Logiczne partycjonowanie systemów

- Grzegorz Jaśkiewicz
- Dariusz Stefański

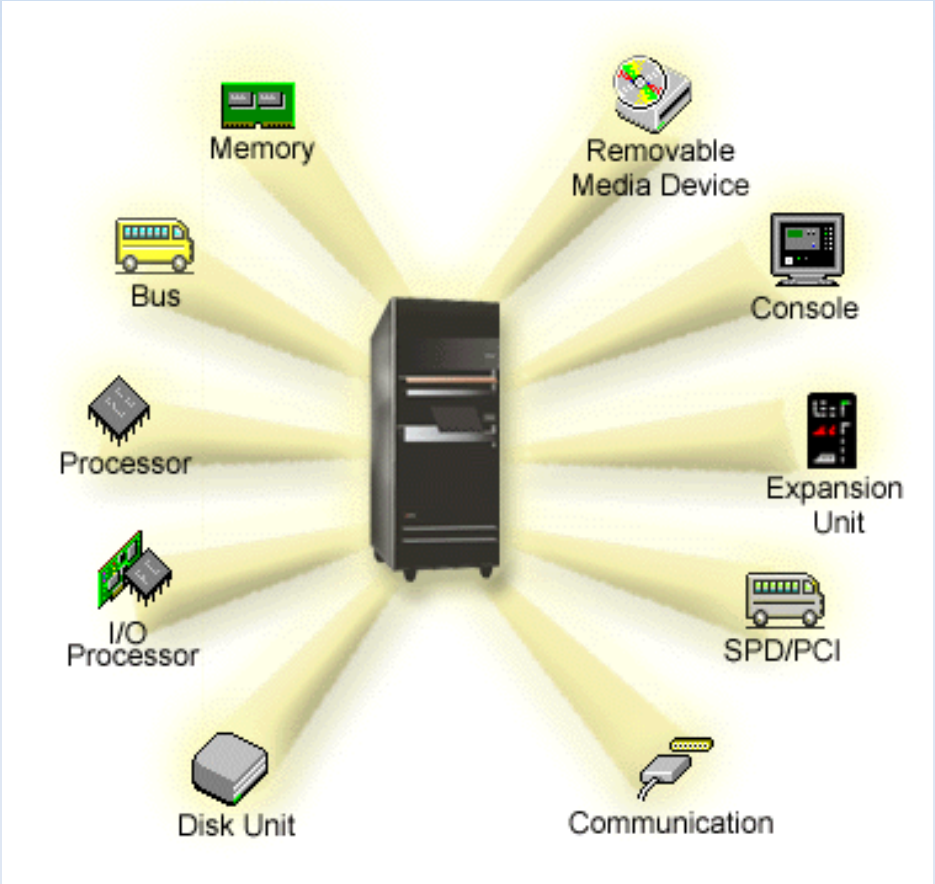
Plan prezentacji

- Podstawowe informacje
- Zastosowanie
- Hypervisor
  - działanie hypervisora
- Wsparcie sprzętowe
- Partycjonowanie
  - sprzętowe
  - LPAR
  - DLPAR
- Konkretne produkty
- (D)LPAR w jednym z banków
- IBM iSeries

Podstawowe informacje

- partycjonowanie systemu – odnosi się do dzielenia komputera na wiele jednostek obliczeniowych, każda z nich jest nazywana partycją
- partycjonowanie umożliwia podział zasobów serwera na kilka mniejszych serwerów, z których każdy jest niezależny od pozostałych

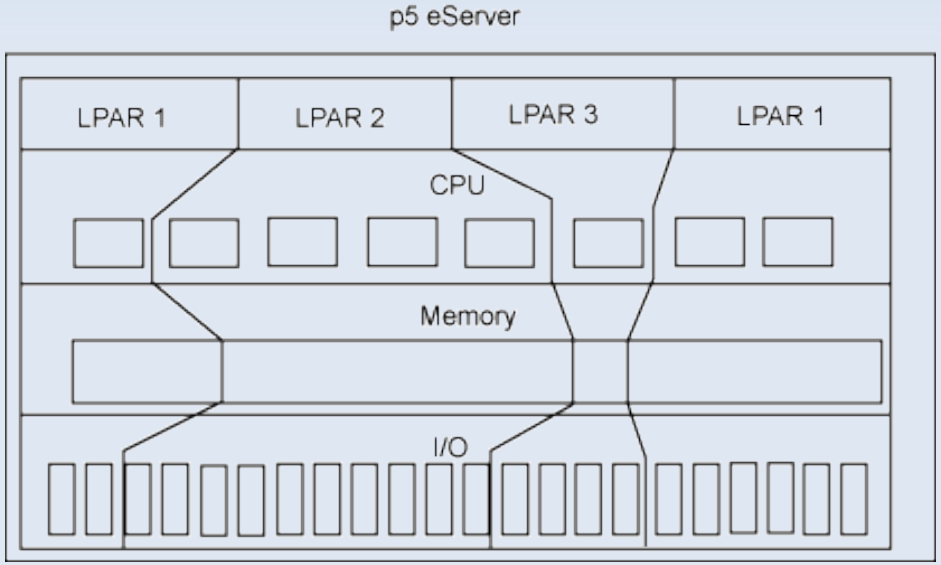
Zasoby podlegające partycjonowaniu



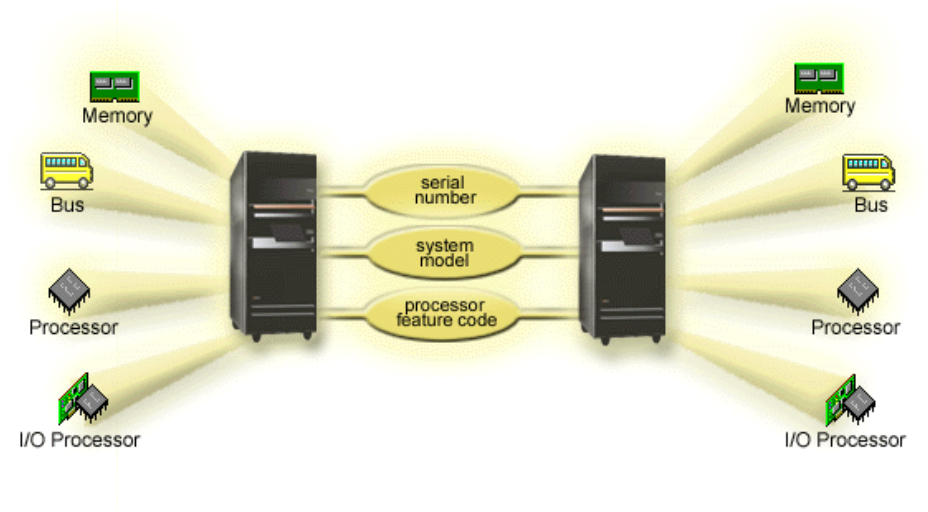
- procesor
- pamięć RAM
- dyski
- porty USB
- urządzenia I/O

