

Käyttöjärjestelmät II

Tietoturva - esimerkki KJ:t

UNIX/Linux: Ch 10.7 [Tane 01]

W2000: Ch 11.8 [Tane01], Ch 16.6 [Stal 05]

Distributed Processing

Ch 14 [Stal 05]

Käyttöjärjestelmät II

UNIX tietoturva

Ch 10.7 [Tane 01]

Unix tietoturva

- n **Käyttäjän tunnistus, tiedot PCB:ssä**
 - u UID (User ID)
 - F kokonaisluku 0-65535
 - u GID (Group ID)
- n **Tiedostossa vastaavasti**
 - u omistaja, joka voi muuttaa oikeuksia
 - u oikeudet omistajalle, ryhmälle ja muille
- n **Tiedoston käyttö: tarkista onko omistajalla/ryhmällä tarvittavat oikeudet tiedostoon**
 - u tarkistus vain tiedoston avaamisen yhteydessä
- n **Kaikki KJ oliot ovat ”tiedostoja”**

UNIX käyttöoikeudet

- n **Tiedoston attribuutit (i-node)**
 - u omistaja (uid), ryhmä (gid)
 - u käyttöoikeudet (mode-kentän rwx-bitit)
- n **Käyttäjän uid ja gid käyttäjätietokannasta**
 - u `/etc/passwd` uid ja ensisijainen gid
 - u `/etc/group` käyttäjän muut ryhmänumerot
- n **uid ja gid periytyvät lapsiprosesseille ja edelleen luoduille tiedostoille**
 - u voi vaihtaa ohjelmallisesti



UNIX käyttöoikeudet



- n rootilla (uid=0) kaikki oikeudet kaikkeen
- n Käyttäjien jaottelu
 - u **u** omistaja
 - u **g** samaan ryhmään kuuluvat
 - u **o** muut käyttäjät
- n Oikeuksien jaottelu **u, g, o**
 - u **-** ei mitään
 - u **r** lukuoikeus
 - u **w** kirjoitusoikeus (oikeus muuttaa)
 - u **x** suoritusoikeus
- n Uusien tiedostojen käyttöoikeudet PCB:ssä olevan umask-oletuksen mukaan
 - u periytyy rajoitetusti
 - u käyttäjän oikeudet, umask, luonnin optiot

UNIX käyttöoikeudet



- n Hakemiston käyttöoikeudet
 - u **r** oikeus listata hakemiston sisältö
 - u **w** oikeus poistaa tiedosto hakemistosta
 - u **x** oikeus käyttää hakemistonimeä polkunimessä

- n Oikeudet oltava kaikkiin polkunimen osiin

- n Käyttöoikeuden hetkellinen laajennus, esimerkki:

- u vain rootilla w-oikeus */etc/passwd* tiedostoon
- u *passwd*-ohjelmalle asetettu **SETUID** bitti
 - F **effective userid** on tämän ohjelman (tiedoston *passwd*) ownerid
- u käyttäjä saa *passwd*-ohjelman suoritusajaksi root-oikeudet (koska root on owner), ja voi muuttaa oman salasanasansa
- u **SETGID** bitti vastaavasti (**SETGID** bitti)
 - F **effective groupid**

normal
- rW- --- --- 1 root

- srW- s--- t--- 1 root

advanced special permissions

"sticky bit"
keep file on swap device

Fig 10-39 [Tane 01]

System call	Description
<code>s = chmod(path, mode)</code>	Change a file's protection mode
<code>s = access(path, mode)</code>	Check access using the real UID and GID
<code>uid = getuid()</code>	Get the real UID
<code>uid = geteuid()</code>	Get the effective UID
<code>gid = getgid()</code>	Get the real GID
<code>gid = getegid()</code>	Get the effective GID
<code>s = chown(path, owner, group)</code>	Change owner and group
<code>s = setuid(uid)</code>	Set the UID
<code>s = setgid(gid)</code>	Set the GID

Fig. 10-39. Some system calls relating to security. The return code *s* is `-1` if an error has occurred; *uid* and *gid* are the UID and GID, respectively. The parameters should be self explanatory.

