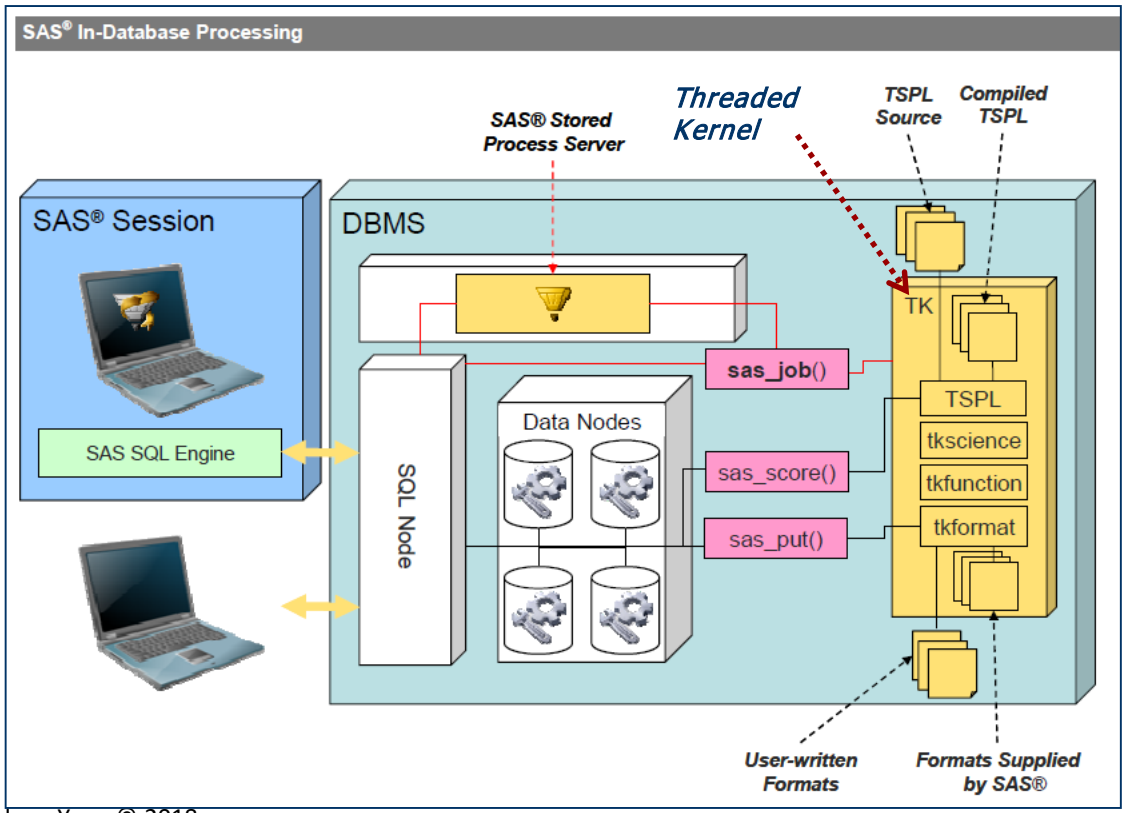
SAS Grid Computing

It allows individual SAS jobs to be split up, with each piece running in parallel across multiple machines in the grid environment using shared physical storage and enables organizations to create a managed, shared environment to process large volumes of data and analytic programs. This makes it a perfect solution for managing multiple SAS users and jobs while enabling efficient use of IT resources and lower-cost commodity hardware.



SAS In-Database Processing

It integrates select SAS technologies into your databases or data warehouses and utilizes the massively parallel processing (MPP) architecture of the database or data warehouse for scalability and better performance. You can run scoring models, some SAS procedures inside a database. This method, by moving relevant data integration, analytics and reporting tasks to where the data resides, reduces unnecessary data movement, promotes better data governance and provides faster results.



The SAS company and the database vendor will have to work together to achieve this goal. For example, SAS has been working with Oracle to enhance this capability.