Zasoby podlegające partycjonowaniu

- Tworzenie mniejszych systemów z większych
  - podział:
    - procesorów
    - pamięci
    - urządzeń I/O
- Oszczędność:
  - miejsca
  - energii
  - administratorów
  - sprzętu



Zasoby podlegające partycjonowaniu



- wszystkie partycje są autonomiczne - system operacyjny na pojedynczej partycji widzi tylko swoje zasoby
- niektóre informacje są identyczne dla wszystkich partycji np. numer seryjny komputera

Zastosowanie

- wiele komputerów na jednej fizycznej maszynie – oszczędność pieniędzy i miejsca

Zastosowania:
konsolidacja serwerów

- Podejście do efektywnego wykorzystania zasobów serwera w celu zmniejszenia ogólnej liczby serwerów a także miejsca ich przechowywania
- W wielu firmach serwer wykorzystuje swoje możliwości na poziomie 15-20%

Zastosowania:
skalowalne serwery

- wzrost przedsiębiorstwa prowadzi do potrzeby rozbudowania infrastruktury serwera
- można przydzielić dodatkowe zasoby serwerowi

Zastosowania:
środowiska produkcyjno-testowe

- Tworzenie oprogramowania w środowisku produkcyjnym i testowanie w środowisku testowym
- Dzięki autonomiczności partycji awaria jednej nie wpływa na inne co zwiększa produktywność

Hypervisor

- platforma wirtualizacyjna, która pozwala wielu systemom operacyjnym pracować na jednej maszynie w jednym czasie

Native Hypervisor

- oprogramowanie, które pracuje bezpośrednio na fizycznym sprzęcie(jako minisystem operacyjny). Systemy operacyjne, które teraz możemy dodawać nie mają bezpośredniego dostępu do fizycznych urządzeń, pracują na drugim poziomie ponad nim
- przykłady: Xen, Vmware ESX Server, L4 microkernels

Hosted Hipervisor

- oprogramowanie, które **nie** pracuje bezpośrednio na fizycznym sprzęcie(jak Native) lecz z wykorzystaniem środowiska systemu operacyjnego. Systemy operacyjne, które teraz możemy dodawać pracują na trzecim poziomie ponad sprzętem fizycznym
- przykłady: Vmware Server, Vmware Workstation, VirtualBox

LPAR – partycjonowanie logiczne

- logiczna partycja, potocznie zwana LPAR, to podzbiór komputerowych, fizycznych zasobów widzianych jako osobny komputer
- jedna fizyczna maszyna może być podzielona na wiele LPARów, które funkcjonują na własnych OS
- technologia stworzona przez IBM około 1990 r, początkowa zaprojektowana dla komputerów klasy mainframe dla architektury ESA/390
- następnie kontynuowana dla komputerów zSeries oraz System z9
- w późniejszym czasie IBM rozszerzył tą ideę o komputery nie będące mainframe'ami – pSeries oraz iSeries
- Systemami operacyjnymi wspierającymi technologię LPAR są z/OS, z/VM, z/VSE, z/TPF, AIX, Linux oraz i5/OS

Zarządca partycji(hypervisor)

- w LPAR stosuje się podejście Native Hypervisor
- minisystem operacyjny
- dzieli sprzęt na logiczne partycje(LPARy)
- zapewnia izolacje między LPARami
- mapuje zasoby fizyczne na wirtualne
- wirtualny dostęp do sieci

DLPAR – dynamiczne partycjonowanie logiczne

- zwiększa elastyczność systemów partycjonowania
- pozwala administratorom dodawać, odejmować lub przenosić między LPARami dostępne zasoby systemowe bez konieczności restartowania systemów pracujących na tych partycjach
- dzięki temu można przydzielać zasoby tam, gdzie są one najbardziej potrzebne

CUoD – Dynamic Capacity on Demond

- dynamiczna moc obliczeniowa na żądanie
- nadmiarowe procesory, które nie są wykorzystywane przez żaden LPAR, ale jeśli ktoś uzna, że jakiejś partycji nie wystarcza dotychczasowa ilość procesorów, to wtedy można jej dodać kolejne
- administrator może je dynamicznie uruchomić, a następnie za pomocą dynamicznego partycjonowania przydzielić wybranym partycjom bez konieczności wyłączania całego systemu

Zastosowanie CUoD

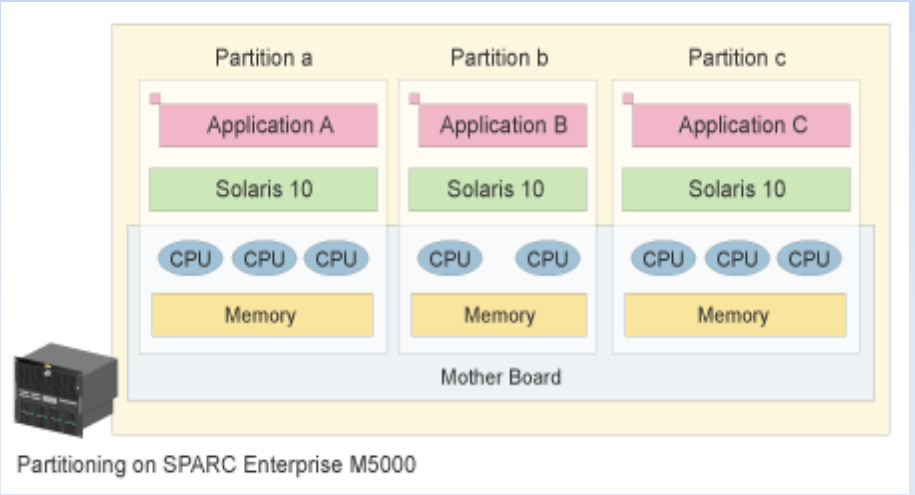
- trwały przyrost mocy
  - zaplanowany rozwój przedsiębiorstwa
- tymczasowy przyrost mocy
  - okresy szczytowej aktywności
    - sklep ze świątecznymi prezentami
- testowy przyrost mocy
  - testowanie nowych aplikacji
- zapasowa moc
  - odtwarzanie po awarii
    - odzyskiwanie bazy danych

Dynamic CPU Guard

- mechanizm pozwalający w razie awarii procesora automatycznie odłączać go od korzystającego z niego systemu operacyjnego zanim dojdzie do awarii całego systemu
- jeśli LPAR zawiera procesory zapasowe podmienia procesor, który uległ awarii

