Capítulo I

Antecedentes

Computación de Alto Rendimiento

La computación de Alto Rendimiento (HPC por sus siglas en ingles High Performance Computing) es un concepto que abarca un amplio espectro de la aún más extensa área de las Ciencias de la Computación, específicamente enfocado a brindar un sistema solido, claro, flexible y estructurado para garantizar el funcionamiento óptimo en la distribución de los recursos.

HPC esta relacionada con la “supercomputacion”, y surgio de los principios de este termino y trabajo, en ocaciones estos dos terminos son utilizados de manera indistinta. Sin embargo, el termino supercomputacion es usualmente aplicado a infraestructuras con propositos especificos, un subconjunto especifico de HPC. Las supercomputadoras suelen ser equipos mas poderosos y con sistemas especializados, y por lo tanto muy caros. Un ejemplo podria ser la supercomputadora Cray [*JE Beasley. Linear programming on Cray supercomputers. The Journal of the Operational Research Society. Vol. 41, No. 2, Feb., 1990*] o el sistema Thinking Machine [*Thinking Machines Connection Machine, Ultra High Speed Graphics Project-LANL*] de 1980. Sin embargo, estos sistemas de supercomputación especializados han sido casi completamente reemplazados por clusters de computadoras basicas no especializadas.

Las computadoras de CPU estandar estan basados en la arquitectura de Von Neumann [*John Backus. Can programming be liberated from the von Neumann style? Communications of the ACM, 21(8), 1977*], y basicamente cuentan con la restriccion de poder ejecutar unicamente una instrucción a la vez, una tras otra (ejemplo: procesamiento en serie). De esta manera, la unica opcion para poder realizar calculos mas veloces en este tipo de arquitectura era incrementando la velocidad con la que los calculos eran realizados. Aunque esta velocidad de calculo ha aumentado de forma constante y rapida desde la invencion del circuito integrado, el cual sigue la Ley de Moore [*Excerpts from A Conversation with Gordon Moore: Moore's Law. G Moore - Interview by Intel Corporation, Intel Corporation.*], muchos calculos deseables son todavia tan complejos y extensos que incluso en el equipo serial mas rapido, estos requiririan semanas o inclusive anos para finalizar el computo. Sin embargo, muchas aplicaciones pueden dividirse en segmentos y estos segmentos resolverse simultaneamente en varios equipos. Cuanto mas se puede dividir la aplicación, mayor sera la ejecucion en paralelo. Todas las aplicaciones HPC involucran la paralizacion y ejecucion del computo en multiples entornos distribuidos.

Computacion Paralela y Distribuida

La computacion paralela permite que varios calculos sean ejecutados simultaneamente. Cuando los problemas complejos pueden ser segmentados en otros menos complejos que puedan ser resueltos concurrentemente (al mismo tiempo o en paralelo), en consecuencia es reducido el tiempo necesario para resolver dicho problema. Un punto importante en la paralizacion de aplicaciones es la frecuencia en que los segmentos de codigo necesitan comunicarse entre si. Algunos calculos pueden ser divididos en segmentos que unicamente necesitan una comunicación parcial del resultado entre ellos e inclusive pueden no ser necesaria la comunicación [*Blaise Barney. Introduction to Parallel Computing.* [*https://computing.llnl.gov/tutorials/parallel\_comp/*](https://computing.llnl.gov/tutorials/parallel_comp/)] .

Cuando las subtareas necesitan comunicarse entre si para poder realizar el calculo en paralelo, un nuevo conjunto de cuestionamientos se introducen que complican la escritura de tales programas correctamente. Especificamente la necesidad de concurrencia y la sincronizacion introducen potenciales bugs (hace falta comentar que es un bug? O poner error nomas?), como podria ser la condicion de deadlocks (o poner bloqueos??) [*E. G. Coffman, M. Elphick, A. Shoshani. System Deadlocks. Journal ACM Computing Surveys (CSUR) Volume 3 Issue 2, June 1971*]. Estos problemas adicionales son el mayor obstaculo para la implementacion de computacion pararela y obtener una buena prestacion paralela de un programa. Otra consideracion para la computacion paralela es que la sobrecarga de la comunicación puede y hara que se pierda todo el rendimiento en la ejecucion simultanea de tareas. El incremento de velocidad de un programa es gobernado por la Ley de Amdahl [*G. M. Amdahl. Validity of the Single-Processor Approach to Achieving Large Scale Computing Capabilities. In AFIPS Conference Proceedings, pages 483–485, April 1967*], la cual que básicamente permite en teoría calcular la cantidad de aceleración que se puede obtener basada en la cantidad de cálculo que se puede en paralelo. En la actualidad, en donde los procesadores multinucleos ya son productos basicos, esta teoria requiere una mayor investigacion basandose en el rendimiento del chip completo en lugar de centrarse en la eficiencia de cada nucleo [*Mark D. Hill and Michael R. Marty. Amdahl’s Law in the Multicore Era. IEEE Computer 2008*].