[Tane 01]

UNIX: Käyttöoikeudet

n Eräissä järjestelmissä myös käyttäjäkohtaisia pääsylistoja (ACL)

u Solaris, HP-UX

F esim. tietotekniikkaosaston kone "sirppi"

F `man acl`

u Linux

F ext2:ssa varauduttu toteuttamaan

- 8 tavua *i-node*:ssa

- **File ACL** ja **Directory ACL** -kentät

```
setfacl -m u:jussi:r tiedostoX
```



Linux PAM

- n **PAM – Pluggable Authentication Module**
- n **Parannettu autentikointi, hylkää huonot salasanat, vaadi salasanan vaihtoa aika ajoin**
- n **Kerberos optio**
 - u keskitetty organisaation turvajärjestelmä
 - u käyttäjän tunnistaminen
 - u TGS – Ticket Granting Service
 - F valtakirjat verkkopalveluihin
 - F väärentämättömiä, vain vähän aikaa voimassa olevia valtakirjoja
- n **Älykortti- ja äänitunnistus optiot**

Linux ext2fs tiedonsuojaus

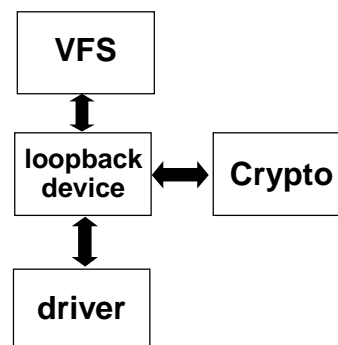
- n **Kuten std UNIX**
 - u user, group, other
 - u r, w, e, x
 - u setuid, setgid
- n **Tiedostolle myös**
 - u a append only
 - u i immutable
 - F ei voi muuttaa, tuhota tai vaihtaa nimeä
 - F ei voi linkittää (hard link, symbolic link)

LSM - Linux Security Module

- n **Määrittely ylimääräiselle valvontamoduulille**
 - u ladattava ytimen moduuli
 - u aktivoituu vasta, kun std pääsynvalvonta on ensin hyväksynyt käyttäjän tai resurssin käytön (LSM on lisäsuoja)
- n **LSM SELinux (Security Enhanced Linux)** <http://www.nsa.gov/selinux/>
 - u NSA – National Security Agency (USA)
 - u MAC – Mandatory Access Control
 - F joka tiedostolle selkeät oikeudet (write up, read down)
 - F sääntöjoukko, jota käyttäjät eivät voi manipuloida
 - u jäykkä, tehokas, luotettava
- n **LSM Capabilities**
 - u valtakirjaperustainen pääsynvalvonta
 - F i-node:n kentät File ACL ja Directory ACL
 - u tarkemmat oikeudet sovellukselle käyttäjästä riippumattomasti
 - u POSIX.1e suojausstandardi

Linux kryptografiamoduuli

- n Cryptographic API -määrittely
- n VFS (virtual file system) ei kutsu laiteajuria suoraan, vaan välissä on **loopback device**
- n Loopback device käyttää tarvittaessa kryptomoduulia aina tiedostoa käytettäessä
- n per hakemisto?
- n per tiedostojärjestelmä?



Käyttöjärjestelmät II

Windows 2000 Tietoturva



W2K Tietoturva

- n **Noudattaa "Orange Book" C2 luokitusta**
 - u Dept of Defence (US) Security requirements C2
 - u Trusted Computer System Evaluation Criteria
- n **C2 – ei kovin paljoa vaadittu**
 - u henkilökohtainen kirjautuminen (ei ryhmä)
 - u pääsy vain sallittuihin tiedostoihin ja ohjelmiin
- n **Muita, parempia turvatasoja**
 - u B1, B2, B3
 - F B1: kuten C2 ja Mandatory Access Control (MAC)
 - F B3: kuten B2 ja kaiken monitorointi ja suojausdomainit
 - u A1, A2
 - F A1: kuten B, mutta formaalisti todistettu oikein toimivaksi
 - F A2: määritellään joskus myöhemmin