Partycjonowanie sprzętowe

- Sprzętowo realizowane operacje partycjonowania

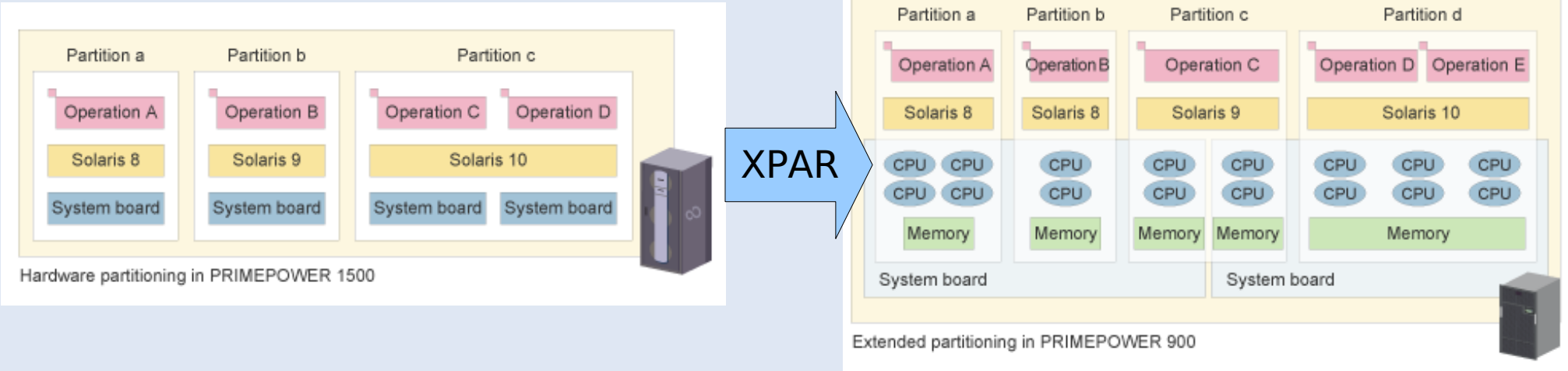


Partitioning on SPARC Enterprise M5000

- Zalety:
  - brak warstwy pośredniczącej pomiędzy sprzętem, a oprogramowaniem – zwiększona wydajność
  - odporność na błędy w warstwie sprzętowej
- Wady:
  - W porównaniu do innych metod mniejsza dowolność podziału systemu

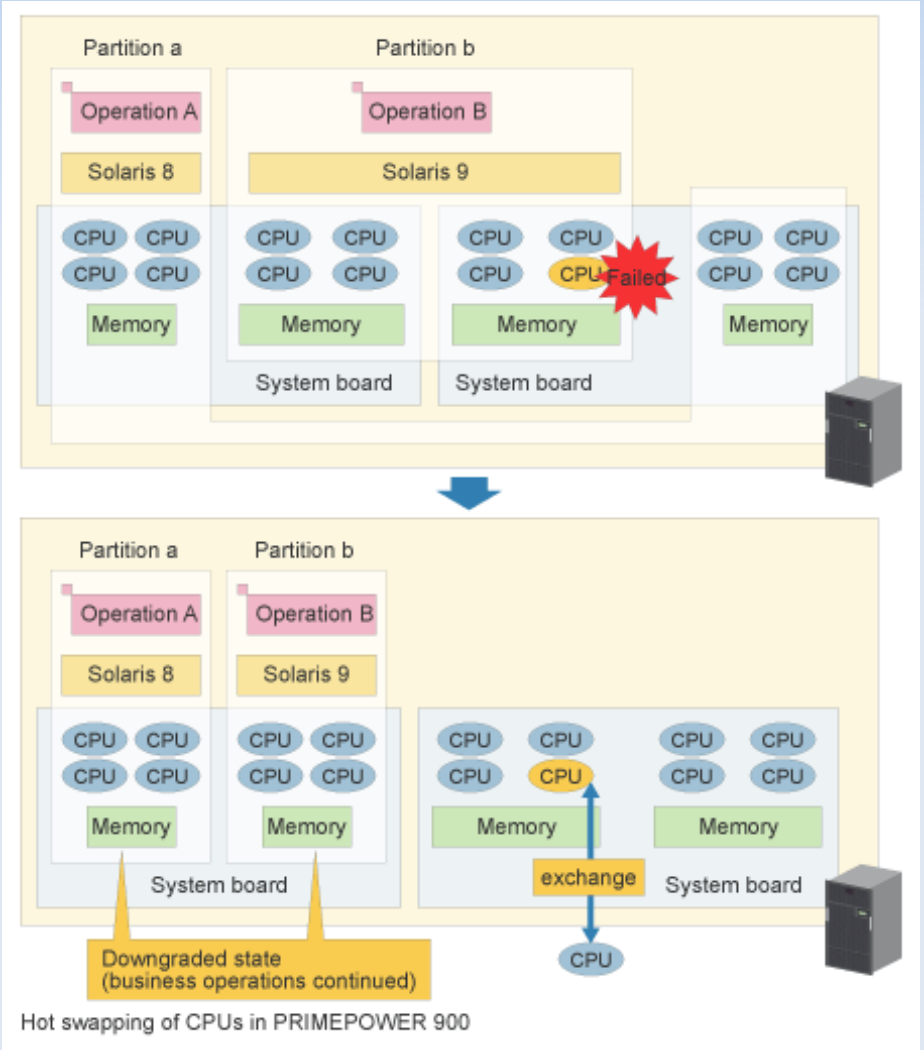
Partycjonowanie sprzętowe:
XPar

- e**X**tended **PAR**titioning
  - Umożliwia dzielenie zasobów w obrębie pojedynczej płyty głównej
  - Możliwość łączenia ze sobą procesorów o różnych szybkościach



Partycjonowanie sprzętowe:
dynamiczna realokacja

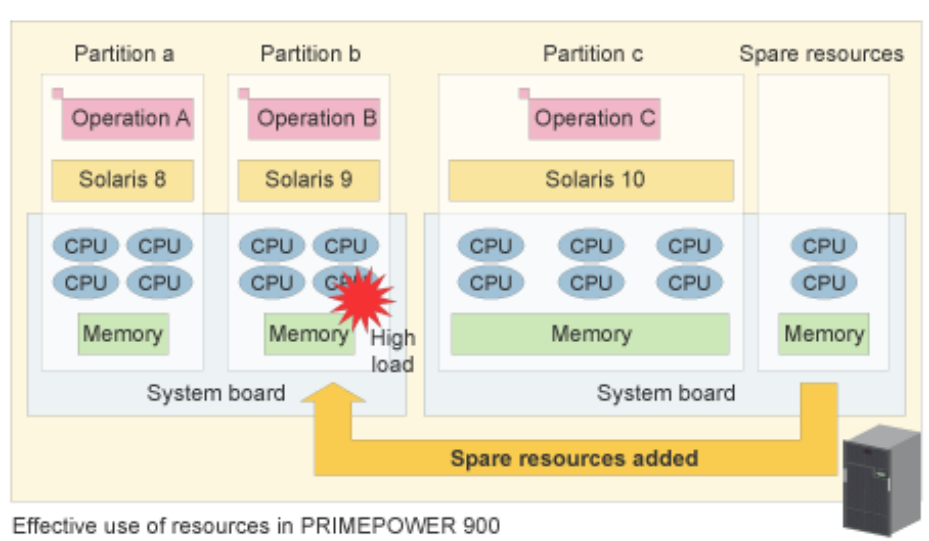
- możliwość przydzielania dodatkowych zasobów w miarę potrzeb
- nieprzerwanie działanie w razie awarii
- równoważenie obciążeń



