

РЕСУРСЕН МЕНИДЖМЪНТ (ВИРТУАЛИЗАЦИЯ) НА МНОГОЯДРЕНИ ПЛАТФОРМИ

Проф. Пламенка Боровска
Катедра Компютърни системи
Технически Университет - София



МОТИВИ

- Броят на ядрата в един чип ще се увеличава при всяка следваща технологична генерация
- Наблюдава се тенденция за изместване на универсалните КС от настолните компютри към разнообразни устройства като клетъчните телефони, дигитални центрове за забавления, и сървъри за центрове за данни (data-centers)
- Многоядрените компютри трябва да притежават важните качества на съвременните КС като висока производителност и програмируемост и при това да удовлетворяват ограниченията за разумна цена, консумация на енергия и производителност в реално време
- Важен аспект на многоядрените чипове е подобреното използване на ресурсите (агрегатна производителност) и занижени изисквания към консумираната енергия



МОТИВИ

- Операционните системи управляват общите апаратни ресурси – процесори, памет и В/И
- Съвременните ОС са ефективни за системи, при които процесорите са независими устройства, всеки от които със собствени микро-архитектурни ресурси
- При многоядрените чипове процесорите изпълняват паралелно нишки, които се конкурират за заемане на ресурсите
- Конвенционалните ОС не могат да осигурят ефективен ресурсен мениджмънт за многоядрените чипове



СТРАТЕГИИ И МЕХАНИЗМИ

- Стратегиите осигуряват решения;
- За гъвкавост могат да се използват стратегии имплементирани в софтуера;
- Механизмите осигуряват примитиви за изграждане на стратегиите;
- Примитивите са универсални и следователно, системните проектанти могат да ги имплементират както в хардуера, така и в софтуера;
- Механизмите, които взаимодействат директно с хардуерните ресурси на ниво микроархитектура, се имплементират хардуерно;
- Механизмите, които управляват хардуерните ресурси на ниво макроархитектура, се имплементират софтуерно;

ВИРТУАЛИЗАЦИЯ НА МНОГОЯДРЕНИ ПЛАТФОРМИ

- Концепцията за хардуерен/софтуерен интерфейс базиран на абстракцията за виртуални машини дава възможност за управление в явен вид на ресурсите на микроархитектурата
- VPM (Virtual Private Machines) – стратегиите се имплементират предимно в софтуера като след транслиране на приложението механизмите мултиплексират, арбитрират или разпределят хардуерните ресурси съобразно изискванията на VPM разпределението



Абстракцията VPM

- При традиционните мултипрограмни системи ОС осигурява за всяко приложение (програма) част от физическите ресурси – напр., физическа памет и слайси процесорно време;
- В случая всяко приложение има собствена private машина със съответната памет и изчислителни ресурси;
- При многоядрените чипове, които съдържат споделени ресурси на ниво микроархитектура машината на приложението вече не е private при което използването на ресурсите на ниво микроархитектура от друго приложение може да окаже негативен ефект



Абстракцията VPM

- VPM по принцип са подобни на класическите виртуални машини
- Класическите виртуални машини виртуализират функционалността на системата игнорирайки особеностите на имплементацията
- VPM виртуализират системната производителност и енергийните характеристики, които са специфични за конкретната имплементация



Абстракцията VPM

- VPM обхваща пълния набор от виртуални хардуерни ресурси, както пространствените (физически) ресурси, така и времевите ресурси (слайси време) т.е. споделените ресурси на ниво микроархитектура
- По определение VPM има същата производителност и енергийна консумация като реална машина с еквивалентен набор от хардуерни ресурси
- **Абстракцията VPM осигурява концептуалния интерфейс между стратегиите и механизмите**



Пространствена виртуализация

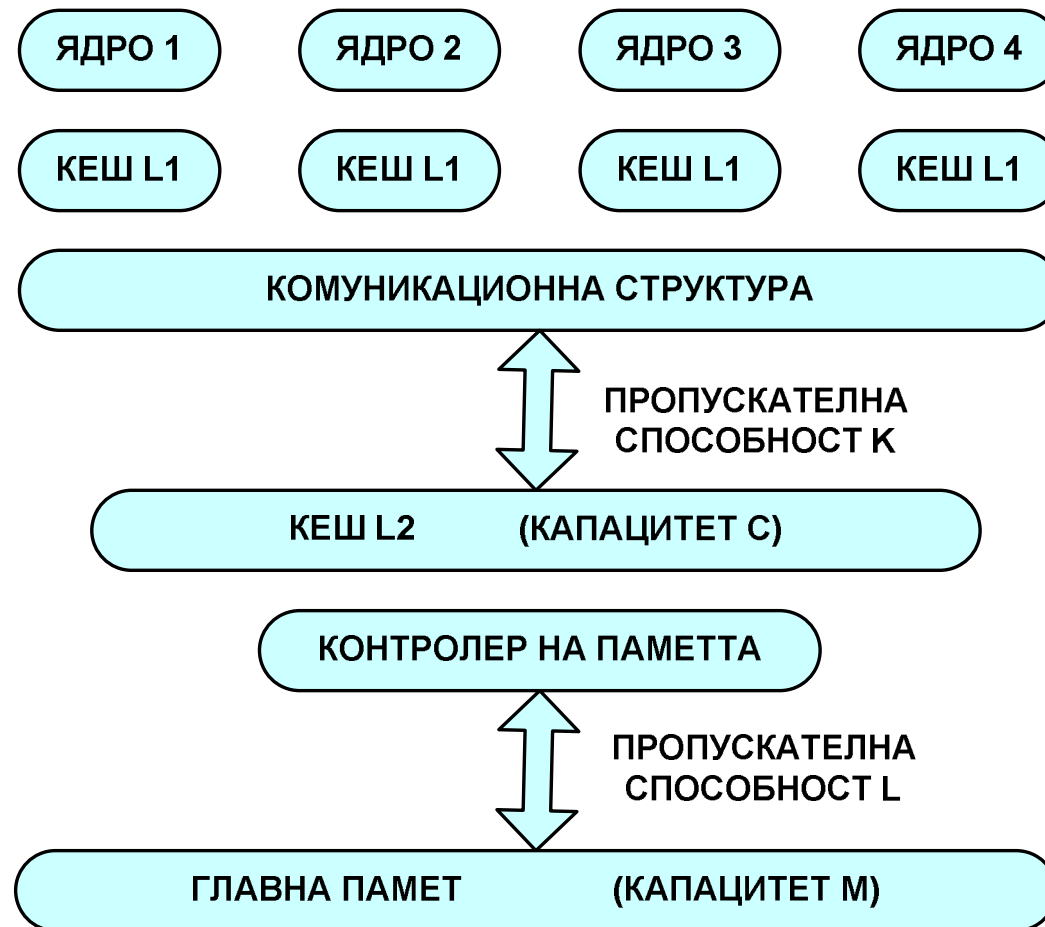
- Пространственият компонент на VPM дефинира частта от системните физически ресурси, които са присвоени на VPM за времето, през което тя изпълнява нишка
- Ресурси на йерархията обща памет, В/И ресурси, вътрешните ресурси на процесора – буфери, опашки, и др.