<http://www.dynamoo.com/orange/summary.htm>

W2K Suojausympäristö

Fig 16.12 (a) [Stal 05]

- n **Joka prosessilla suojauslipuke (access token)** (valtakirja)
 - u prosessin tunnistetiedot, "kuka minä olen"
 - F annetaan järjestelmään kirjautumisen yhteydessä
 - F omistaja, ryhmä (POSIX)
 - u luotaville objekteille määrättyt oletusoikeudet
 - F default ACL
 - u mahdolliset erityisoikeudet ('special power')
 - F shutdown, write file Y
 - u periytyy lapsiprosesseille
 - u voidaan muuttaa prosessikohtaisesti
- n **Joka oliolla suojauskuvaaja (security descriptor)**
 - u suojauskuvaajassa pääsylista
 - F discretionary ACL
- n **Tarkistus: vertaa prosessin (käyttäjän) pääsylippua olion (kohteen) pääsylistaan**

Fig 16.12 (b,c) [Stal 05]

W2K suojauskuvaaja (security descriptor)

- n **Joka oliolla oma suojauskuvaaja**
 - u "kuka saa tehdä mitä?"
 - u lipukkeita (esim. mitkä kentät käytössä)
 - u kohteen omistaja (**owner SID**) tai ryhmä (**group SID**)
 - F joku olion luoja suojauslipukkeen SID'eistä
 - u **DACL pääsylista** (discretionary access control list)
 - F ketkä käyttäjät, mitkä ryhmät saavat käyttää
 - F omistaja voi manipuloida
 - u **SACL pääsylista** (system ACL)
 - F mitä auditointilokiin, erityisoikeuksien käyttö
 - F omistaja ei saa manipuloida (yleensä)

Fig 16.12 (b) [Stal 05]

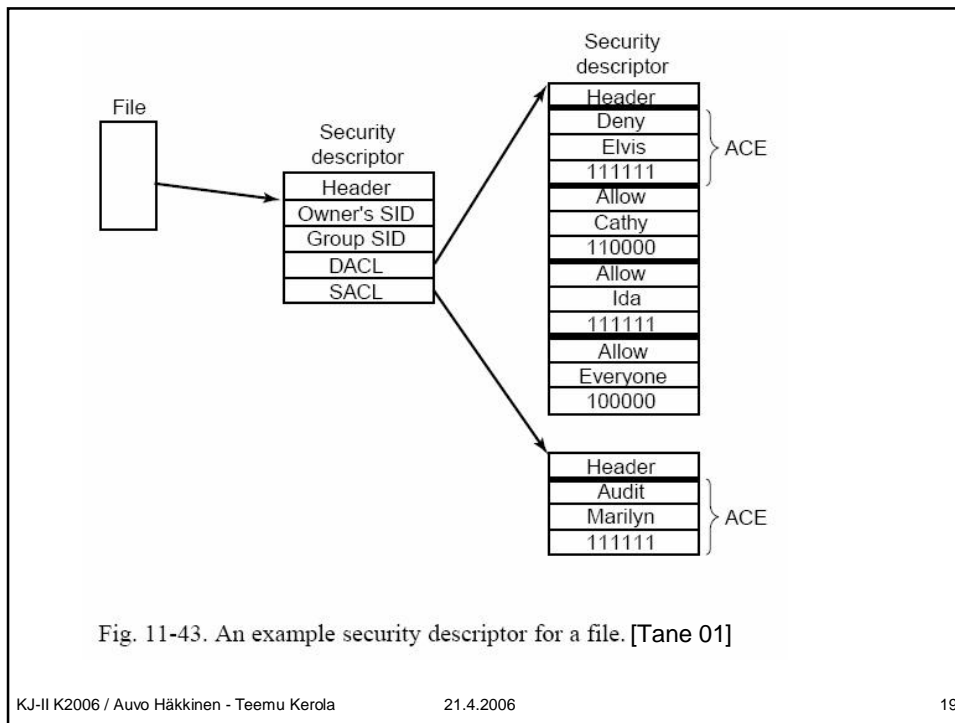
discretionary = vapaa harkinta, päätösvalta, harkinnan varainen