Hot swapping of CPUs in PRIMEPOWER 900

Partycjonowanie sprzętowe:
dynamiczna realokacja

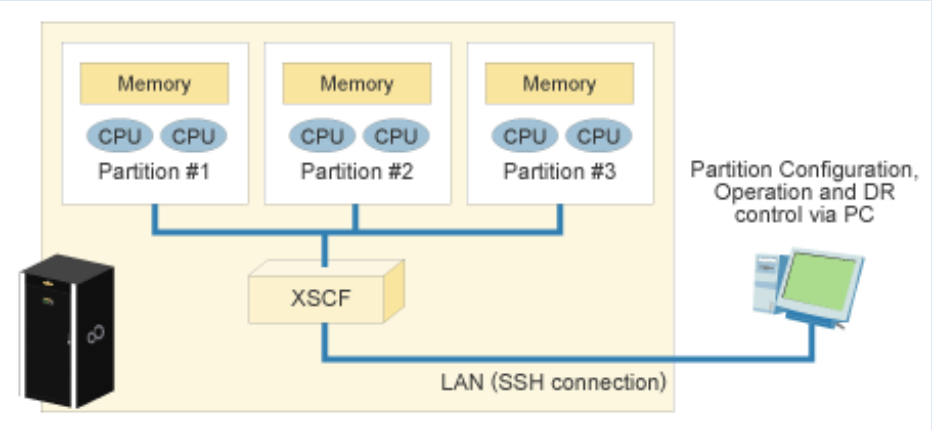
- możliwość przydzielania dodatkowych zasobów w miarę potrzeb
- nieprzerwanie działanie w razie awarii
- równoważenie obciążeń



Effective use of resources in PRIMEPOWER 900

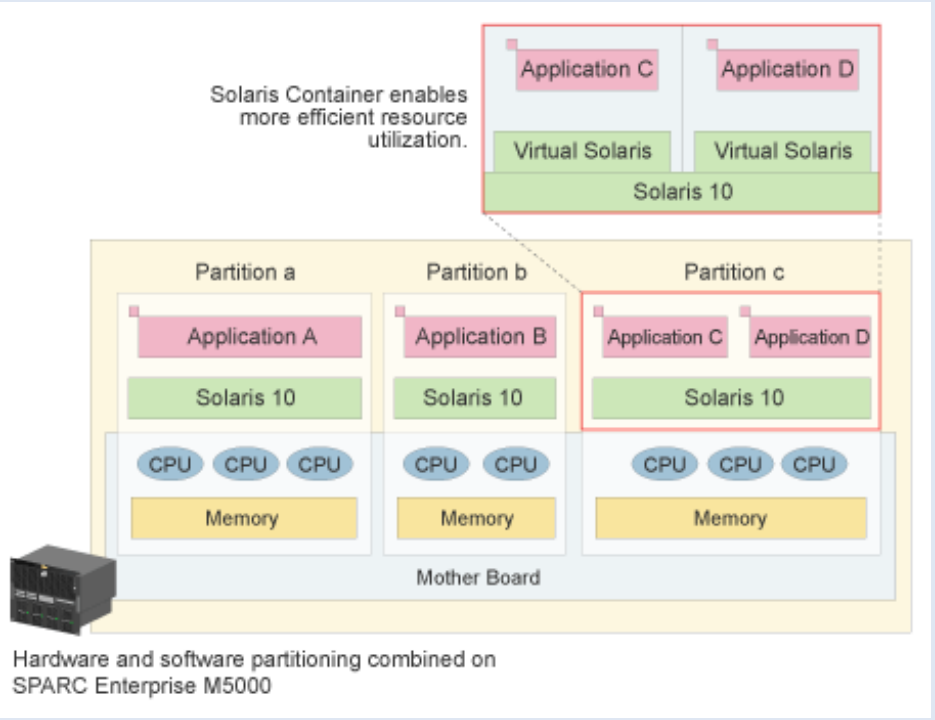
Partycjonowanie sprzętowe:
konfiguracja

- interfejs XSCF (e**X**tended **S**erver **C**ontrol **F**acility)
- Zarządzanie partycjami



Partycjonowanie sprzętowe:
wraz z partycjonowaniem logicznym

- możliwe jest łączenie partycjonowania sprzętowego z logicznym:
  - Solaris containers
    - mechanizm softwareowy z Solaris 10
  - Inne mechanizmy:
    - Linux-VServer
    - FreeBSD Jails



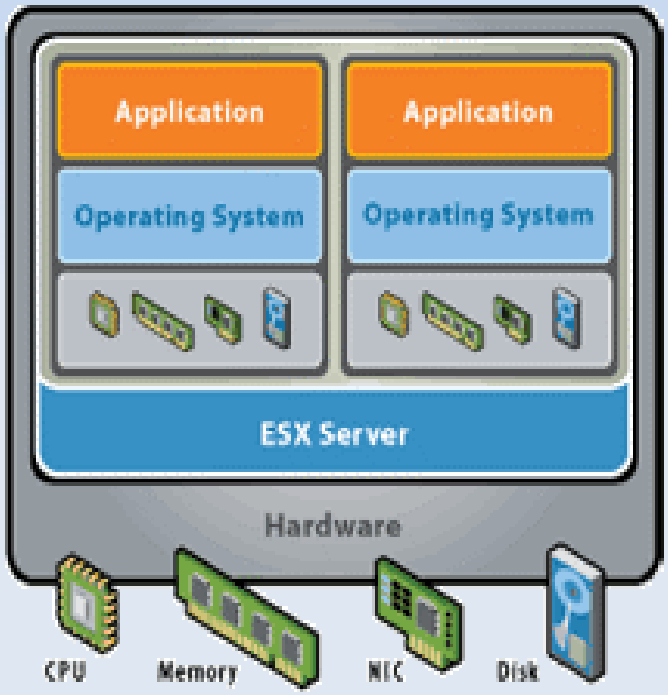
Hardware and software partitioning combined on SPARC Enterprise M5000

Partycjonowanie sprzętowe:
Fujitsu

- Technologia użyta w serwerach Fujitsu M4000-M9000
- procesory SPARC64

VMWare ESX Server

- Systemy:
  - Windows
  - Linux
  - Solaris
- Procesory
  - Intel AMD



- Granularność pamięci: 4kB
- Wirtualne: procesory, pamięć, I/O, dyski. konsole, sieć
- Dostępne połączenia LAN między partycjami
- Daje opcje parawirtualizacji, ale jej nie wymaga

VMWare ESX Server

- Service console:
  - system operacyjny do zarządzania serwerem i używany do uruchamiania systemu
- Guest system:
  - system operacyjny-gość

