

# ПАРАЛЕЛНИ КОМПЮТЪРНИ АРХИТЕКТУРИ

ПРОФ. ПЛАМЕНКА БОРОВСКА



# ОСНОВНИ НАПРАВЛЕНИЯ

- Еволюционни паралелни архитектури – базирани на фон Ноймановата архитектура
- Радикални паралелни архитектури – драстично се различават от фон Ноймановата архитектура по фундаменталните механизми за управление и данни

# Механизъм за управление

- **От инструкции** – фон Нойманова архитектура (**instruction-driven**) – програмата е управляващ инструкционен поток (instruction flow)
- **Атрибути** – *програмен брояч, оператор за присвояване, и управляващата променлива на цикъла*, формират т.нар. “усмирителна ризница” на фон Нойман, т.к. задължително сериализират изпълнението дори на потенциално паралелни участъци на кода



# DATA FLOW ARCHITECTURES

- **Управление от данни** – фон Нойманова архитектура (**data-driven**) – програмата се управлява от потока данни (data flow architectures)
- **Атрибути** – липсват *програмен брояч, оператор за присвояване, както и управляващата променлива на цикъла.*
- Програмата се представя като насочен граф, възлите на който представят операциите, а дъгите – операндите



# DATA FLOW ARCHITECTURES

- **Основен принцип** – дадена операция се изпълнява (fired), когато са готови нейните операнди
- Няма ограничения за паралелно изпълняваните операции
- Архитектурната концепция е предложена от Джек Денис МІТ (Технологичен Институт на Мазачусетс)
- Статични и динамични data flow архитектури



# DATA FLOW ARCHITECTURES

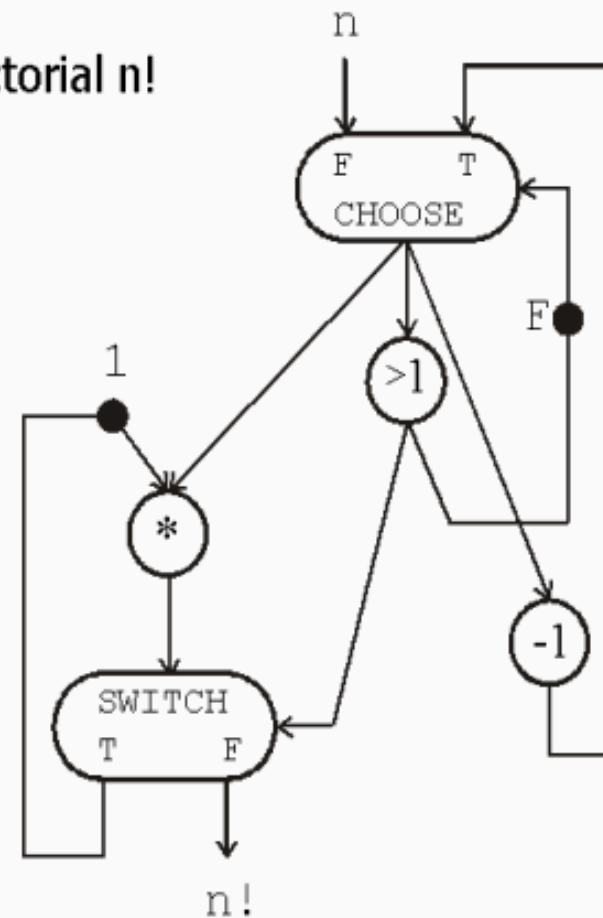
- Доминантната изчислителна парадигма от 1990-те год. е с промяна на реда на изпълнение на инструкциите (out-of-order execution) и по същество представлява ограничена форма на data flow принципа – от 2 до 6 изпълнителни устройства и капацитет на инструкционния прозорец от 32 до 200 инструкции



## Example in Id

- Id program segment computes factorial  $n!$  of integer  $n$

```
( initial j <- n; k <- 1  
  while j > 1 do  
    new j <- j - 1;  
    new k <- k * j;  
  return k )
```