W2K suojattujen olioiden käyttö

- n **Ensimmäinen viite (esim. tiedoston avaus)**
 - u vertaa prosessin pääsilyppua olion pääsilystaan (DACL)
 - u etsi ensimmäinen ACE (access control element), joka sopii tähän käyttäjään tälle käyttötavalle
 - u jos kaikki kunnossa, anna **kahva (handle, valtakirja)** olioon
- n **Myöhemmät viitteet kahvan avulla**
 - u tarkista aina, että käyttötapa on sellainen, joka oli mukana jo ensimmäisellä kerralla kun pääsy olioon sallittiin
 - u jos prosessi yrittää saamansa "read"-oikeuden asemesta kirjoittaa, niin se ei onnistu
 - u jos olion omistaja poistaa "read" oikeuden, niin se ei estä vanhoja käyttäjiä lukemasta

W2K DACL – Discretionary ACL

- n **Koostuu useasta pääselementeistä**
 - u **ACE** (Access Control Element) Fig 16.12 (c) [Stal 05]
- n **Kaksi ACE-tyyppiä**
 - u **Allow** – kuka ei saa käyttää ja miten Fig 11-43 [Tane01]
 - u **Deny** – kuka saa käyttää ja miten
- n **Käyttö: käy listaa läpi kunnes tälle käyttäjälle (SID) ja käyttötavalle löytyy ensimmäinen ACE ja menettele sen mukaan**
 - u sijoita Deny ACE -elementit ennen Allow ACE -elementtejä!
 - ⌘ esim. kaikki saa, mutta Elvis ei
- n **Käyttötavat koodattu pääsyoikeusmaskiin (access mask)**
 - u ks. seuraava kalvo (Fig 15.12 [Stal01])



W2K DACL Esimerkki (NTFS)

demo.txt Properties

General Security Summary

Group or user names:

- Administrators (WRL-142\Administrators)
- nikelnde (TKTL\nikelnde)
- SYSTEM
- Teemu Kerola (TKTL\kerola)
- Users (WRL-142\Users)

Permissions for Administrators

	Allow	Deny
Full Control	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Modify	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read & Execute	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Write	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Special Permissions	<input type="checkbox"/>	<input type="checkbox"/>

For special permissions or for advanced settings, click Advanced.

Advanced

OK Cancel Apply

Advanced Security Settings for demo.txt

Permissions Owner Effective Permissions

To view more information about Special permissions, select a permission entry, and then click Edit.

Permission entries:

Type	Name	Permission	Inherited From
Deny	nikelnde (TKTL\nikelnde)	Read Permissions	<not inherited>
Allow	Administrators (WRL-142\Admin...	Full Control	C:\
Allow	Teemu Kerola (TKTL\kerola)	Full Control	C:\
Allow	SYSTEM	Full Control	C:\
Allow	Users (WRL-142\Users)	Read & Execute	C:\

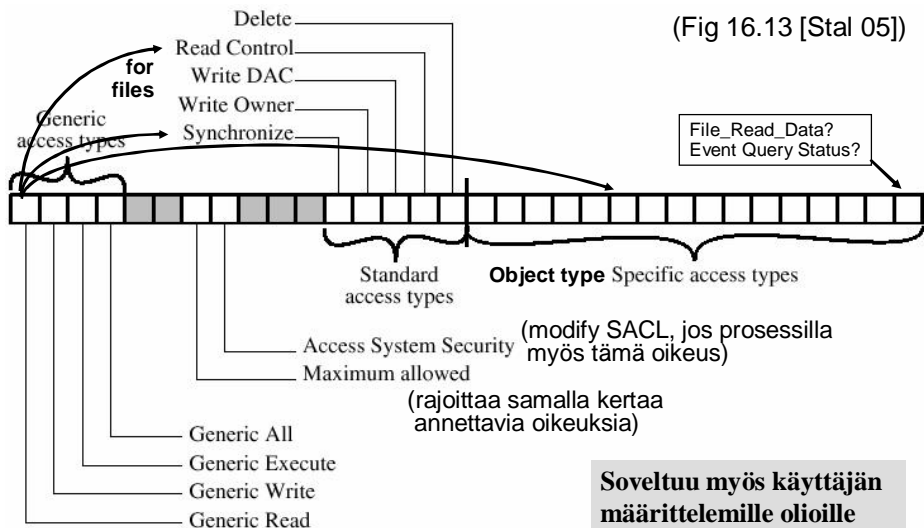
Inherit from parent the permission entries that apply to child objects. Include these with entries explicitly defined here.