Arquitectura de procesamiento en paralelo

Taxonomia para organizar las arquitecturas con el motivo de estudiarlas [*M. Flynn. Computer organizations and their e  
ectiveness. IEEE Trans-actions on Computers, September 1972*], esta se basa en las instrucciones de control de secuencias de datos dispuestas en la arquitectura. Está dividida en:

* SISD (del inglés Single Instruction and Single Data): Tomando como ejemplo la Arquitectura de Von Neumann, corresponde a una única instrucción operando con una única secuencia de datos.
* SIMD (del inglés Single Instruction and Multiple Data): Una misma instrucción aplicada a múltiples secuencias de datos.
* MISD (del inglés Multiple Instruction and Single Data): Múltiples Instrucciones simultaneas aplicados a una única secuencia de datos.
* MIMD (del inglés Multiple Instruction and Multiple Data): trata sobre las arquitecturas que permiten un múltiple acceso a secuencias de datos por instrucciones múltiples. Tratando de organizar esta última a finales de la década del 90 se procede a subdividirla en dos secciones [*Ralph Duncan. A Survey of Parallel Computer Architectures. IEEE Computer Society Press. Vol 23, Issue 2*.][*K. Hwang, F. Briggs. Computer Architecture and Parallel Processing. McGraw-Hill. 1994*]:
  + SMP (Symetric MultiProcesing): Arquitecturas que comparten una memoria física única y cuya velocidad disminuye según se incrementan los procesadores.
  + DMM (Distibuited Memory Multiprocessing): Son sistemas en las cuales el procesador posee su propia memoria, evitan problemas de bloqueos y evita el acceso a la memoria física principal.

En el ámbito de HPC, en la mayoria de los casos se trata de sistemas MIMD, aunque los sistemas SIMD, en forma de clusters de computadores basados en GPU [*Zhe Fan, Feng Qiu, Arie Kaufman, and Suzanne Yoakum-Stover. GPU cluster for high performance computing. Proceedings of the 2004 ACM/IEEE Supercomputing Conference (SC'04), page 47059, November 2004*], se están convirtiendo en un subconjunto significativo y muy prometedor en la labor en la computacion de alto rendimiento. Los sistemas MIMD pueden ser divididos en dos grupos principales, los que comparten una memoria principal (SMP), y aquellos que no comparten una memoria (DMM). Al momento de implementar una infraestructura de Cluster, usualmente se hace uso de ambos tipos de la arquitectura MIMD. Un procesador de memoria compartida, tambien llamado SMP (del ingles Symmetric multiprocessor), es aquel donde varios CPUs conviven en la misma maquina, un ejemplo de esto son los procesadores multinucleos.

HPC basado en el modelo de Cluster

Existen varias formas de presentar un modelo correspondiente al cómputo en altas prestaciones, como por ejemplo una estructura basada en el Modelo de Clúster. [agregar lo de hpcnotes waht is a cluster computer]

La utilización de estas infraestructuras (te referis a las altas prestaciones o al cluster?) en varias áreas investigativas, no necesariamente informáticas como (faltan los ejemplos), permiten que las Altas Prestaciones formen parte esencial en la solución a los problemas reales. Este trabajo trata sobre (consiste en) la situación de la Computación de Altas Prestaciones a nivel internacional en comparación con su uso en el territorio paraguayo, sus aplicaciones actuales y los requisitos básicos para posicionarlas como herramienta útil y necesaria a la hora de solucionar situaciones de aplicaciones cotidianas.

Middleware SSI para implementar HPC

Base de Datos

Altas Prestaciones en Paraguay

En la actualidad en Paraguay, la utilización del cluster goza de un panorama muy alentador, ya que infraestructuras basadas en Clúster HA, HR y/o HP son muy comúnmente utilizadas por empresas cuya necesidad de garantizar características propias de la computación de altas prestaciones cumple un factor más que importante en el optimo desempeño de sus funciones.

En el sector académico, instituciones como el Centro Nacional de Computación (CNC) provee infraestructuras de tipo clúster para trabajos de investigación, dichos trabajos no necesariamente se enfocan en el ámbito computacional, más bien se extiende a una interrelación con otras áreas académicas como son la electrónica, la genética, la agricultura entre otros.

(mencionar unos de estos ejemplos, al menos alguno donde se utilice esos cluster,hacer mención mas especifico del proyecto en que consiste y como funciona el cluster en ese caso.)

(También incluir aca una previsión de cómo se avanzaría o hacia donde se dirige todo esto de altas prestaciones en paraguay, si se va a exteder aun mas al sector privado, si otras organizaciones van a empezar a usar también, la manera en como podes hacer esto es tomando ejemplos de otros países que ya están mas avanzados y en base a eso predecir como probablemente va a cambiar en paraguay también..)

Bibliografia

Aquí iran numeradas todas las citas que ahora se encuentran insertadas en el texto anterior.