In-Database Processing Overview

In-Database Feature	Software Required	DBMSs Supported
format publishing and the SAS_PUT() function	• Base SAS • SAS/ACCESS Interface to the DBMS	DB2 under UNIX Netezza Teradata Aster
scoring models	• Base SAS (and SAS EM) • SAS/ACCESS Interface to the DBMS • SAS Scoring Accelerator • SAS Model Manager (optional)	Aster *n*Cluster DB2 under UNIX Greenplum Netezza Teradata Oracle, Hadoop, Aster
Base SAS procedures: FREQ RANK REPORT SORT SUMMARY/MEANS TABULATE	• Base SAS • SAS/ACCESS Interface to the DBMS	DB2 under UNIX and PC Hosts Oracle Netezza Teradata Aster, Hadoop
SAS/STAT procedures: CORR CANCORR DMDB DMINE DMREG FACTOR PRINCOMP REG SCORE TIMESERIES VARCLUS	• Base SAS (for CORR) • SAS/ACCESS Interface to Teradata • SAS/STAT (for CANCORR, FACTOR,PRINCOMP, REG, SCORE, VARCLUS) • SAS/ETS (for TIMESERIES) • SAS Enterprise Miner (for DMDB, DMINE, DMREG) • SAS Analytics Accelerator	Teradata

SAS In-Memory Analytics

- It allows complex data exploration, model development and deployment steps to be processed in-memory and distributed in parallel across a dedicated set of nodes. New computational requests can be handled much faster and with better response times because data can be quickly pulled within/into the memory.
- It has three components:
  - SAS High-Performance Analytics (only available in Greenplum and Teradata)
  - SAS High-Performance Markdown Optimization
  - SAS High-Performance Risk

Capabilities and Benefits of the Three SAS HPC Methods

	Capability	Benefit
Grid computing	Workload balancing and management	Manage jobs and users more efficiently
	High availability of resources	Avoid user or source disruption
	Distributed processing	Enhance performance
	Commodity hardware use	Reduce cost
In-database processing	Data and analytic functions processed inside the database	Achieve better data governance
	Streamlined model development and analytical lifecycle deployment	Gain faster time-to-results
	Use of existing database architecture	Maximize your IT infrastructure
	Existing code can be run without modifications	Improve analytic efficiency and productivity
In-memory analytics	In-memory architecture for data and analytic processing	Solve your most complex problems in near-real time
	High-performance analytic capabilities within select SAS products and solutions	Derive highly accurate results through improved modeling
	Database appliances used for persistent storage and failover	Overcome service-level constraints