Sun Logical Domains

- Systemy:
  - Solaris 10+
- Procesory
  - UltraSPARC T1/2

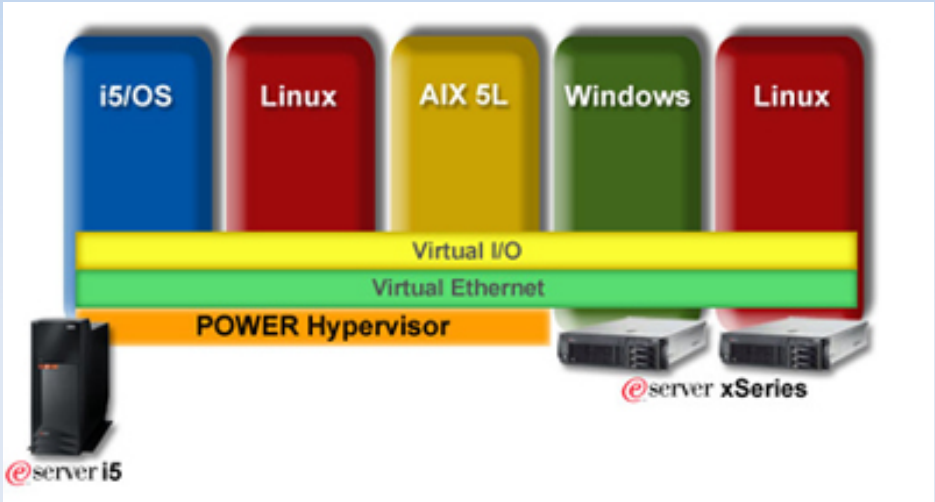
- Granularność pamięci: 8kB
- Wirtualne: procesory, pamięć, I/O, dyski. konsole, sieć
- Dostępne połączenia LAN między partycjami
- Zawiera specjalne partycje kontrolne do konfiguracji

Sun Logical Domains

- Domeny
  - Control domain
    - konfiguracja i zarządzanie systemu
  - Service domain
    - kontrola dysków, LAN
  - I/O domain
    - karty sieciowe, PCI
  - Guest domain
- na ogół control domain, service domain i I/O domain są jedną domeną

IBM pSeries LPAR

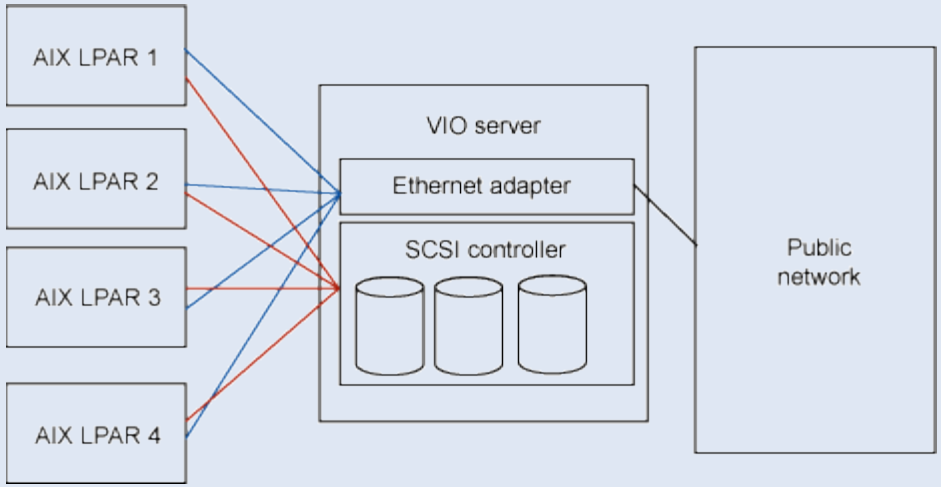
- Systemy:
  - AIX
  - Linux
  - i5 OS
- Procesory
  - POWER4/5



- Granularność pamięci: 16MB
- Wirtualne: procesory, pamięć, I/O, dyski. konsole, sieć
- Dostępne połączenia LAN między partycjami

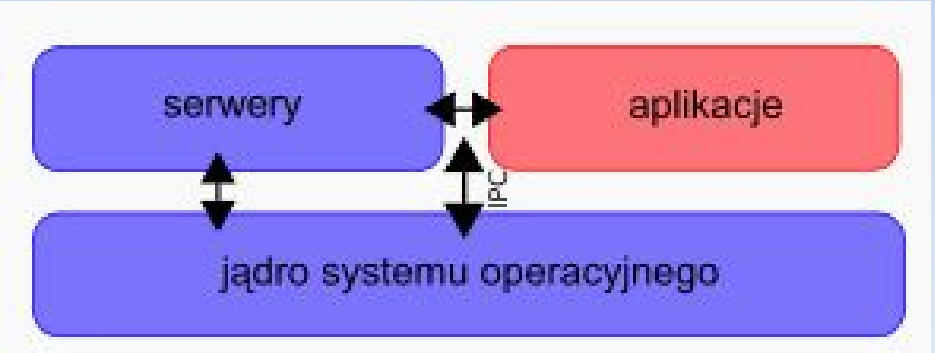
IBM pSeries LPAR

- konsola HMC
- partycja podstawowa
  - funkcje administracyjne
- serwer VI/O
  - funkcje wejścia / wyjścia
    - dyski
    - sieć



L4 Microkernel

- jako przykład hypervisora



- idea: minimalne jądro
- jak najwięcej zadań realizowane jako serwery w przestrzeni użytkownika
- również mogą być to systemy operacyjne jak: Linux

Działanie hypervisera
na przykładzie LDom

- procesor UltraSPARC T1 i 2 mają 3 tryby pracy:
  - user
  - privillaged
  - hyper-privillaged
- Hyperviser używa trybu hyper-privillaged
- mini system operacyjny
  - 374 linii schedulera w Xenie w porównaniu do 5000+ linii scheulera w Linux-ie

Działanie hypervisera
na przykładzie LDom

- Hypervisor jako mini-system operacyjny:
  - odpowiednik schedulera
  - komunikacja między partycjami i urządzeniami (Logical Domains Channel)
  - stronicowanie – 8KB – 16GB
  - obsługuje sygnały
- Udostępnia swoje API systemom operacyjnym na partycjach
  - parawirtualizacja systemów w celu ich efektywniejszego działania

Działanie hypervisera
na przykładzie LDom

- Wywołanie API Hypervisera za pomocą instrukcji **trap** (tcc)
  - numer funkcji jako argument
- 2 konwencje wywołania:
  - fast-trap
  - hyper-fast-trap
- hypervisor udostępnia swoje API w zakresie: partycji, CPU, pamięci, urządzeń, konsoli, PCI I/O, MSI, cache, „core dumpów”, czasu i statystyk

Działanie hypervisera
na przykładzie LDom

- API:
  - wersji – wersje API
  - partycji – operacje dot. całej domeny
  - CPU – operacje na procesorach wirtualnych
  - MMU – translacje adresów pamięci
  - cache&memory – operacje pamięci wirtualnej
  - urządzenia – operacje na urządzeniach i przerwaniach
  - czasu – pobieranie/ustawianie daty i czasu
  - konsoli – operacje konsoli
  - core dump – możliwość robienia 'dumpów'