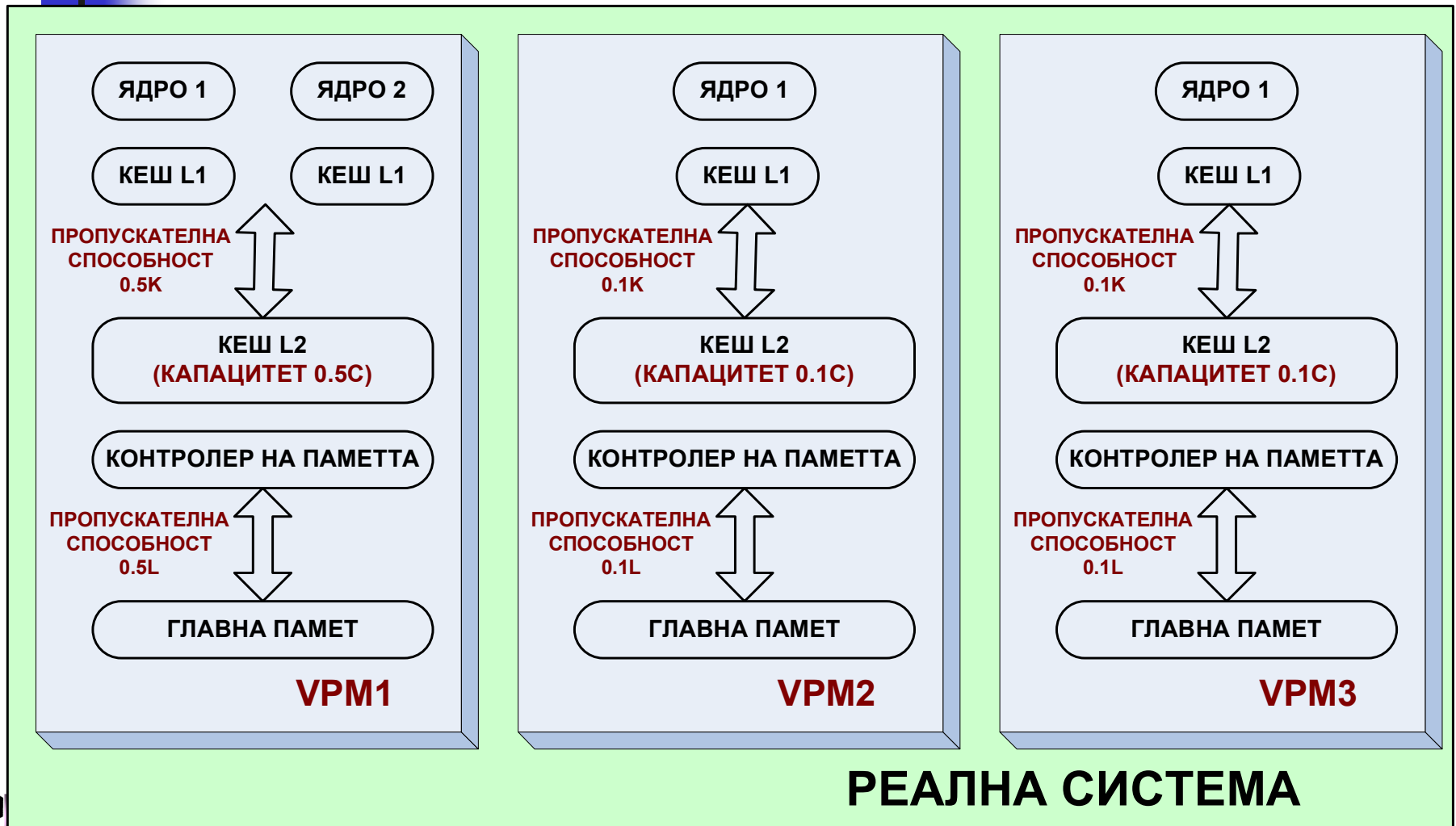
Пример

- Система с 4-ядрен процесор, всеки с индивидуален кеш L1
- Процесорите споделят кеш L2, главната памет и вътрешнопроцесорните комуникационни структури
- Разпределение на ресурсите между три виртуални машини VPM1 – многонишково мултимедийно приложение (50% от ресурсите), VPM2 – 10%, VPM3 – 10%
- 30% от ресурсите на кеша и главната памет не се използват – тези ресурси се наричат свободни ресурси или допълнителна услуга (excess service)
- Стратегиите за използване на свободните ресурси целят подобряване на цялостното използване на системните ресурси и за оптимизация на вторични фактори на производителността