# Редукционни архитектури

- Базират се на функционалните езици за програмиране, в които основната структурна единица е функция от висок ред
- Функция от висок ред – аргументите на функцията са също функции
- Програмата се представя под формата на стринг – функция от висок ред



# Принцип на “мързеливите” изчисления

- Lazy evaluation
- Demand- driven – механизъм за управление от заявки
- Една функция се активира за изпълнение, само когато нейният резултат се използва като аргумент на друга активирана функция

# КЛАСИФИКАЦИЯ НА КОМПЮТЪРНИТЕ АРХИТЕКТУРИ НА МАЙКЪЛ ФЛИН



- Michael J. Flynn (IBM, Университет на Станфорд)
- 1992 Eckert-Mauchly Award Recipient
- Предлага най-обща класификация на еволюционните паралелни архитектури, на основата на инструкционните потоци и потоците данни
- Класификация - 4 архитектурни класа

# Класификация на Флин

- Архитектурен клас **SISD** (**S**ingle **I**nstruction **F**low **S**ingle **D**ata **F**low) – единичен поток инструкции, единичен поток данни – фон Нойманова архитектура
- Архитектурен клас **MISD** (**M**ultiple **I**nstruction **F**lows **S**ingle **D**ata **F**low) – единичен поток инструкции, множествени потоци данни – спорен клас



# Класификация на Флин

- Архитектурен клас **SIMD** (**S**ingle **I**nstruction Flow **M**ultiple **D**ata Flows) – единичен поток инструкции, множествени потоци данни – векторни процесори
- Архитектурен клас **MIMD** (**M**ultiple **I**nstruction Flows **M**ultiple **D**ata Flows) – множествени потоци инструкции, множествени потоци данни – мултипроцесори (с обща памет) и мултикомпютри (без обща памет)



# СКАЛИРУЕМИ ПАРАЛЕЛНИ АРХИТЕКТУРИ (SCALABLE PARALLEL ARCHITECTURES)

## ИЗМЕРЕНИЯ НА СКАЛИРАНЕТО (МАЩАБИРАНЕТО):

- **Мащабиране на ресурсите:** (модулно увеличаване на ресурсите на паралелния компютър) - увеличаване на размера на машината т.е. увеличаване на броя на процесорите, капацитета на паметите, кешовете, дисковете, усъвършенстването на програмното осигуряване и т.н.



- **Мащабирането на приложението** - повишаване на производителността при решаването на проблеми с по-големи обеми на обработваните данни и изобщо, при по-голям работен товар;

*Изпълнението на дадено приложение на конкретен паралелелен компютър формира т.нар. "двойка приложение/машина" и се нарича паралелна система.*