Działanie hypervisera
na przykładzie LDom

- API (cd):
  - trap trace – możliwość śledzenia wywołań systemowych
  - LDC – komunikacja pomiędzy partycjami i usługami
  - PCI I/O – interfejsy PCI
  - statystyki i wydajność – odczytywanie statystyk procesora

Działanie hypervisera
na przykładzie LDom

- API Domeny:
  - mach_exit - zatrzymuje procesory domeny i zmienia jej stan na 'zatrzymany'
  - mach_sir - wykonuje reset domeny
  - mach_watchdog - ustawia czas wywołań watchdoga
- API CPU:
  - cpu_start/cpu_stop – start/stop CPU
  - cpu_yield – zatrzymuje obliczenia na procesorze
  - cpu_state – pokazuje stan procesora
  - cpu_myid – pokazuje id procesora

Działanie hypervisera
na przykładzie LDom

- API Konsoli (pełne):
  - cons_getchar/cons_putchar – wczytanie/wypisanie znaku
- API czasu (pełne):
  - tod_get/tod_set – wczytanie/ustawienie daty
- Pełne API hypervisera zawiera ok. 50 deklaracji funkcji

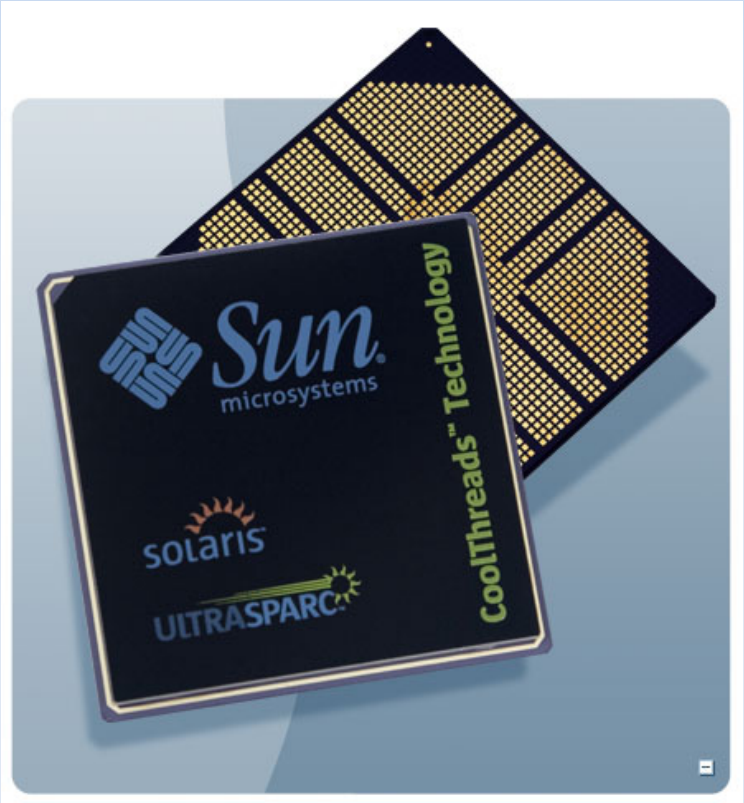
Działanie hypervisera

- Parawirtualizacja systemów przeznaczonych do działania na partycjach logicznych
  - dynamiczna realokacja zasobów
  - obsługa niektórych funkcji systemu-gościa np: sched_yield()
- Linux, Windows może wymagać dodatków umożliwiających parawirtualizację
- AIX, Solaris są parawirtualizowane od początku

Wsparcie sprzętowe

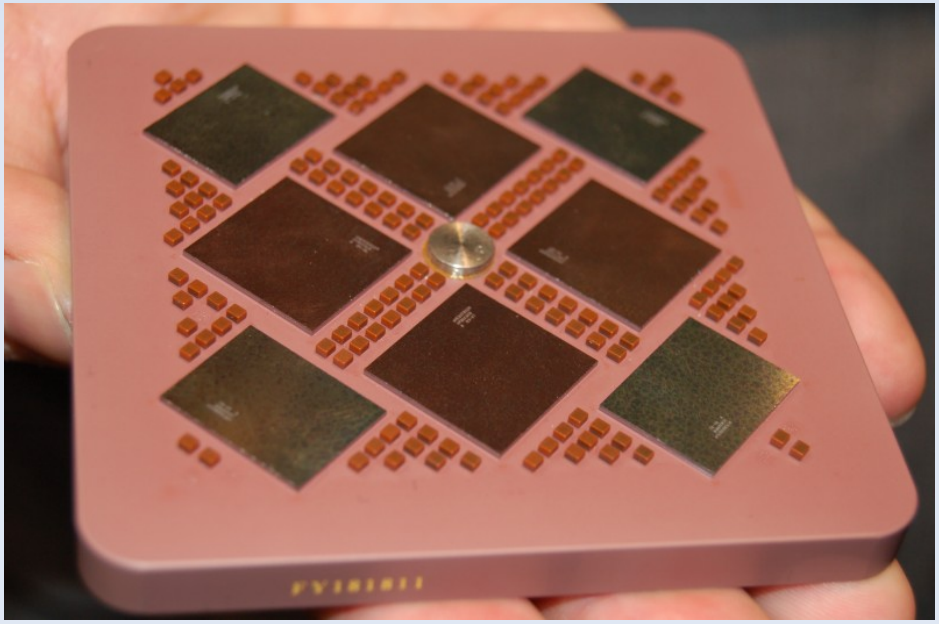
- Procesory:
  - RISC
  - wielowątkowe
    - 4 wątki – POWER5 (2 rdzenie po 2 wątki))
    - 32 wątki – UltraSPARC T1 (4 rdzenie po 8 wątków)
    - 64 wątki – UltraSPARC T2 (8 rdzeni po 8 wątków)
    - 64 wątki – SPARC64 (8 rdzeni po 8 wątków)
- Można procesor podzielić na jednostki obliczeniowe

UltraSPARC T2



- 4MB L2 cache
- 8 rdzeni (po 8 wątków)
  - może być podzielony na 64 partycje

POWER5



- moduł POWER5

- pozwala na utworzenie do 10 partycji
- L3 cache: 4 X 36 MB w module
- 4 procesory w module

Wsparcie sprzętowe

- Niekiedy logiczne partycjonowanie jest dedykowane dla określonej rodziny serwerów:
  - IBM Lpar: iSeries, pSeries
  - Sun LDom: Sun Fire Server, Netra