ЧЕТИРИЯДРЕНА ПЛАТФОРМА



ВИРТУАЛИЗАЦИЯ





ПРОСТРАНСТВЕНА ВИРТУАЛИЗАЦИЯ

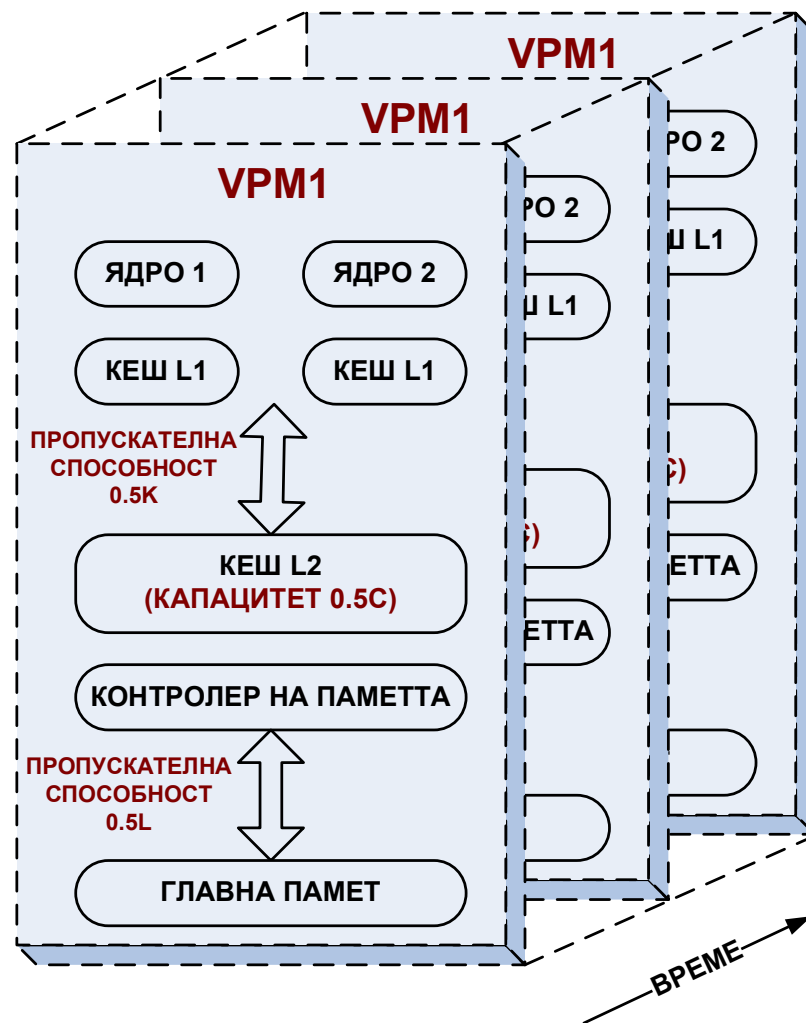
- Пространствената виртуализация поддържа йерархично управление на ресурсите
- Планиращите програми за мултипроцесори могат да се използват за виртуалните машини
- Мултипроцесорните VPM могат да поддържат **рекурсивна виртуализация**



ВРЕМЕВА ВИРТУАЛИЗАЦИЯ

- Концепцията за времевия компонент на VPM се базира на концепцията за идеалното пропорционално споделяне
- Времевият компонент специфицира частта от процесорно време (броят на процесорните слайси), през което пространствените ресурси са предоставени на дадена VPM
- Поддържа рекурсивна виртуализация и йерархично управление на ресурсите, както и могат да съществуват свободни ресурси (excess service)

ПРОСТРАНСТВЕНИ И ВРЕМЕВИ КОМПОНЕНТИ НА ВИРТУАЛИЗАЦИЯТА





Принципи на виртуализацията

- В зависимост от целите стратегиите за виртуализация могат да присвояват на приложенията **минимални или максимални VPM** или и двете
- Механизмите осигуряват на дадено приложение да бъде предложена VPM, която е по-голяма или равна на минималната VPM за приложението
- Прилага се **принципа на монотонност на производителността** – производителността на приложението ще се подобри ако му се предложат допълнителни ресурси
- **Принцип на изолираност** – производителността на приложението е толкова добра, колкото ако то се изпълнява на реална машина, конфигурирана с ресурсите на присвоената VPM

Принципи на виртуализацията

- На практика механизмите предлагат на приложението поне ресурсите на неговата минимална VPM
- Механизмите на виртуализацията трябва да гарантират, че приложението няма да получи повече от ресурсите, заложиени в неговата максимална VPM
- Стратегиите могат да присвояват максимална VPM на приложението с цел управление на консумираната енергия, при което неизползваните ресурси трябва да преминават в състояние на ниска консумация
- Принцип на монотонност на консумираната енергия - консумираната енергия от приложението е монотонно нарастваща функция от използването на ресурсите
- Проектантите на софтуер могат да използват максималната VPM за тестване дали минималната VPM конфигурация удовлетворява изискванията на приложението за производителност в реално време