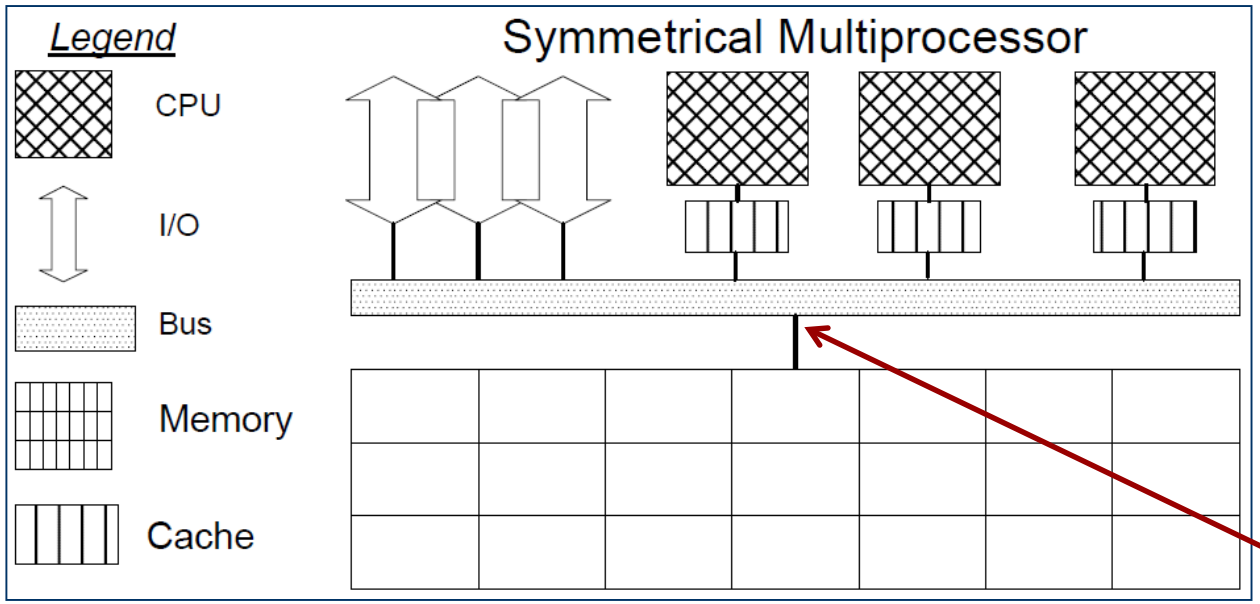
SAS Parallel Computing

- The term ***parallel*** in the computing context refers to simultaneous or concurrent execution—individual tasks being done at the same time.
- Since SAS 9.1, SAS has introduced parallel computing features that can dramatically reduce the clock-time-to-solution for jobs that are both CPU bound and/or I/O bound (clock-time-to-solution is how long a user has to wait to get his/her answer).
- It employs a strategy of Divide and Conquer. SAS takes the code that a programmer submits and attempts to split that task into small tasks, which can execute independently, and writes mini-programs called threads to execute the small-tasks.
- Many SAS PROCs implement threading, but to gain any benefit, the hardware must be able to execute multiple tasks independently. A rough recommendation for mini. hardware capable of taking advantage of SAS 9.1 (or later) features is: a machine with 4 independent CPUs, several Gigs of RAM (or more), an I/O channel for each CPU and no more than two disk drives per I/O channel.

Symmetrical Multiprocessor (SMP) Systems

The most widely used parallel hardware architecture today is the *symmetrical multiprocessor* (SMP) systems, an incredible advance over single CPU systems.

The most common way that SMP is used is in *multitasking*. While one CPU is executing one program, another CPU executes another. Because the operating system keeps the processes separate and invisible from each other, it appears to each program that it is running on a single-CPU sequential processing machine. This is what allows SAS to run on machines like Sun's multi-CPU Enterprise family of servers.