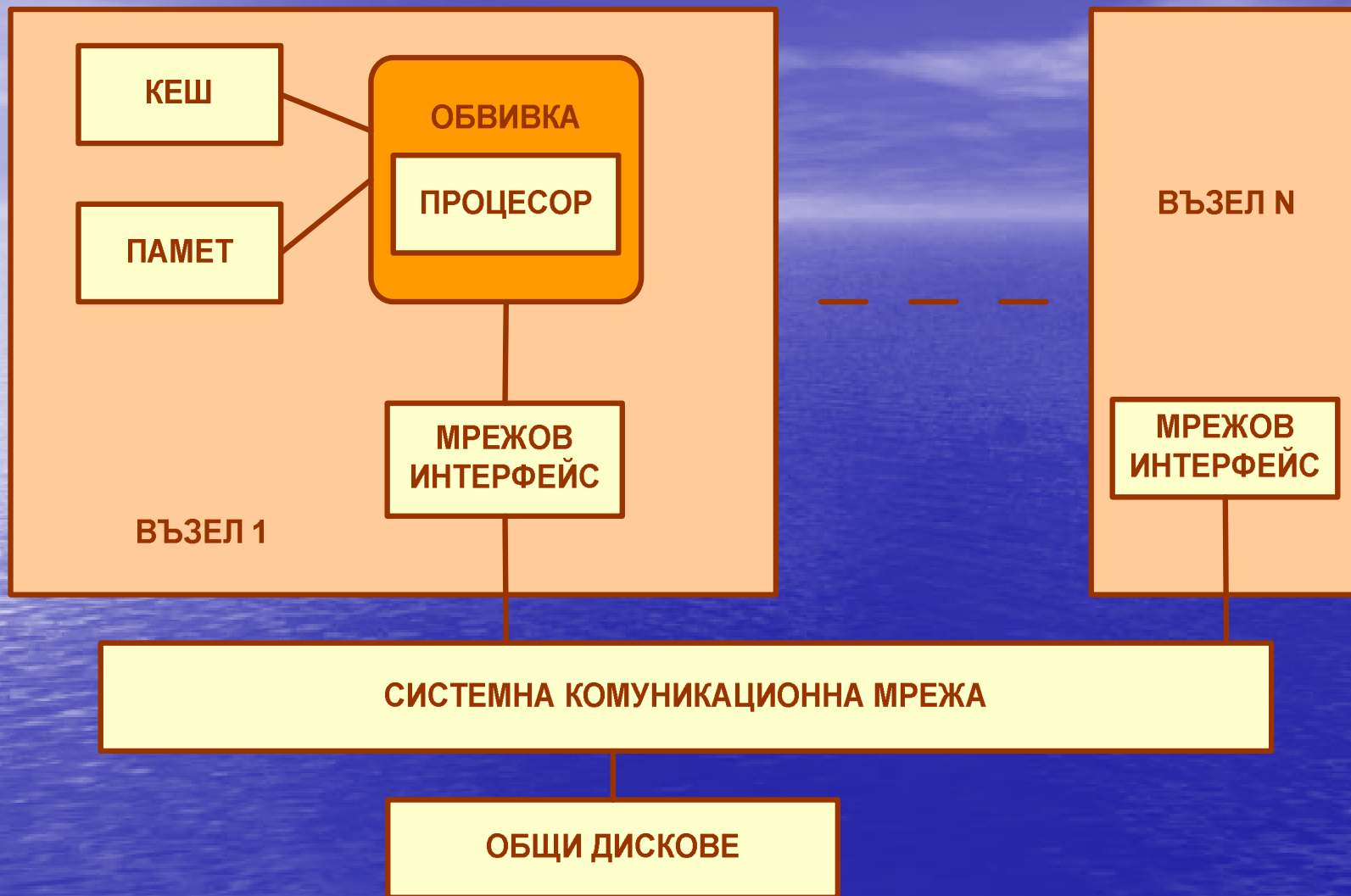
- **Технологичното мащабиране** - осигурява възможност за адаптиране на системата към иновационните технологии: *3 аспекта: мащабиране на генерацията, пространството и хетерогенността*