Стратегии за виртуализация

- **Стратегии на приложно ниво** – удовлетворяват изискванията за качеството на обслужването QoS на приложението като предлагат VPM конфигурация за планиране и управление на ресурсите, които се предоставят на приложението
- **Стратегии на системно ниво** – удовлетворяват системните изисквания за управление на разпределението на ресурсите между VPM на приложенията и отменя разпределението на VPM ресурсите при претоварване на системата



СИСТЕМНА АРХИТЕКТУРА НА VPM

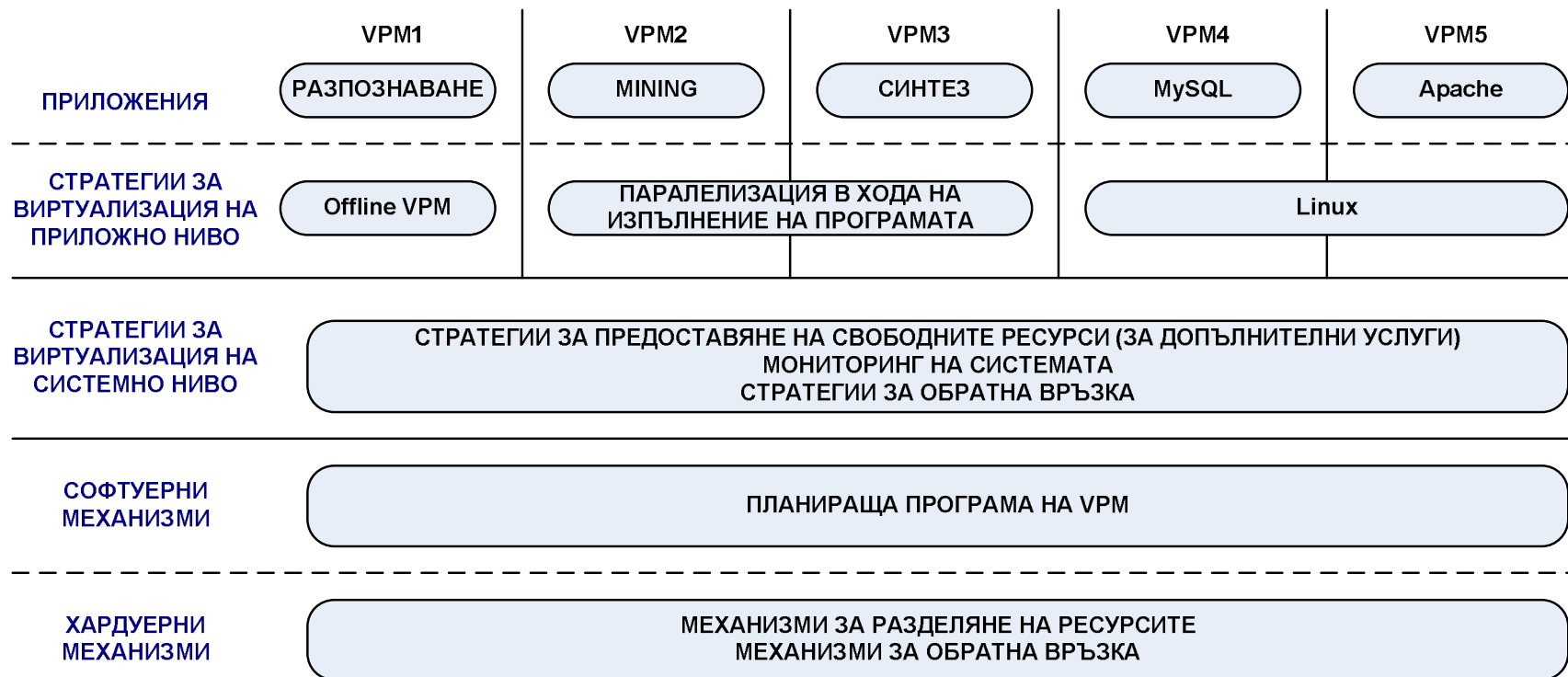
- Стратегиите лесно могат да бъдат заменяни или модифицирани на системна или приложна основа
- Напр., системата изпълнява 5 приложения, всяко с присвоена VPM:
- Приложение 1 за разпознаване в реално време – използва VPM, която е създадена от програмиста offline
- Приложения 2 и 3 (data mining и синтез) – на език за паралелно програмиране, включващ изчисляване на VPM в хода на изпълнение на програмата (runtime) и оптимизиращ паралелизмите на приложението в съответствие с ресурсите на присвоената VPM
- Приложения 2 и 3 са изолирани едно от друго, но споделят общи библиотеки



СИСТЕМНА АРХИТЕКТУРА НА VPM

- Приложения 4 и 5 MySQL и Apache са стандартни Linux приложения, които се изпълняват на паравиртуализирана версия на Linux
- Паравиртуализираната версия на Linux се изпълнява на програмен слой, който прави мониторинг на характеристиките на работния товар на приложението и изчислява конфигурациите на VPM с помощта на евристични и ad hoc техники
- Стратегиите за виртуализация на приложно ниво дават възможност на приложните програмисти да създадат профил на поведението на системата за удовлетворяване на изискванията на специфициран приложен спектър

VPM СИСТЕМНА АРХИТЕКТУРА



СТРАТЕГИИ ЗА ВИРТУАЛИЗАЦИЯ НА ПРИЛОЖНО НИВО

- Двете фази при определяне на VPM разпределенията са моделиране и транслиране
- На практика двете фази могат да бъдат комбинирани
- Стратегиите за виртуализация на приложно ниво изчисляват конфигурациите на VPM автоматично (online) или ръчно (offline)
- Стратегиите online за виртуализация на приложно ниво се основават предимно на анализ в хода на изпълнение на програмата (runtime) като например могат да използват броячите за производителност на процесора (performance counters)
- При стратегиите offline програмистът трябва да направи анализа на програмата
- Използват се и хибридни стратегии – online/offline



