

CLUSTER COMPUTING

Oleh: Ahmad Syauqi Ahsan

PARALLEL COMPUTING

- ✘ Membagi permasalahan menjadi beberapa bagian dan kemudian diselesaikan secara bersamaan.
- ✘ Kenapa ?
 - + Keterbatasan fisik dari hardware komputer (kecepatan cahaya, kecepatan elektron, dll)
 - + Pertimbangan ekonomi: lebih kompleks = lebih mahal
 - + Keterbatasan performance: dua kali frekwensi <> dua kali performance
 - + Aplikasi besar membutuhkan terlalu banyak memory dan waktu penyelesaian
- ✘ Keuntungan
 - + Penambahan kecepatan proses dan optimalisasi penggunaan resources
- ✘ Kerugian
 - + Membutuhkan pemrograman yang kompleks → pengembangan aplikasi sulit

LEVEL OF PARALLELISM

× CPU

- + Multiple CPU
- + Multiple CPU Cores
- + Threads – time sharing

× Memory

- + Shared
- + Distributed
- + Hybrid (Virtual Shared Memory)

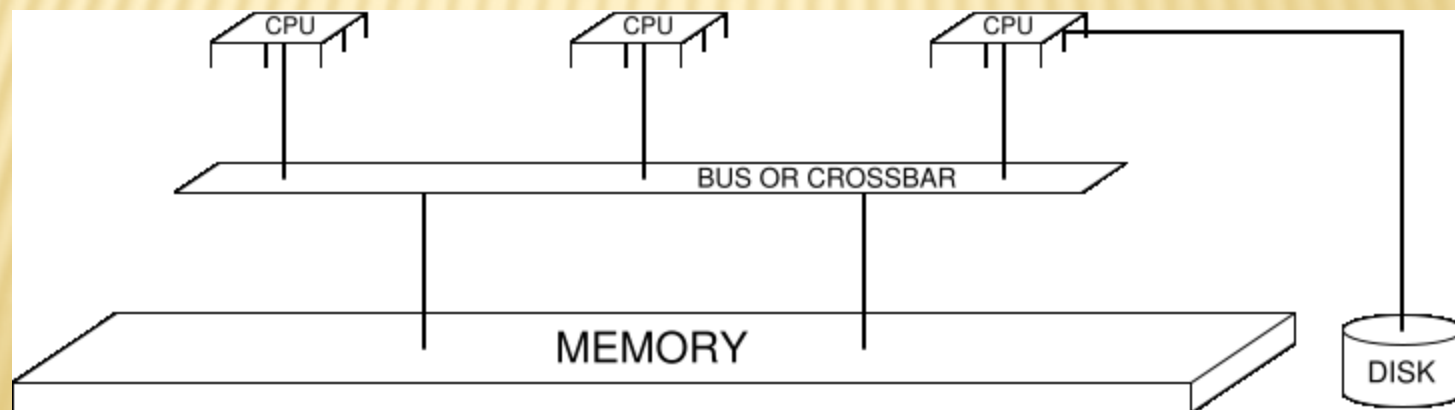
- *Parallel Computer bisa berupa satu PC dengan prosesor Intel Core2Duo atau suatu sistem yang kompleks seperti IBM Blue Gene*

PARALLEL ARCHITECTURES

- ✘ Symmetric Multiple Processing (SMP)
- ✘ Non Uniform Memory Access (NUMA)
- ✘ Massively Parallel Processors (MPP)
- ✘ Vector Machine
- ✘ Cluster Computing

SYMMETRIC MULTIPLE PROCESSING (SMP)

- ✘ Komputer dengan beberapa processor identik yang saling terhubung menggunakan *bus*.
- ✘ Menggunakan Shared Memory
- ✘ Karena keterbatasan dari kemampuan *bus*, kekurangan dari *SMP* ini adalah scalability.
- ✘ Contoh: SUN UltraSparc II, SGI MIPS, DEC Alpha, Itanium, Opteron, Xeon, Core2Duo, Athlon 64 X2, HP PA-RISC, dll.



NON UNIFORM MEMORY ACCESS (NUMA)

- ✘ Menyelesaikan permasalahan scalability pada arsitektur SMP.
- ✘ Model hybrid memory. Memory campuran antara shared memory dan distributed memory.
- ✘ Kecepatan akses memory tergantung dari jarak relatif antara memory dengan prosesor.
- ✘ Contoh: SGI Origin/Altix, Alpha GS, HP Superdome

MASSIVELY PARALLEL PROCESSORS (MPP)

- ✘ Satu komputer dengan banyak CPU yang saling terhubung.
- ✘ Hubungan antar CPU menggunakan jaringan khusus dengan kecepatan tinggi.
- ✘ Setiap CPU mempunyai memory, sistem operasi, dan aplikasi masing2.
- ✘ Contoh: IBM Blue Gene