Advanced Security Settings for demo.txt

OK Cancel Apply

KJ-II K2006 / Auvo Häkkinen - Teemu Kerola 21.4.2006 20

W2K pääsyoikeusmaski (access mask)



KJ-II K2006 / Auvo Häkkinen - Teemu Kerola

21.4.2006

21

W2K SACL – Security ACL

- n **Mistä tapahtumista tähän olioon kerätään auditointilokia**
 - u käyttäjä ei tiedä
 - u olion omistaja ei tiedä, ei voi muuttaa
- n **Esimerkkejä**
 - u Marilyn'in kaikki operaatiot tähän olioon pistetään lokiin Fig 11-43 [Tane 01]
 - u Kaikkien käyttäjien kaikki operaatiot tähän suojattuun olioon pistetään lokiin
- n **Auditointiloki on olio, jolla oma suojauskuvaaja ja DACL pääsyylista**

KJ-II K2006 / Auvo Häkkinen - Teemu Kerola

21.4.2006

22

W2K Security API

Win32 API function	Description
InitializeSecurityDescriptor	Prepare a new security descriptor for use
LookupAccountSid	Look up the SID for a given user name
SetSecurityDescriptorOwner	Enter the owner SID in the security descriptor
SetSecurityDescriptorGroup	Enter a group SID in the security descriptor
InitializeAcl	Initialize a DACL or SACL
AddAccessAllowedAce	Add a new ACE to a DACL or SACL allowing access
AddAccessDeniedAce	Add a new ACE to a DACL or SACL denying access
DeleteAce	Remove an ACE from a DACL or SACL
SetSecurityDescriptorDacl	Attach a DACL to a security descriptor

Fig. 11-44. The principal Win32 API functions for security.[Tane 01]

ACL tarkemmin: Microsoft TechNet artikkeli: [click](#)

Operating Systems II

Distributed Processing

Ch 14 [Stal 05]

Distributed Processing

n Survey of distributed processing capabilities

- u client-server
- u database applications
- u middleware
- u distributed message passing
- u remote procedure calls
- u clusters

Now:
Ch 14
Oper. Syst. II

n Distributed Process Management

- u "what is in the OS to support distributed processing?"

Later:
Ch 15
separate course
on Distr. Systems
(Hajautetut järj.)

Client/Server

n Server provides shared services

- u database server
- u name server
- u web server
- u password server

Fig 14.1 [Stal 05]

n Access through network (LAN, WAN, Internet)

Fig 14.2 [Stal 05]

n Server may also be a client

n Database server

- u database layer below application layer

Fig 14.3 [Stal 05]

Client/Server Application Classes

- n **Where is processing done? What part?** Fig 14.5 [Stal 05]
 - u Host-based
 - F E.g., stupid terminal, not really a client
 - u Server-based
 - F E.g., web browsing
 - u Cooperative processing
 - F E.g., general database application
 - u Client-based
 - F E.g., web browsing with applets
- n **Which class best for this application?**
- n **What OS support is available?**

Middleware

- n **What if client does not know who the server is?**
 - u "I just want this type of service"
- n **Clearinghouse for service requests: middleware** Fig 14.6 [Stal 05]
 - u uniform access to many resources Fig 14.8 [Stal 05]
 - u platform independent Fig 14.9 [Stal 05]
 - F OS: Unix, Linux, SVR4, W2000
 - F database: Oracle, Gupta
 - F DECnet, Novell, TCP/IP Fig 13.10 [Stal01]