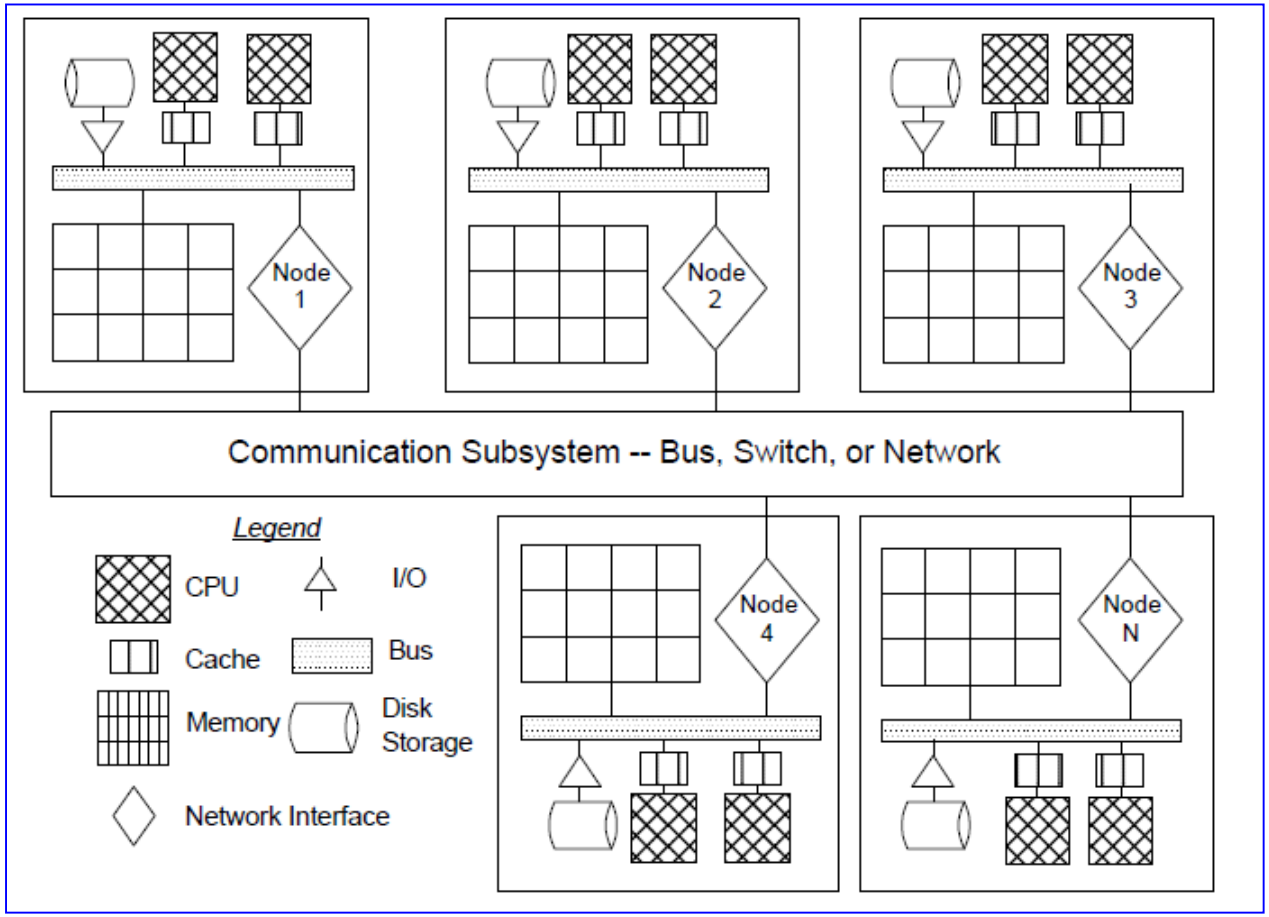


Multiple CPUs and associated resources running under a single operating system. Memory and disk resources are shared.

SMP has Bandwidth Limitations

Massively Parallel Processor (MPP) Systems

It also known as *distributed memory* systems. Unique instances of the operating systems and applications run on many physically separate *nodes*, each composed of a single or multiple processors and associated resources. Often no resources are shared between the nodes and communication between the nodes is done by passing messages between the nodes' operating systems.



MPP overcomes the limitations of SMP.

Integrating SAS with a DBMS

Understand how to communicate SAS with a standard database management system, such as Oracle, so that you can analyze the data stored in a standard DBMS with powerful SAS statistical analysis capabilities. This combination will bring you great power for analysis.

Integrating SAS with a DBMS

- We use Oracle as the DBMS to work with SAS to show how to communicate between these two systems.
- Oracle and other DBMSes have limited ability in data analysis, especially complicated data analysis.
- The SAS software provides an extensive suite of analytical tools, e.g., regression, analysis of variance (ANOVA), time series analysis, non-linear analysis, cluster analysis, etc.
- Together, the two products make a formidable combination.

What is Needed?

- Base SAS (minimum).
- Oracle (the Express version is OK).
- The SAS/Access software for Oracle.
- Oracle Client software is installed and properly configured on the client machine.
- Knowledge of SQL.
- Knowledge of SAS programming.

Methods for Accessing Oracle

- Using the "oracle" engine on the libname statement. This is a variation of the usual use of SAS Libraries. This method is generally called "Libname Oracle".
- Using "SQL Pass-Through" to connect directly to the Oracle instance.

The Libname Engine

The Oracle engine is specified:

libname *library_ref* **oracle** user=<'> *User_ID* <'>
password= <'> *Password* <'>
path= <'> *Oracle_Connection* <'> <*other_options*> ;

For example:
libname mydblib **oracle** user='SCOTT'
password='TIGER' path='OR92Rock' ;