Фаза на моделиране

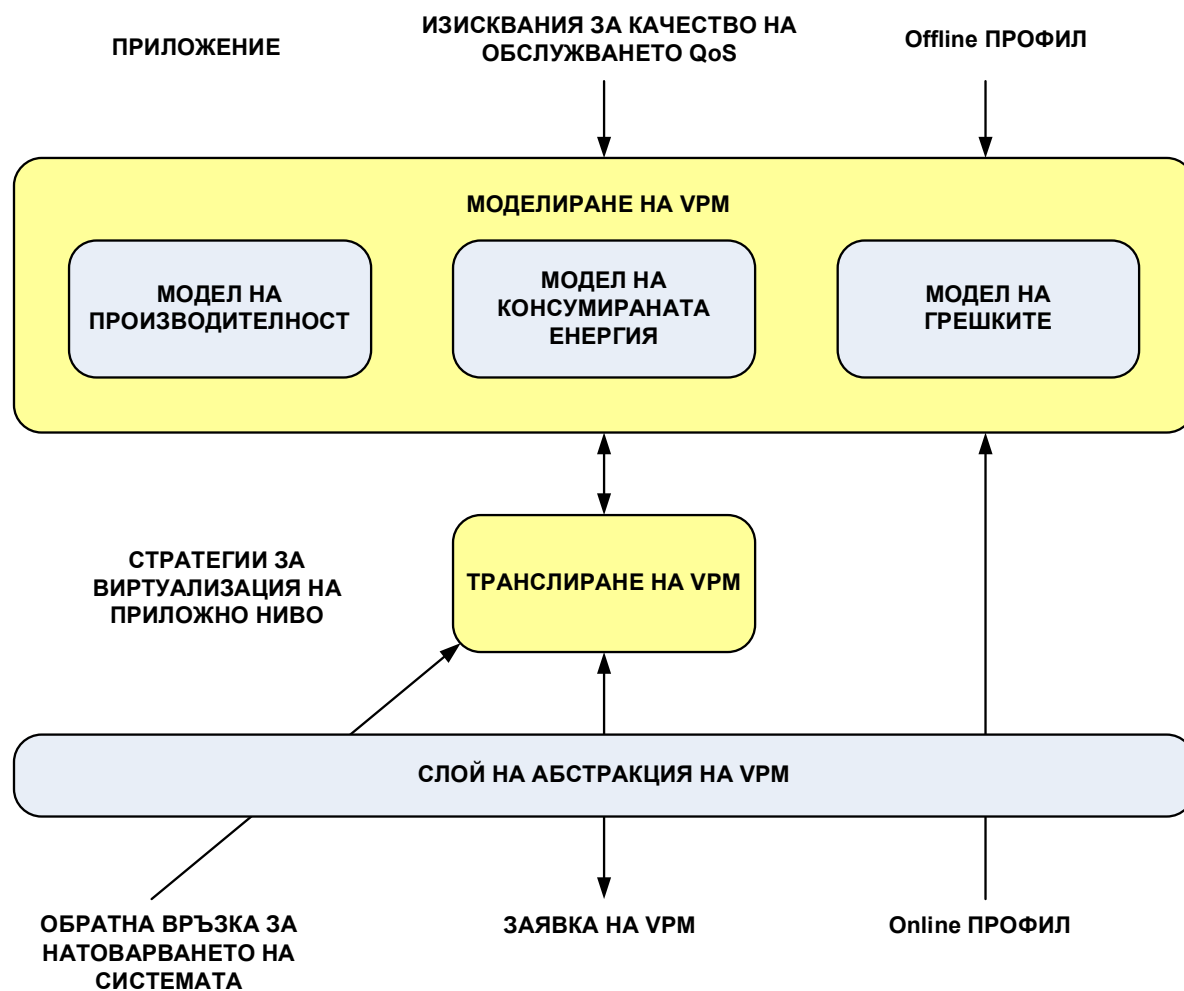
- Изгражда се съответствие (mapping) между изискванията на приложението за качество на обслужването QoS като минимална производителност и максимална консумирана мощност, и конфигурацията на виртуалната машина
- Използва се широк спектър от техники за моделиране:
 1. Прости универсални аналитични модели, основани на обща профилираща online информация за прогнозиране на производителността и консумираната енергия при различни конфигурации на виртуалната машина
 2. Offline профилиране и характеризиране
 3. Специализирани модели на специфичното приложение или на приложната област



Фаза на транслиране

- Създадените модели се използват за намиране на **множеството конфигурации на виртуалната машина**, които удовлетворяват изискванията на приложението за качествено обслужване
- Пространствената компонента на VPM може да съдържа различни комбинации от ресурси
- Някои конфигурации могат да съдържат по-голяма пространствена компонента и по-малка времева компонента и обратното
- За прецизиране и ограничаване на множеството от VPM конфигурации може да се използва многоцелева оптимизация (напр. максимизиране на производителността в реално време и минимизиране на консумираната енергия)

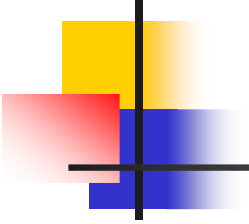
Стратегия за виртуализация на приложно ниво