Distributed Message Passing

- n **Plain messages for client/server** Fig 14.10 (a) [Stal 05]
 - u reliable or not? blocking or not? Fig 14.11 [Stal 05]
- n **RPC - Remote Procedure Call** Fig 14.10 (b) [Stal 05]
 - u use just like local procedure calls
 - u standardized interface Fig 14.12 [Stal 05]
 - u reusable modules
 - u parameter problems
 - F marshalling
 - F pointers – call-by-reference
 - u non-persistent/persistent binding
 - F save handle for remote process or not?
 - u synchronous/asynchronous (to block or not)
- n **RMI – Remote Method Invocation**
 - u for Java users

Object Oriented Mechanisms

- n **ORB – Object Request Broker** Fig 14.10 (c) [Stal 05]
 - u higher level concept than RPC or RMI
- n **The good thing about standards is that you can choose which one to use**
 - u **DCOM** – Distributed Component Object Model
 - F Microsoft, Digital
 - F each object can have multiple interfaces
 - interface must be defined when requesting service
 - u **CORBA** – Common Object Request Broker Architecture
 - F OMG - Object Management Group (non-profit)
 - F IBM, Apple, Sun, ...
 - F **ORB** (Object Request Broker) to ORB communication
 - F **IDL** (Interface Definition Language) for programming language independent interface definition
 - one interface per object

Cluster Computer

Shared memory multicomputer

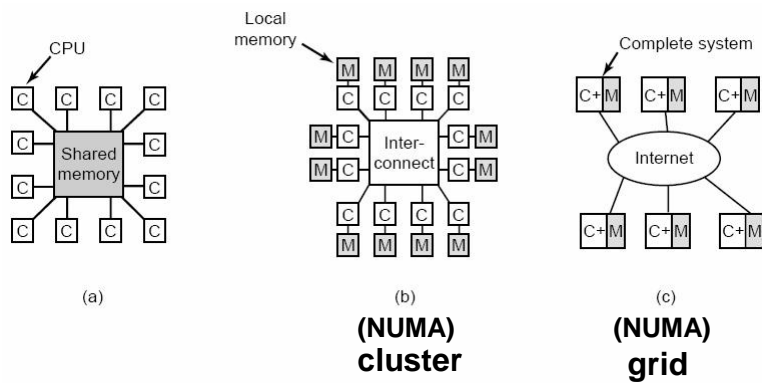


Fig. 8-1. (a) A shared-memory multiprocessor. (b) A message-passing multicomputer. (c) A wide area distributed system. [Tane 01]

Cluster

- n **Cluster, multicomputer, COWS (cluster of workstations)**
- n **Many whole (similar?) computers**
 - u can work independently if needed
- n **Interconnected**
- n **Work together**
- n **Unified computing resource**
 - u e.g. memory, disk
- n **Illusion of one machine**

Benefits of Clustering

- n **Absolute (?) scalability**
- n **Incremental scalability**
- n **High availability**
- n **Superior price/performance**
 - u as compared to what? SMP? Grid? Supercomputer?

- n **Disadvantages?**
 - u more complexity than uniprocessing or SMP
 - F E.g., synchronization
 - u communication delay vs. memory access
 - u which applications suitable for it?

Cluster Configurations

- n **Shared disk or not?** Fig 14.13 [Stal 05]

- n **Passive standby** Tbl 14.2 [Stal 05]
 - u you would call this "clustering"?
 - F need many whole computers

- n **Active secondary**
 - u separate servers
 - F each has its own disks
 - u servers connected to disks – "shared nothing"
 - F shared disks, disks partitioned to servers
 - F each disk has one "owner" (user)
 - u servers share disks
 - F shared disks
 - F need mutex locks

Cluster Failure Management

- n **High Performance Cluster**
 - u no redundancy, just lots of processing power
 - u example: Magnetic Resonance Image (MRI) scanner
- n **Highly Available Cluster**
 - u probably all resources available
 - F some resources serve as backups
 - u no guarantee of transaction execution
 - u application provides for consistency
 - u example: soft real time
- n **Fault Tolerant Cluster**
 - u guarantees that all resources available
 - F HW redundancy, transaction logging
 - u application does not need to provide consistency
 - u trouble at resource X?
 - F start using alternative (spare) resource **failover** (varalaite käyttöön)
 - F repair X or replace X
 - F return to using X **failback** (laite takaisin käyttöön)
 - u examples: hard real time, aircraft control system