VECTOR MACHINES

- ✘ CPU dapat memproses instruksi sama yang mempunyai data set dalam jumlah besar. Contoh vektor operation: $A = B \times C$ dimana masing2 A, B dan C adalah vektor dengan 64 elemen dari 64bit floating point number.
- ✘ Shared memory
- ✘ Keuntungan: performance, pemrograman mudah
- ✘ Kekurangan: harga mahal, scalability
- ✘ Contoh: Cray-1, NEC SX series
- ✘ Processor modern mempunyai beberapa kemampuan vector processing, seperti AltiVec pada prosesor PowerPC dan SSE pada prosesor Intel Pentium

VECTOR MACHINES (2)



NEC SX-9 (2008)



Cray-1 (1976an)

CLUSTER

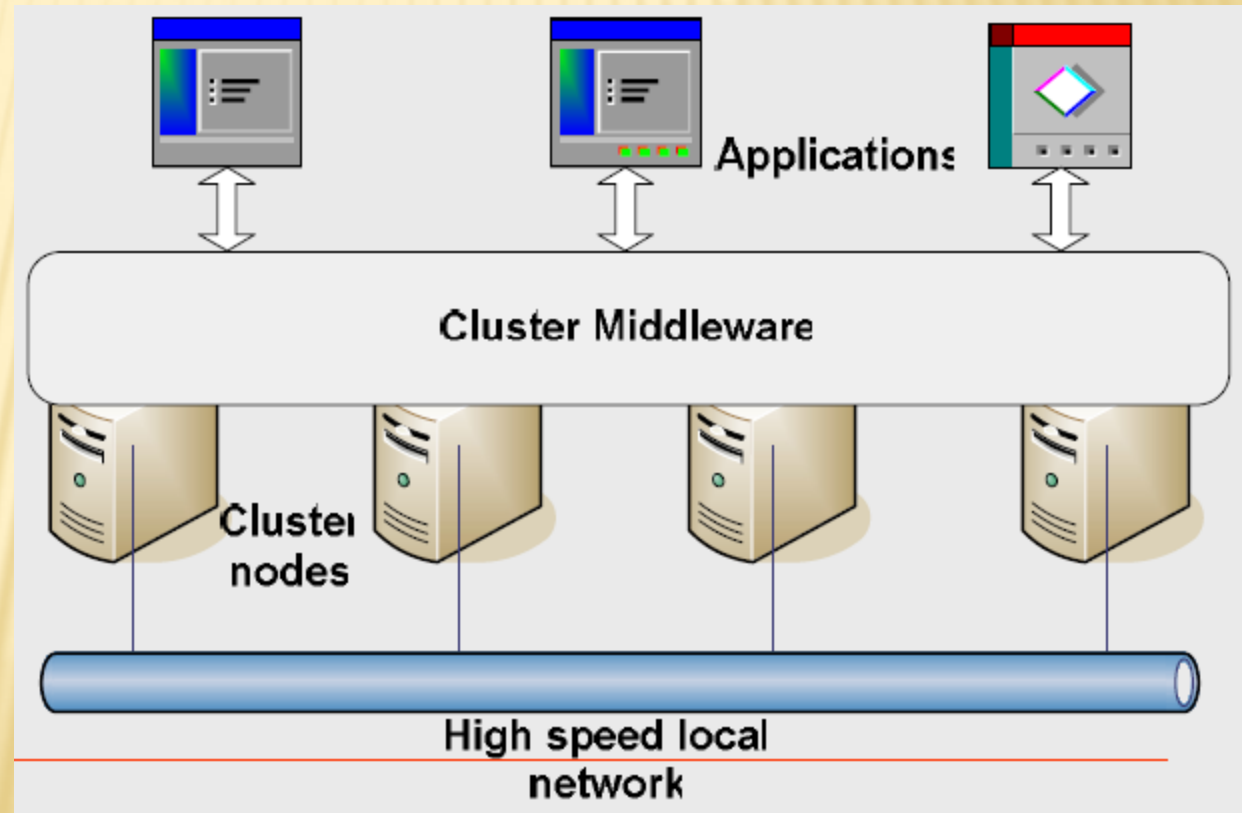
- ✘ Poor's man supercomputer
- ✘ "... collection of interconnected stand-alone computers working together as a single, integrated computing resource" – R. Buyya
- ✘ Menggunakan komponen2 standart untuk menghindari mahalnya harga2x komponen proprietary

KLASIFIKASI CLUSTER

- ✘ High performance clusters (HPC)
 - + Parallel, tightly coupled applications
- ✘ High throughput clusters (HTC)
 - + Large number of independent tasks
- ✘ High availability clusters (HA)
 - + Mission critical applications
- ✘ Load balancing clusters
 - + Web servers, mail servers, ...
- ✘ Hybrid clusters
 - + e.g. HPC+HA