Виртуализация

- След като стратегията за виртуализация на приложно ниво открие подходяща конфигурация за VPM, тя инициира заявка към стратегията за виртуализация на системно ниво за предоставяне на VPM ресурсите
- Ако ресурсите са налични, те се предоставят на виртуалната машина
- Ако системата е претоварена заявката се отказва или предишното предоставяне на ресурсите се отменя



Реконфигурация на разпределението на ресурсите

- Стратегията за виртуализация на приложно ниво трябва да имплементира процедура за отказ и отменяне на предоставянето на ресурси на виртуалните машини
- В случаите на отказ или анулиране на предоставяните ресурси приложната стратегия трябва да открие други подходящи конфигурации на VPM или да реконфигурира приложението с цел намаляване на изискванията за системни ресурси
- Следователно, **оптимизирането се прави по три измерения – производителност, консумирана енергия и налични ресурси (динамичен профил на работния товар в системата) → многоцелева (многофакторна) оптимизация**

Виртуализацията като комбинаторен проблем

- На практика за много приложения намирането на оптималната конфигурация на виртуалната машина е NP-трудно
- Следователно, фазата на online транслирането в общия случай трябва да използва приблизителни или евристични оптимизационни техники
- За приложения с критични изисквания към QoS приложният програмист трябва да извърши част от транслирането offline
- Моделирането и транслирането могат да бъдат обединени в една фаза – приложният програмист може да използва ресурсите на максимална VPM и да направи опити с намаляване на ресурсите до намирането на минималната VPM, удовлетворяваща изискванията за производителност в реално време



Слой на абстракцията на VPM

- Високо ниво на абстракция – виртуалната машина обхваща дялове от кеша и неговата пропускателна способност, и пропускателната способност на главната памет
- На практика реалните апаратни ресурси са много по-сложни
- Механизмите на VPM не правят абстракция на апаратните ресурси, а само предават специфичните детайли на имплементацията на стратегиите – създава се интерфейс, който е по-ефективен от абстракциите на високо ниво