Load Balancing Cluster

- n **Incremental scalability**
 - u automatic use of new resources
- n **Migrate services/work from one computer to another**
 - u how to migrate processes?
 - F code, data, PCB?
- n **Load balancer node**
 - u one node dedicated to load balancing
- n **Example**
 - u e-business with high user volumes

Cluster Application Concurrency

- n **Must have application level concurrency**
 - u middleware layer to enable co-operation
 - u how to find it?
- n **Parallelizing compiler**
 - u compiler does the parallelization work
 - F "dusty decks" OK, though may not be so good
 - u may make compiled application dependent on cluster size
- n **Parallelized application**
 - u programmer does the parallelization work
 - F hard work, complex
 - u may make application really dependent on cluster size
- n **Parametric computing (parallelized problem)**
 - u run many instances of same application, one in each node, with different parameters
 - F simple, but not suitable so often

Fig 14.14 [Stal 05]

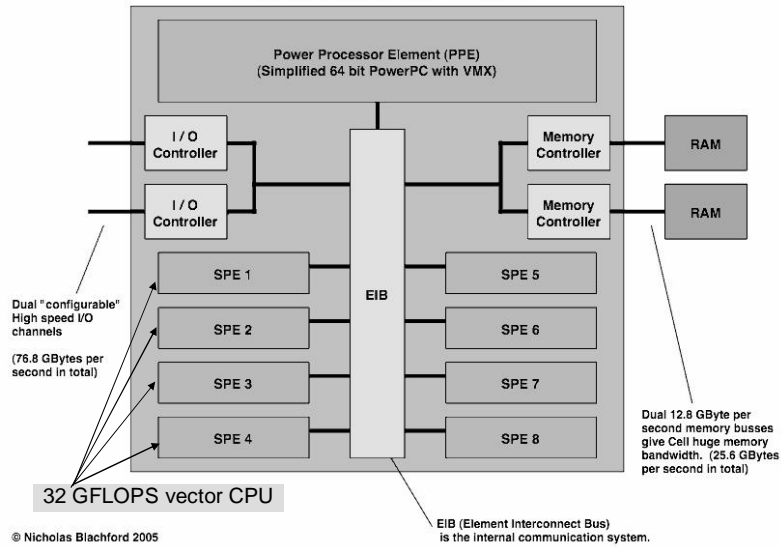
Cluster Middleware

- n **Single everything – feels like one computer**
 - u system image, entry point, control point
 - u virtual networking
 - u memory space
 - u job management
 - u user interface
 - u I/O space
 - u process space
- n **Checkpointing**
 - u failure recovery
- n **Process migration**
 - u load balancing

Fig 14.14 [Stal 05]

Sony/Toshiba/IBM Cell Processor Architecture

This diagram is based on data released by STI
Some acronyms have changed: SPE = APU, PPE = PU



© Nicholas Blachford 2005

<http://www.blachford.info/computer/Cells/Cell1.html>

KJ-II K2006 / Auvo Häkkinen - Teemu Kerola

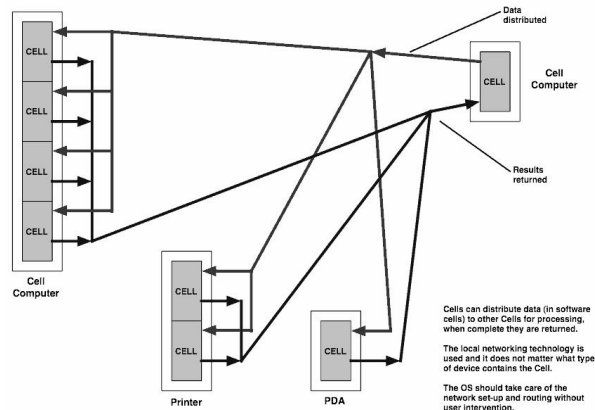
21.4.2006

39

Cell Operating System

- n Cell OS?
- n Load balancing?
- n Migration?
- n Shared mem?
- n Parallelizing compiler?
- n Shared memory cluster?
- n Distributed shared memory cluster?