KOMPONEN DARI CLUSTER

- ✘ Cluster terdiri dari:
 - + Cluster middleware
 - + Nodes
 - + Network
 - + OS



MANFAAT CLUSTER COMPUTING

- ✘ Menurunkan harga dari High-Performance Computer
- ✘ Bisa mempunyai Super Komputer pribadi
- ✘ Semakin banyak orang, semakin inovatif, lebih banyak aplikasi, dll
- ✘ Tanpa cluster computing, penggunaan High-Performance Computer hanyalah sebuah harapan untuk kelompok2 kecil dan negara berkembang

CLUSTER MIDDLEWARE

- ✘ Merupakan suatu interface diantara user dengan cluster hardware
- ✘ Middleware mensupport user dan hardware cluster dalam hal management, programming, dan implementation.
- ✘ Layer2 pada Middleware:
 - + Layer Single System Image (SSI)
 - + Layer Availability: memungkinkan suatu sistem Cluster untuk melakukan:
 - ✘ Checkpointing, Automatic Failover, Recovery from Failure
 - ✘ Toleransi terhadap kesalahan diantara node2 pada sistem cluster

TUJUAN DARI MIDDLEWARE PADA CLUSTER

- ✘ Complete Transparency
 - + Lets the user see a single cluster system..
 - ✘ Single entry point, ftp, telnet, software loading...
- ✘ Scalable Performance
 - + Pengembangan cluster dengan mudah
 - ✘ Tanpa harus merubah API (Application Programming Interface) & automatic load distribution.
- ✘ Enhanced Availability
 - + Automatic Recovery from failures
 - ✘ Menggunakan checkpointing & fault tolerant technologies
 - + Menjaga konsistensi data ketika terjadi replikasi

SINGLE SYSTEM IMAGE (SSI)

- ✘ Merupakan suatu “*ilusi*” yg dibuat oleh software atau hardware, sehingga kumpulan dari elemen2x cluster computing terlihat sebagai satu sumber saja.
- ✘ SSI membuat suatu sistem Cluster terlihat sebagai satu mesin baik oleh user, oleh aplikasi, maupun oleh network
- ✘ Suatu cluster tanpa SSI adalah bukan cluster

KEUNTUNGAN DARI SSI

- ✘ Penggunaan system resources secara transparan
- ✘ Meningkatkan reliability (ketahanan) dan availability (ketersediaan)
- ✘ Memudahkan pengelolaan sistem
- ✘ Mengurangi resiko kesalahan operator
- ✘ User tidak perlu mengenal arsitektur dari sistem untuk dapat menggunakan mesin cluster secara efektif

PROYEK2 RISET DALAM CLUSTER COMPUTING

- × **Beowulf** (CalTech and Nasa) - USA
- × **CCS** (Computing Centre Software) - Paderborn, Germany
- × **Condor** - Wisconsin State University, USA
- × **DJM** (Distributed Job Manager) - Minnesota Supercomputing Center
- × **DQS** (Distributed Queuing System) - Florida State University, USA
- × **EASY** - Argonne National Lab, USA
- × **HPVM** -(High Performance Virtual Machine),UIUC&now UCSB,US
- × **far** - University of Liverpool, UK
- × **Gardens** - Queensland University of Technology, Australia
- × **Generic NQS** (Network Queuing System),University of Sheffield, UK
- × **NOW** (Network of Workstations) - Berkeley, USA
- × **NIMROD** - Monash University, Australia
- × **PBS** (Portable Batch System) - NASA Ames and LLNL, USA
- × **PRM** (Prospero Resource Manager) - Uni. of S. California, USA
- × **QBATC**H - Vita Services Ltd., USA

SOFTWARE2 KOMERSIAL UNTUK CLUSTER COMPUTING

- ✘ **Codine** (Computing in Distributed Network Environment) - GENIAS GmbH, Germany
- ✘ **LoadLeveler** - IBM Corp., USA
- ✘ **LSF** (Load Sharing Facility) - Platform Computing, Canada
- ✘ **NQE** (Network Queuing Environment) - Craysoft Corp., USA
- ✘ **OpenFrame** - Centre for Development of Advanced Computing, India
- ✘ **RWPC** (Real World Computing Partnership), Japan
- ✘ **Unixware** (SCO-Santa Cruz Operations,), USA
- ✘ **Solaris-MC** (Sun Microsystems), USA

KESIMPULAN

- ❏ Cluster Computing menyelesaikan paradoks dari parallel processing
- ❏ Cluster Computing menawarkan pertumbuhan secara bertahap dan sesuai dengan pola pendanaan
- ❏ Trend2 baru dalam teknologi hardware dan software membuat clusters lebih menjanjikan
- ❏ Supercomputer berbasiskan clusters ada dimana2