# Архитектурни стилове на паралелни компютри с мащабиране

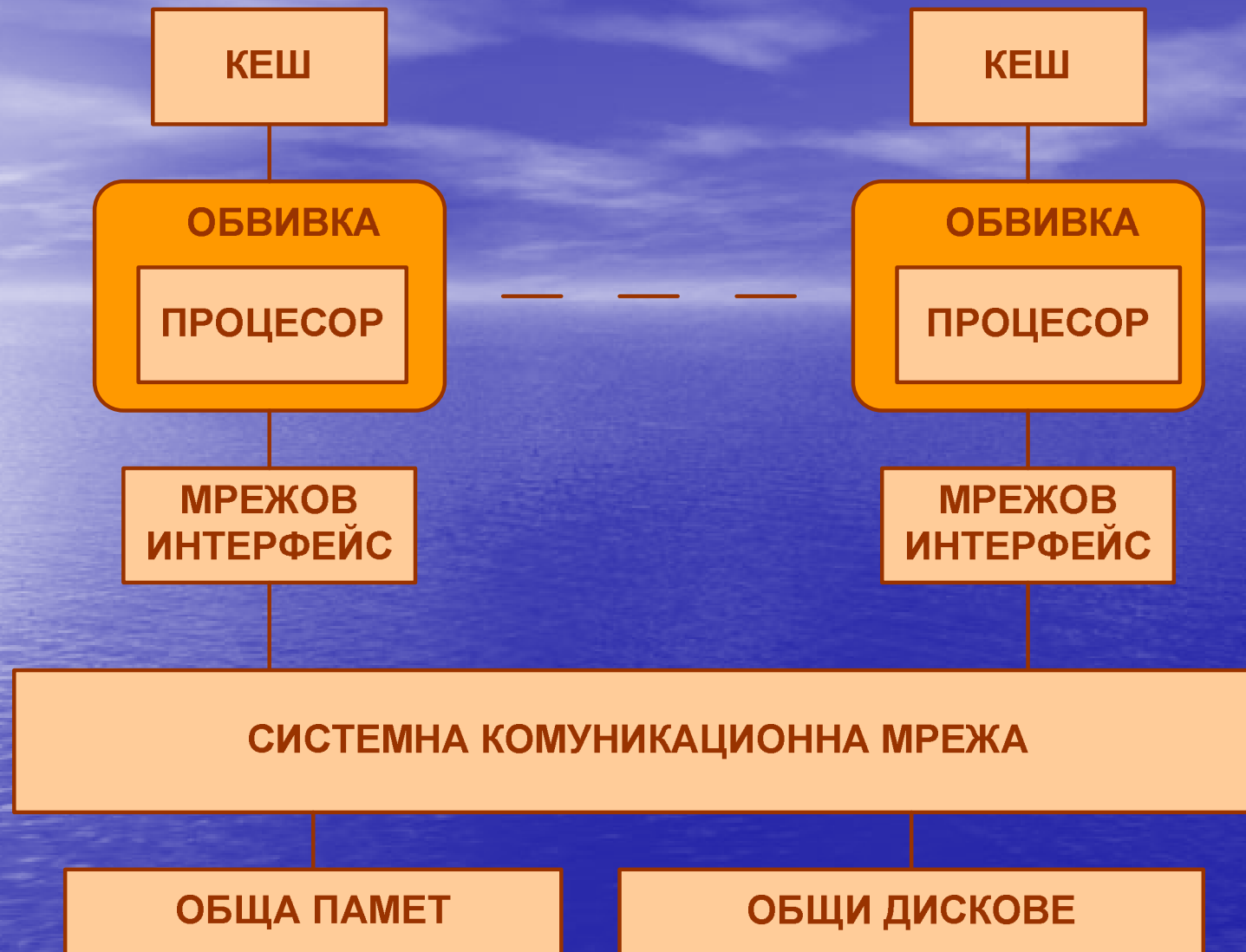


архитектура без общи ресурси



## архитектура с общи дискове





архитектура с обща памет

Цялостната структура на компютърната система се нарича **макроархитектура**, докато структурата на процесора и неговата обвивка формират т.нар. **микроархитектура**.

Повечето съвременни комерсиални паралелни компютри са конструктивно оформени в шкаф, с фабрично произведен хардуер и съответни софтуерни компоненти.

*Актуални са паралелните архитектури, включени в класа MIMD от класификацията на Флин . Този архитектурен клас включва пет физически машинни модела : **паралелни векторни процесори (PVP), симетрични мултипроцесори (SMP), масивно паралелни процесори (MPP), машини с разпределена обща памет (DSM), и клъстери от работни станции (COW).***