Distributed Processing with Cells



© Nicholas Blachford 2005

<http://www.blachford.info/computer/Cells/Cell1.html>

KJ-II K2006 / Auvo Häkkinen - Teemu Kerola

21.4.2006

40

W2000 Cluster Server (Wolfpack)



- n **Shared nothing** Fig 14.15 [Stal 05]
 - u shared disk, each disk volume has one owner/user
 - u max 32 nodes, max 32 GB memory
- n **Cluster service (cluster middleware)**
 - u at each node
- n **Cluster node resource**
 - u disk drive, network card, application, database, TCP/IP address, ...
 - u "**online**", if resource available to others
 - u packaged into **groups**
 - F e.g., all resources needed to run one application
 - F unit of failover and load balancing
- n **(New and better: W2003 Cluster Server)**

W2000 Cluster Service (contd)

- n **Middleware layer**
- n **Node manager** Fig 14.15 [Stal 05]
 - u who is in cluster now?
 - u heartbeat messages to other node managers
 - u no heartbeat from node X for a while → X is dead!
- n **Configuration database manager**
 - u who owns what resources
 - u fault-tolerant transactions
- n **Resource manager & failover manager**
 - u startup, reset, failover
- n **Event processor**
 - u cluster components synchronize with events

Beowulf Cluster with Linux

Fig 14.18 [Stal 05]

- n **Beowulf 1994**
 - u are many cheap PC's better than one good workstation?
 - u yes....
- n **Beowulf features**
 - u normal cheap components, no custom components, many vendors
 - u dedicated processors, dedicated network
 - u one controlling node (**front end node**, or **head node**)
 - u similar slave computers (for easy load balancing)
 - u scalable I/O
 - u freely available software
 - u freely available distribution computing tools
 - u give design and improvements to the community (free?)
- n **Examples**
 - u ETH Zurich, 251 nodes, 502 processors (June 2001)
 - u Niflheim Linux cluster, 5.0-TeraFLOPS, 945 node supercomputer
- n **Beowulf Cluster with Windows**



Beowulf Software

Fig 14.18 [Stal 05]

- n **Each node has own copy of Linux kernel**
- n **Autonomous Linux system**
- n **Kernel extensions to participate in global namespaces**
 - u cluster middleware
 - u Beowulf Distributed Process Space (BPROC)
 - F start remote processes without login
 - F remote processes visible in cluster *front end* node
 - u Beowulf Ethernet Channel Bonding
 - F load balancing over multiple Ethernets
 - F LAN, not WAN, not internet
 - u PvmSync
 - F distributed synchronization within cluster
 - u EnFuzion
 - F tools for parametric computing
 - control jobs in remote nodes



Grid Computing

n Utilize idle computing resources in Web

- u home computers?
- u company computers?

n Many layers to utilize heterogeneous computers

- u **application layer**
- u **collective layer** for coordination
- u **resources layer** for sharing resources
- u **connectivity layer** for connections
- u **fabric layer** for physical resource usage

n Examples

- u SETI@home [click](#)
- u Globus toolkit for business solutions (Globus Alliance) [click](#)

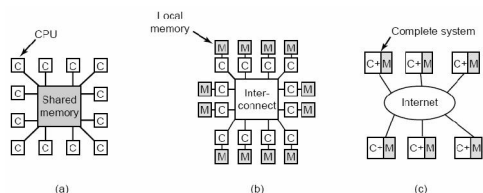


Fig. 8-1. (a) A shared-memory multiprocessor. (b) A message-passing multicomputer. (c) A wide area distributed system.

grid

[Tane 01]

Review Questions

- n How do Linux and W2000 security features differ?
- n What is good/bad with Linux/W2000 security?
- n What can be done with Linux but not in W2000?
- n What can be done with W2000 but not in Linux?

- n What is needed from OS to support clusters?
- n What is needed from OS to support grids?
- n What synchronizations primitives can (not) be used with clusters?
- n What synchronizations primitives can (not) be used with grids?

-- END --



Operating Systems II