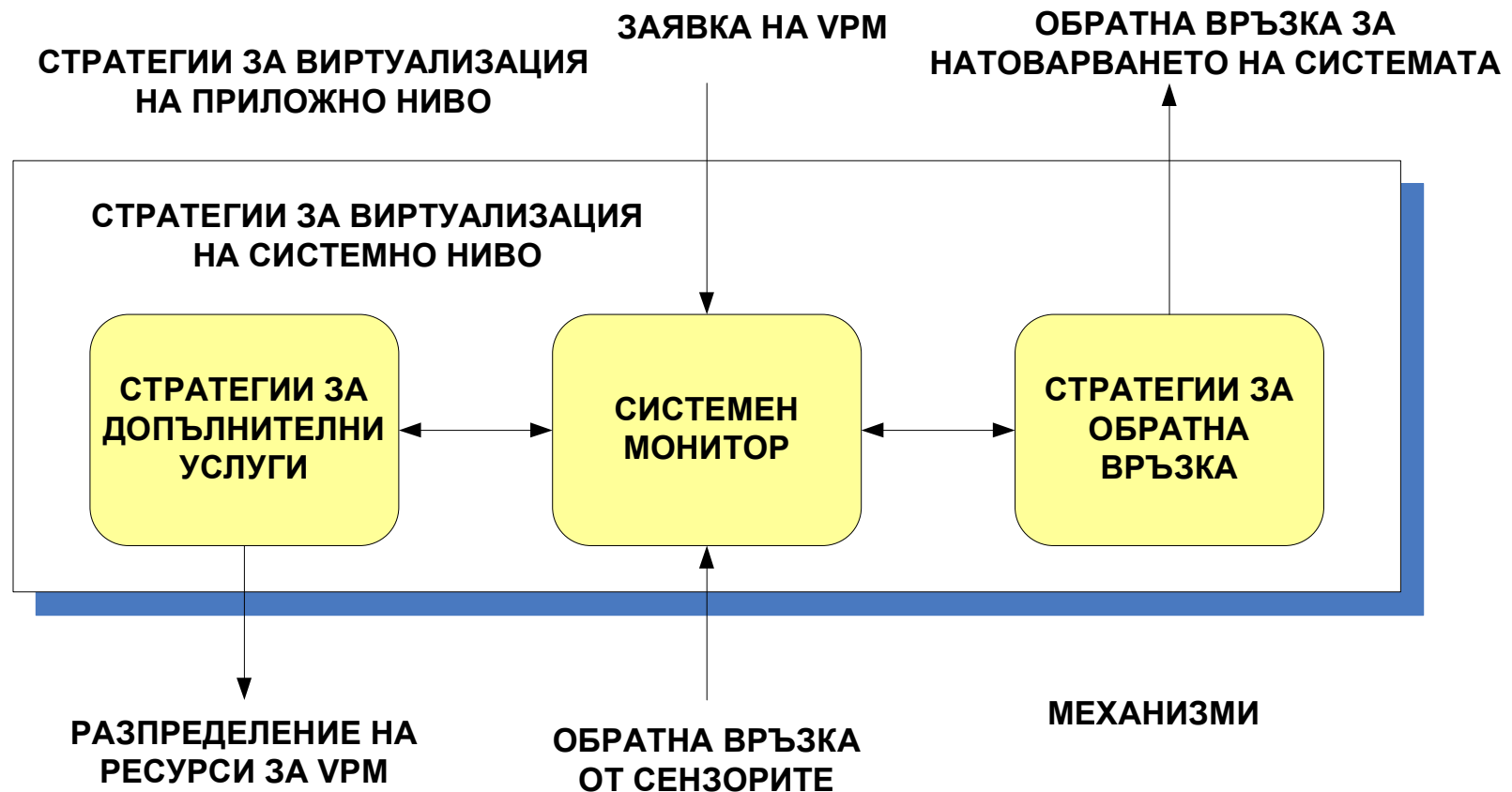
Слой на абстракцията на VPM

- От друга страна при съобразяването с детайлите на имплементацията съществува риск приложенията да станат зависими от имплементацията
- За много приложения не са необходими детайли от ниско ниво за удовлетворяване на изискванията за QoS
- За повишаване на съвместимостта на приложенията се използва стандартен слой на абстракция на виртуалната машина
- Стандартният слой на абстракция на виртуалната машина игнорира специфичните детайли на имплементацията и взема предвид атрибутите на системната производителност и енергийните характеристики
- По време на изпълнението на програмата (runtime) слойът на абстракция ефективно транслира конфигурациите на високо ниво на виртуалната машина в присвояване на ресурси на ниско ниво, което се поддържа от системните механизми

Стратегии за виртуализация на системно ниво

- Те управляват разпределението на ресурсите между виртуалните машини като изпълняват функциите на брокер на заявките на виртуалните машини на приложенията
- Системните стратегии имат три базови компонента: системен монитор, стратегии за обратна връзка и стратегии за допълнителни услуги (предоставяне на свободните ресурси)

Стратегии за виртуализация на системно ниво





Системен монитор

- Проследява натоварването на системните ресурси и детектира ситуации на претоварване на системата
- Системата е претоварена ако не може да осигури достатъчно ресурси за удовлетворяване на изискванията на виртуалните машини на приложенията
- Препоръчително е детектирането на системно претоварване да се осъществява по възможност най-рано, още в етапа на инициране на заявката за VPM на приложението, която предизвиква претоварване т.е. преди системата да се е претоварила



Системен монитор

- За детектирането на претоварване мониторът трябва да проследява натоварването на всеки споделен ресурс като напр. капацитет на кеша, изчислителната пропускателна способност, капацитета на охлаждането и консумираната мощност
- За да се провери дали дадена заявка на VPM претоварва системата може да се прави тест за допустимост като напр. сравнение на множеството присвоени ресурси на VPM с капацитета на физическите системни ресурси



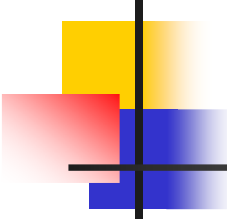
Системен монитор

- Моделите на VPM, които използва системния монитор, се използват като допълнение на сензорите за напрежение и температура
- Ако сензорът детектира претоварен ресурс, това означава, че системните стратегии са приели повече заявки на виртуалните машини отколкото физическите ресурси на системата могат да удовлетворят
- В случаите на претоварване системните стратегии трябва да отменят предоставянето на претоварените ресурси на виртуалните машини на приложенията



Стратегии за обратна връзка

- Определят на кои виртуални машини трябва да бъде отказано или отменено предоставянето на претоварени ресурси
- Информират приложенията за причините за отказа или отменянето
- Стратегиите на приложно ниво могат да използват предоставената информация за създаването на нова виртуална машина адекватна на наличните физически ресурси
- Всяко приложение трябва да получава актуална информация за натоварването на системните ресурси като по този начин стратегиите за обратна връзка могат да направляват глобалната оптимизация на използването на системните ресурси



Стратегии за допълнителни услуги

- След разпределението на системните ресурси между изпълняваните приложения е възможно да останат неизползвани ресурси
- Неизползваните ресурси могат да бъдат ресурси, които не са присвоявани на приложения или ресурси, които са присвоени на приложения, но не се използват от тях
- Стратегиите за допълнителни услуги осигуряват допълнителни ресурси с цел намаляване на времето за отговор, агрегатната пропускателна способност, икономия на енергия, съхраняване живота на транзисторите или комбинация от тях
- Неизползваните ресурси преминават в състояние на ниска консумация