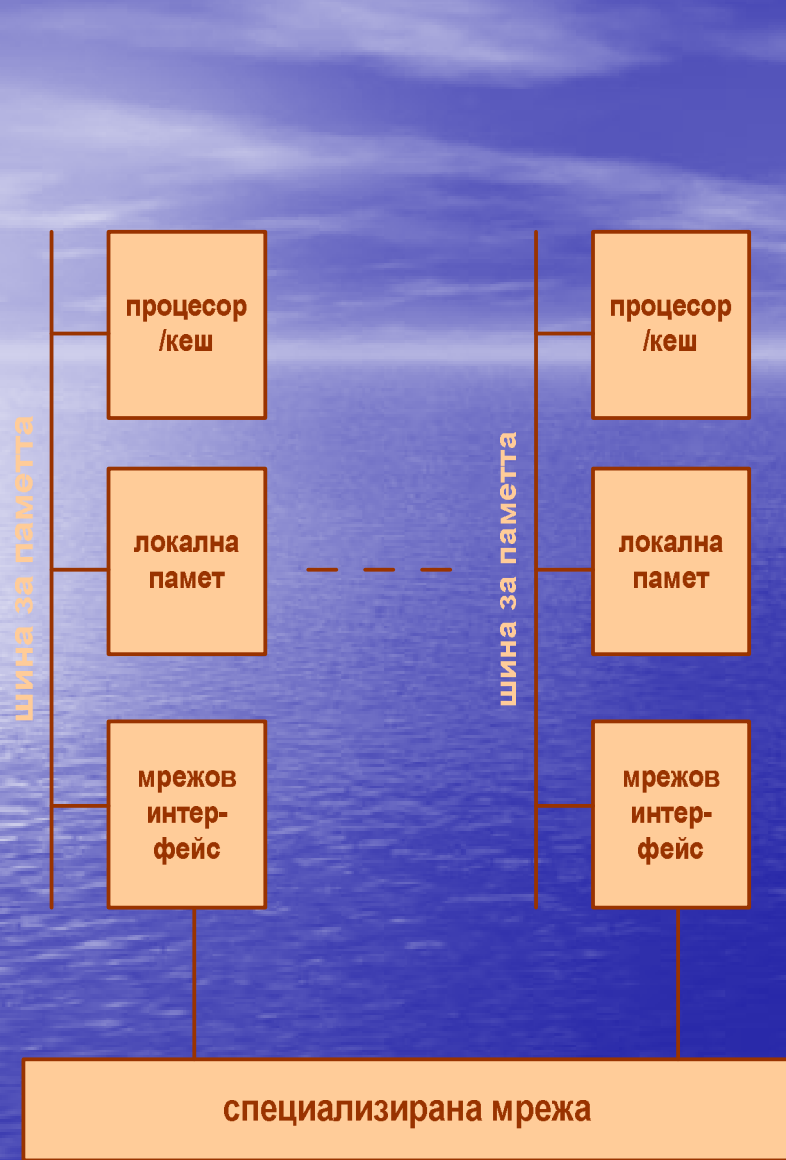




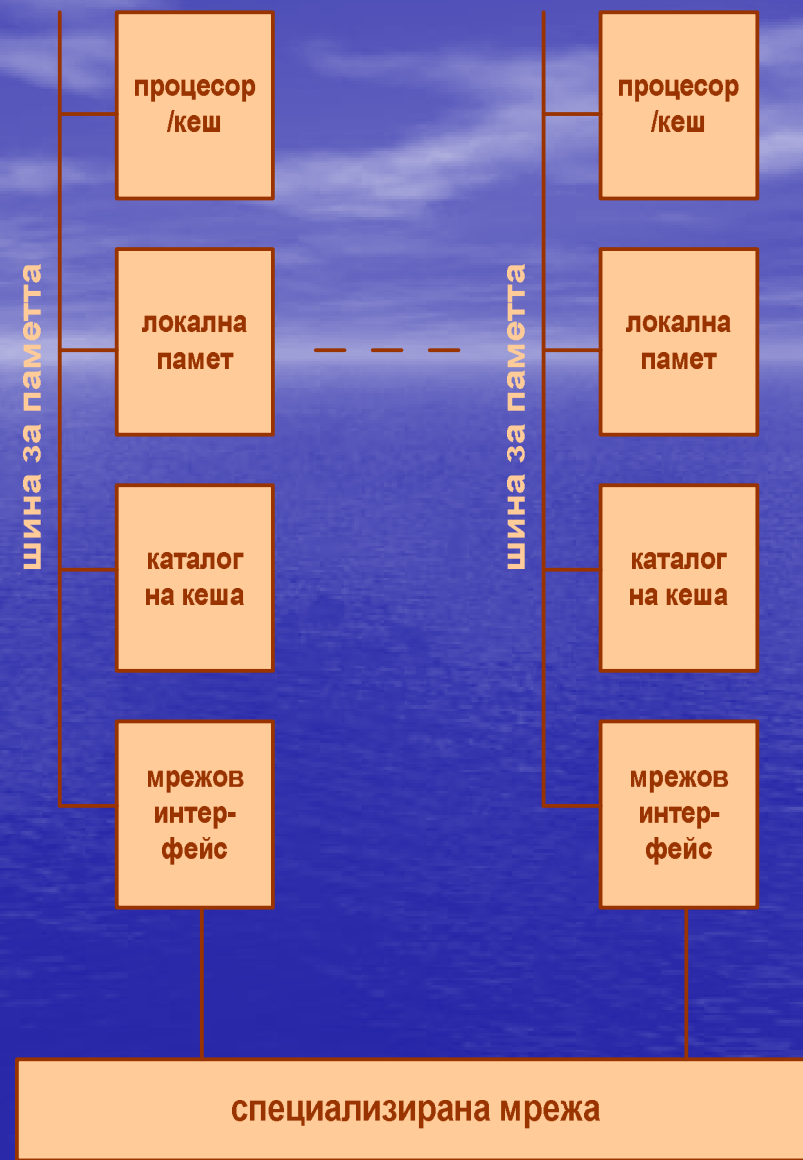
Паралелен векторен процесор



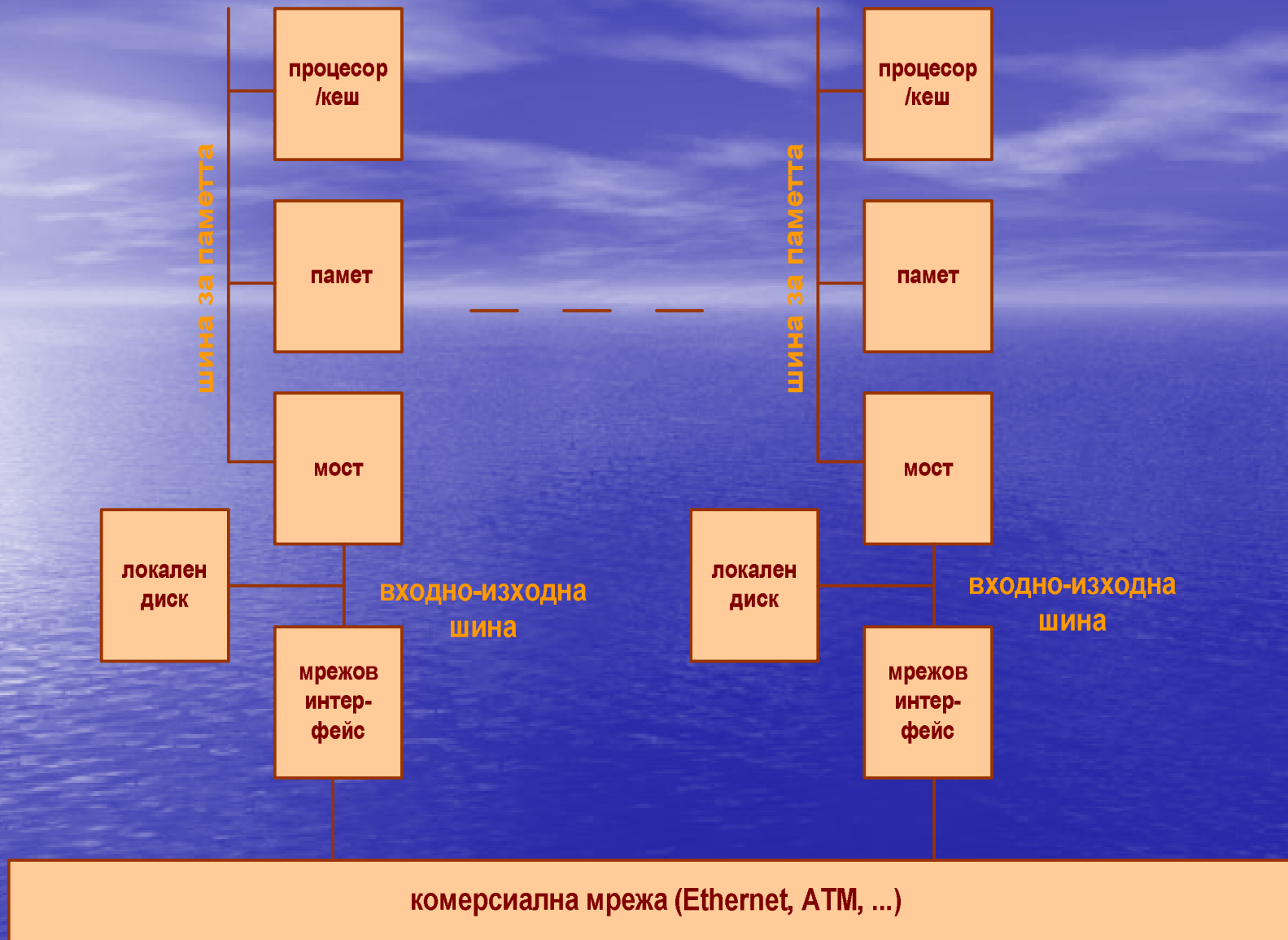
Симетричен мултипроцесор



Масивно паралелен процесор



Машина с разпределена обща памет



# Клъстер от работни станции

**Семантичните атрибути**, характеризиращи физическите модели на паралелните машини са **хомогенност, синхронност, механизъм за взаимодействие на паралелните процеси, адресно пространство, вид на достъпа, модел на паметта**. По отношение на семантичния атрибут **синхронност**, и петте архитектурни класа паралелни компютри представляват асинхронни или слабо синхронни системи. **Механизмът за взаимодействие на паралелните процеси** при паралелните векторни процесори, симетричните мултипроцесори и машините с разпределена обща памет се основават на **обща променливи**, които споделят и **общо адресно пространство**.