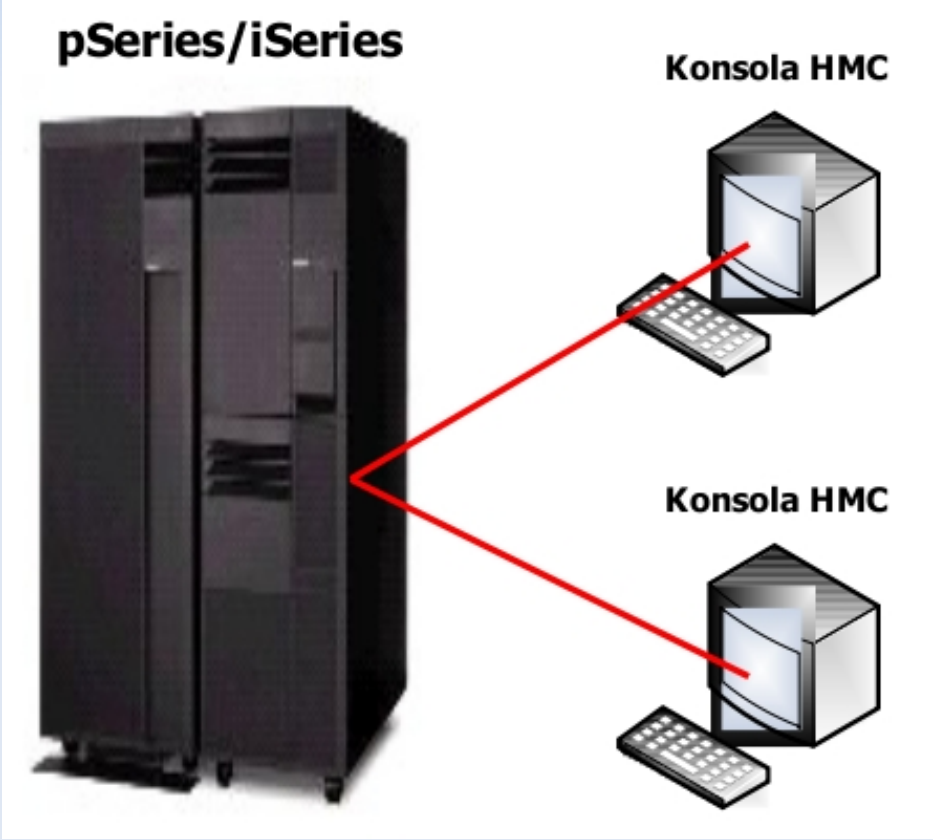
(D)LPAR na podstawie projektu konsolidacji serwerów jednego z banków

- **147 logicznych partycji na 3 serwerach i595**
- 7 LPARów z systemem i5/OS
- 136 LPARy z systemem AIX
- 4 LPAR-y z systemem SUSE Linux Enterprise Server.
- **Używają one jedynie:**
- 79 procesorów POWER5
- 320 GB RAM
- 25 adapterów ethernetowych.

(D)LPAR - składniki:

- **POWER Hypervisor**
- **Hardware Managment Console(HMC)** – konsola użytkownika umożliwiająca zarządzanie LPARami
- **Virtual I/O Server –** oprogramowanie umieszczone na LPAR umożliwiające wirtualizacje urządzeń dyskowych i sieci

Hardware Management Console



- jedna / dwie konsole.
- podstawowy interfejs zarządzania
- program napisany w Javie i uruchamiany na oddzielnym komputerze pod kontrolą Linuksa
- komunikacja pomiędzy komputerem konsoli a serwerem głównym jest realizowana przez TCP/IP
- dostęp do konsoli bezpośredni lub zdalny(warto rozważyć, bo praca w serwerowni prowadzi do kataru)
- dane o konfiguracji LPARów przechowywane na serwerze a nie konsoli.
- backup danych o konfiguracji LPARów wykonywany za pośrednictwem konsoli HMC.
- awaria konsoli nie ma wpływu na działanie serwera.

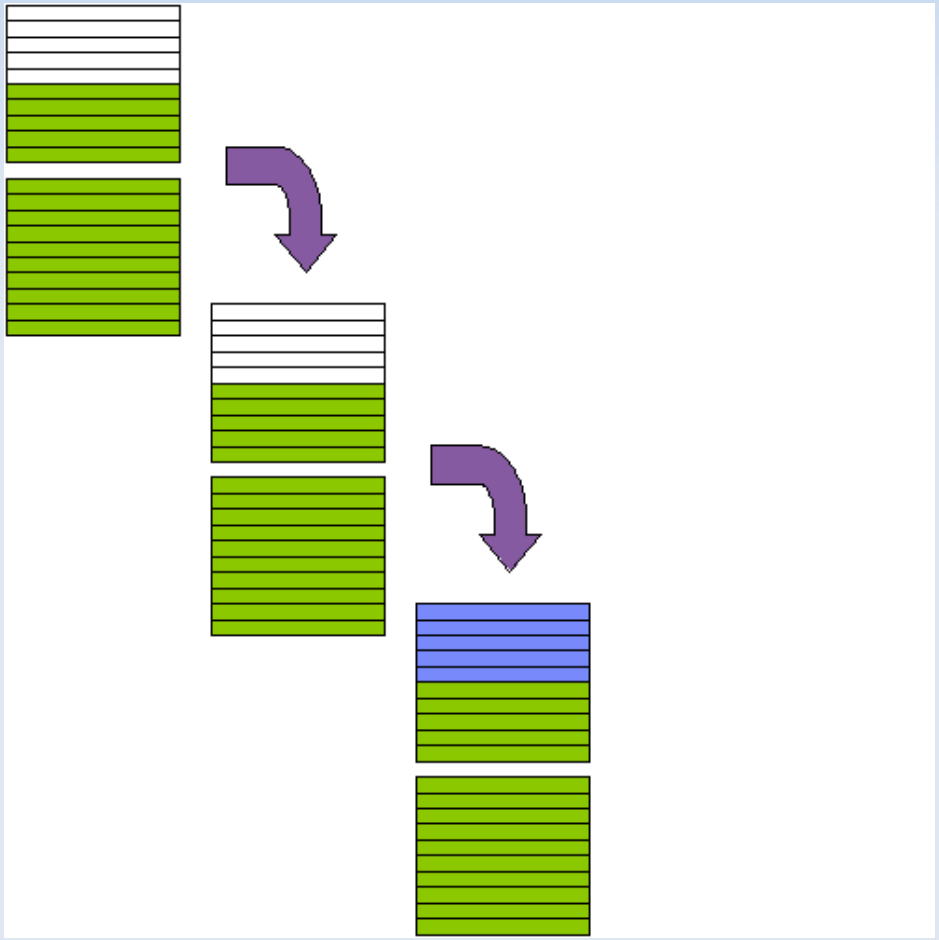
Partycja logiczna

- Procesor
  - zbiór procesorów widziana przez system
  - moc procesorów
  - automatyczna zmiana mocy według potrzeb(procesory typu capped/uncapped)
  - zakres dynamicznych zmian parametrów
- Pamięć
  - ilość pamięci i zakres dynamicznych zmian
- Wirtualne urządzenia(Virtual SCSI/Network)

Pamięć

- minimum
- desired
- minimum 128 MB
- granularność 16 MB(wcześniej 256 MB)