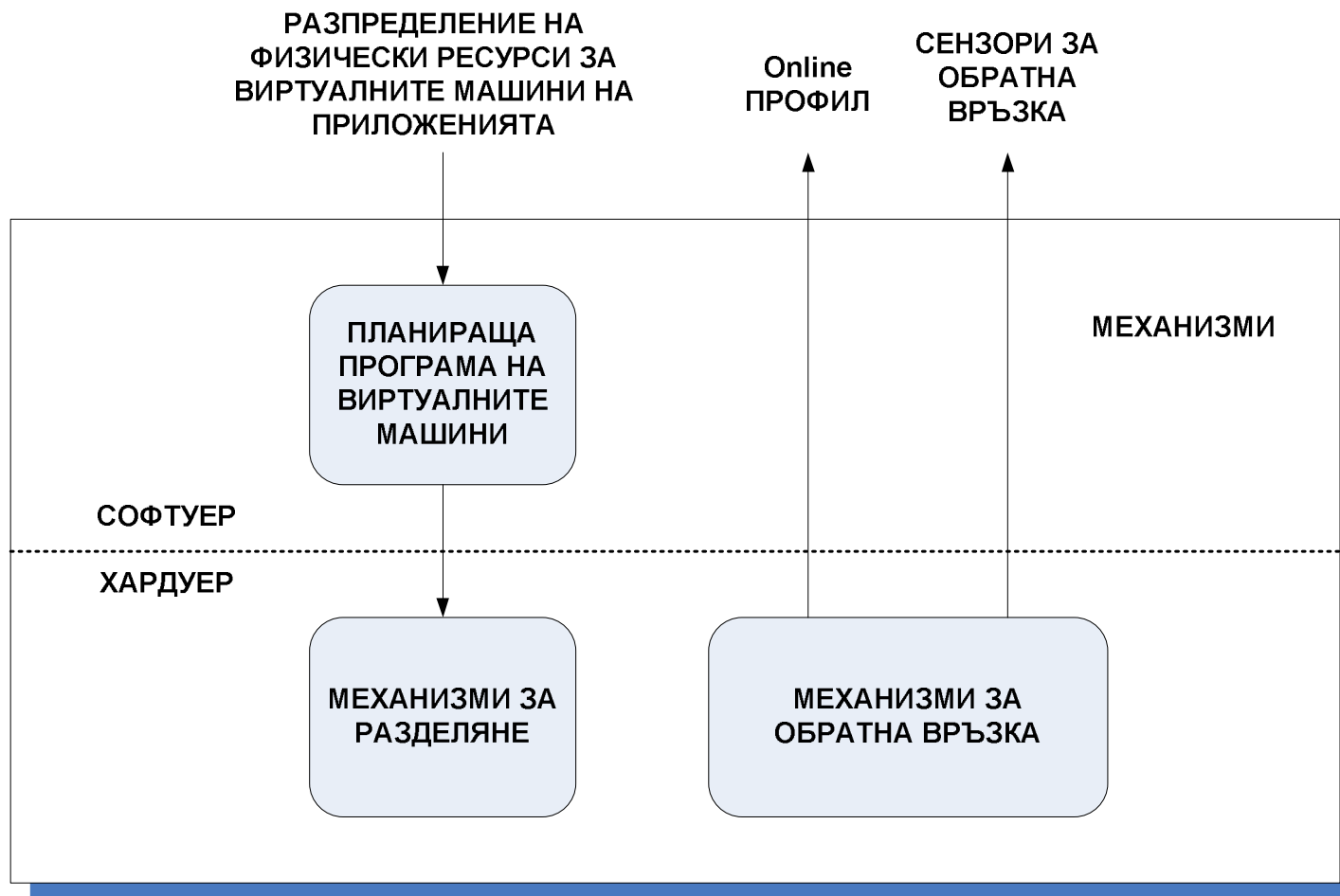
Стратегии за допълнителни услуги

- Предоставянето на допълнителни услуги не трябва да води до превишаване на изискванията на максималната виртуална машина на приложението
- Използват се за донастройка на разпределението на ресурсите – могат да разширят предоставяните ресурси за минималните VPM и да намалят ресурсите за максималните VPM (прозрачно) без да нарушават разпределенията на приложните стратегии
- При допълнителното предоставяне на някои ресурси трябва да бъдат информирани стратегиите на приложно ниво, особено планиращата програма на нишките при предоставянето на допълнителни ядра за VPM

Механизми на виртуалните машини

- Три базови типа механизми поддържат работната рамка с виртуални машини на приложенията: планираща програма за виртуалните машини (VPM scheduler), механизъм за разделяне (partitioning) и механизъм за обратна връзка (feedback)
- Планиращата програма и механизъмът за разделяне мултиплексират, арбитрират или разпределят хардуерните ресурси за удовлетворяване на изискванията на виртуалните машини на приложенията при висока степен на сигурност
- Механизъмът за обратна връзка осигурява взаимодействието с приложенията и стратегиите за виртуализация на системно ниво

Механизми на виртуалните машини





Планираща програма

- Удовлетворява времевите изисквания на разпределенията за виртуалните машини като предоставя слайси време на хардуерните нишки
- Тя е пропорционално справедлива (p-fair), но трябва да осигури безконфликтното планиране на едновременно изпълнявани приложения
- Пространствените ресурсни разпределения трябва да съответстват на наличните физически ресурси и да не се допуска припокриване на приложенията при ресурсите на микроархитектурата
 - Планирането на виртуалните машини при пропорционална справедливост без пространствени конфликти е обект на интензивни изследвания
 - Трябва да осигурява ефективна поддръжка на йерархично планиране



Механизми за разделяне

- Управляват споделените ресурси на микроархитектурата
 - Ресурсите на микроархитектурата могат да бъдат следните основни типове: памет, буфер или ресурс с пропускателна способност
 - За всеки тип ресурс е осигурен съответен основен тип на механизъм за разделяне, напр. нишковите алгоритми за замяна разделят ресурсите памет (главната памет и кешовете), механизмите за управление на потока разделят буферните ресурси, арбитрите за формиране на трафика и опашките арбитрират използването на изпълнителните портове и портовете на паметта

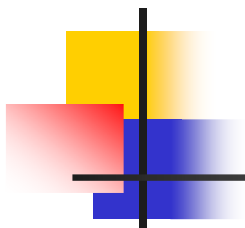


Механизми за разделяне

- За разпределения за максимални VPM механизмите за разделяне трябва да бъдат комбинирани с механизми, които привеждат в състояние на ниска консумация ресурси, които не се използват, напр. неизползвани фази на конвейера и неактивни ресурси памет
- Управлявайки консумацията на енергия на ресурсите стратегиите за виртуализация могат да управляват важни системни характеристики като температурата на кристала и износването на транзисторите

Механизми за обратна връзка

- Осигуряват обратна връзка за приложенията относно капацитета на физическите ресурси и тяхното използване
- Осигуряват информация за стратегиите на системно ниво за капацитета на системните ресурси и наличните механизми за разделяне
- Осигуряват информация за стратегиите на приложно ниво за използването на ресурсите от индивидуалните приложения
- Осигуряват информация за системните стратегии относно цялостното използване на системните ресурси като температура на кристала и консумирана енергия



КРАЙ