*Механизмът за взаимодействие на паралелните процеси при масивно паралелните процесори и клъстерите от работни станции се базира на обмен на съобщения и използването на множество отделни адресни пространства. Видовете достъп са унифициран достъп до паметта UMA (Unified Memory Access) при паралелните векторни процесори и симетричните мултипроцесори, неунифициран достъп до паметта NUMA (Non-Unified Memory Access) при машините с разпределена обща памет и без дистанционен достъп до паметта (NORMA) при масивно паралелните процесори и клъстерите от работни станции.*



*Архитектурата с обща памет* трябва да бъде разграничавана от *програмна среда с обща памет*. Архитектурата с обща памет (мултипроцесорът) може да поддържа както програмен модел с обща памет, така и програмен модел с обмен на съобщения. Програмен модел с обща памет може да бъде имплементиран както в архитектура с обща памет (мултипроцесор), така и в архитектура с разпределена памет (мултикомпютър). В общия случай се приема, че симетричните мултипроцесори имат архитектура с обща памет, докато традиционните масивно паралелни процесори (напр., Intel Paragon), имат архитектура с разпределена памет.





Паралелната компютърна архитектура е с обща памет (*мултипроцесор*) ако всеки процесор има директен достъп до всяка локална или дистанционна памет в цялата система. В противен случай архитектурата няма обща памет (*мултикомпютър*).

Ключовата дума тук е "*директен достъп*", която означава, че при изпълнението на инструкциите *load* и *store* се осъществява достъп до произволна клетка на цялата системна памет на паралелния компютър.