Przydzielanie zasobów przy starcie

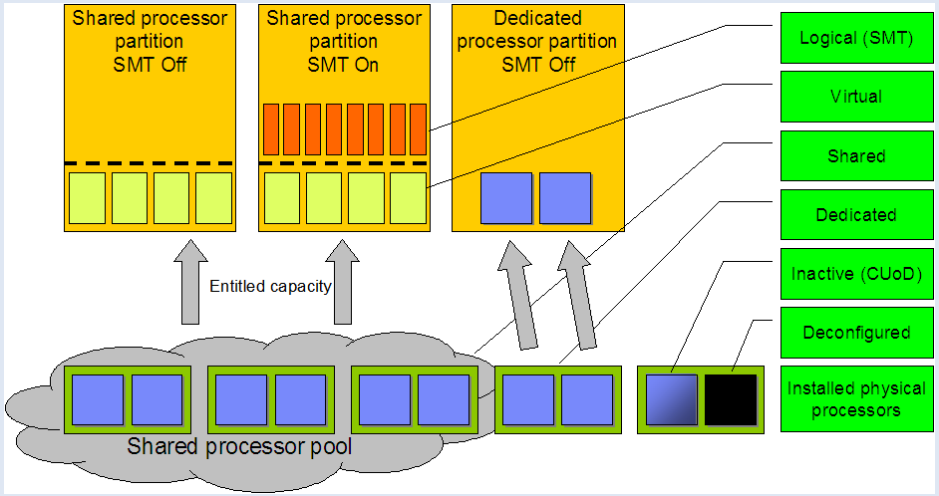


- partycja1
  - min 1.0
  - desired 1.5
- partycja2
  - min 1.0
  - desired 1.0
- partcja3
  - min 0.1
  - desired 0.8

POWER4 vs POWER5

- POWER4
  - tylko całe procesory przydzielane logicznym partycjom
  - ograniczenie na ilość partycji = ilości CPU
- POWER5
  - co najmniej 1/10 na partycje
  - granularność 1:100

Procesor POWER5

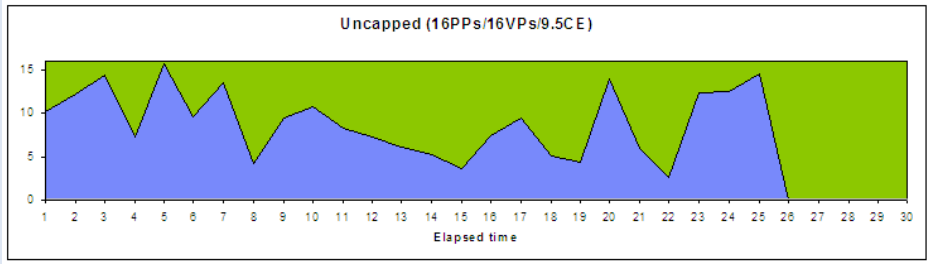


- moc obliczeniowa
- tryb pracy
  - dzielony(ułamki)
  - dedykowany
    - domyślnie nieaktywne trafiają do puli dzielonej
- Wirtualny procesor
  - abstrakcja fizycznego
  - przypisany konkretnej partycji
  - = ilość operacji współbieżnych
  - max 64 na partycje
  - >=0.1 mocy CPU

Partycje typu cappped i uncaped

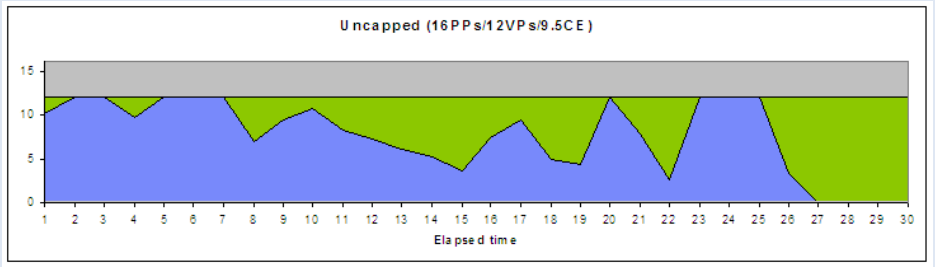
- capped
  - nie można przekroczyć ustalonej pojemności
- uncapped
  - pozwala na przydzielenie większej mocy w razie potrzeby dla danej partycji
  - weight
    - weight = 0 (soft capped)
    - 255 max

Uncapped



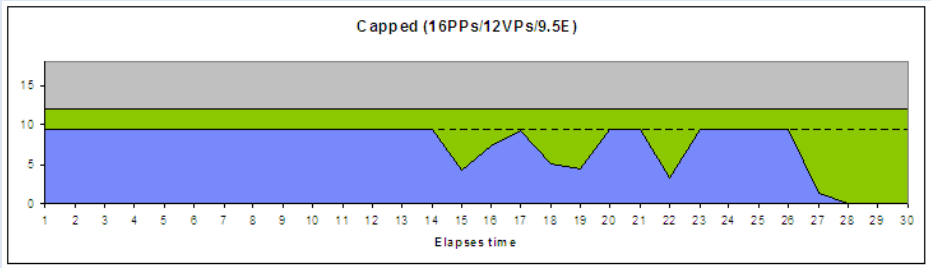
- partycja
  - ustawione na 10
  - 16 wirtualnych procesorów
- 16 fizycznych procesorów dostępnych w puli dzielonej

Uncapped



- partycja
  - ustawione na 10
  - 12 wirtualnych procesorów
- 16 fizycznych procesorów dostępnych w puli dzielonej

Capped



- partycja
  - ustawione na 10
  - 12 wirtualnych procesorów
- 16 fizycznych procesorów dostępnych w puli dzielonej

Virtual I/O Server(VIOS)

- oprogramowanie umieszczone na LPAR, które umożliwia wirtualizacje urządzeń dyskowych i sieci
- Sieci zewnętrzne:
  - dostęp poprzez wirtualne adaptery Ethernet
  - duże zużycie mocy procesora na wirtualizację transferów sieciowych
- Urządzenia dyskowe:
  - możliwość dynamicznego dodawania/usuwania urządzeń dyskowych do/z LPARa
  - niewielkie zaangażowanie mocy procesora w operacje dyskowe oraz mały spadek prędkości obsługi urządzeń dyskowych(kilka procent)

Ograniczenia

- Maksymalna ilość aktywnych partycji na serwer = **254**
- Maksymalna ilość wirtualnych procesorów na LPAR = **64**
- Minimalna ilość mocy procesora na LPAR = **0,1**
- Maksymalna ilość mocy procesora na wirtualny procesor = **1**
- W ramach jednego LPARa procesory wszystkie procesory muszą pracować albo w trybie dedykowanym albo dzielonym

Bibliografia

- http://www-05.ibm.com/
- http://www.fujitsu.com/
- http://opensparc-t1.sunsource.net/specs/Hypervisor-api-curren
- http://unixdays.gda.pl/
- http://www.vmware.com/products/vi/esx/
- http://www.sun.com/servers/coolthreads/ldoms/
- http://www.xensource.com/
- http://os.inf.tu-dresden.de/fiasco/

Dziękujemy za uwagę

- Pytania?
  - A)Nie
  - B)Innym razem
  - C)Drogą mailową
- To dziękujemy.