В действителност дистанционният достъп до паметта не е директен, а се осъществява посредством програмен слой като например библиотечни подпрограми, извиквани от потребителя. Това рефлектира в различната *латентност* на достъпите до паметта.



## *Симетрични мултипроцесори*

Симетричните мултипроцесори се използват интензивно в комерсиалните приложения като бази данни и системи с он-лайн транзакции. От особена важност при симетричните мултипроцесори е *ВСИЧКИ процесори да имат равнопоставен достъп до общата памет, В/И устройства и услугите на операционната система*. Симетричността на тези системи дава възможност да се оползотвори висока степен на паралелизъм, което не е възможно при асиметричните (главен-подчинени) мултипроцесори.

*Главното ограничение* се обуславя от централизираната обща памет, както и от общата шина или комутатора кросбар, които след като веднъж са изградени, трудно подлежат на модулно разширение.



## *Масивно паралелни процесори*

- Възлите съдържат специално проектирани микропроцесори;
- Използват памет, физически разпределена между възлите;
  - Съдържат системна мрежа с много висока пропускателна способност и ниска латентност;
  - Имат възможности за модулно разширение до стотици и дори хиляди процесори;
  - Представяват асинхронна MIMD машина.  
*Процесите, обаче, се синхронизират чрез операции за обмен на съобщения, а не чрез операции за синхронизация с общи променливи!*

*Програмата се състои от множество процеси, всеки от които има индивидуално адресно пространство. Процесите взаимодействат чрез обмен на съобщения.*



**Масивно паралелните процесори** трябва да осигуряват универсална система за информационна обработка, която *поддържа всички видове приложения (технически и комерсиални), различни алгоритмични парадигми и различни работни режими.* В архитектурен аспект те поддържат асинхронен MIMD режим и стандартния програмен модел с обмен на съобщения (PVM – паралелна виртуална машина и MPI – интерфейс с обмен на съобщения). Тясно свързаните масивно паралелни процесори функционират под управлението на разпределена операционна система. В общия случай, микроядро на операционната система се изпълнява във всеки възел. На приложното програмно ниво се поддържат стандартни езици и библиотеки като C, Fortran, HPF, PVM и MPI.



## Клъстери от компютри

- **Всеки възел представлява цялостна работна станция без някои от периферните устройства** (дисплей, клавиатура, мишка, и др.). Понякога *такъв възел се нарича работна станция "без глава"*. Възелът също така може да бъде *симетричен мултипроцесор или РС*.
- *Възлите се свързват чрез Ethernet, FDDI, Fiber-Channel или ATM комутатор.*
- *В рамките на възела мрежовият интерфейс е слабо свързан към входно/изходната шина.* Това контрастира с тясно свързания мрежов интерфейс на масивно паралелните процесори, който е свързан към шината на паметта на обработващия възел.
- *Винаги има локален диск,* който не е задължителен за възела на масивно паралелните процесори.
- *Всеки възел се управлява от цялостна операционна система, докато при някои масивно паралелни процесори се използва само ядро.*



*Клъстерът представлява колекция от цели компютри (възли), които физически са свързани от високо производителна мрежа или локална мрежа.*

В общия случай, всеки възел представлява сървър, който е симетричен мултипроцесор, работна станция или персонален компютър.

*Всички възли на клъстера трябва да работят колективно като един интегриран изчислителен ресурс. Допълнително отделните възли трябва да изпълняват конвенционалната си роля индивидуално за интерактивните потребители.*



## *Архитектурно сравнение*

### *четири припокриващи се архитектурни концепции*

*Възелът на клъстера е по-сложен от възела на масивно паралелния процесор, тъй като има диск и пълна операционна система. Сървърът с архитектура на симетричен мултипроцесор е по-сложен от възела на клъстера, защото има по-богата периферия като терминали, принтери, RAID дискове. Някои от тези периферни устройства могат да липсват при възлите на клъстера. **Тенденцията** е в бъдеще *границите на клъстера да продължават да се разширяват до припокриване във все по-голяма степен с масивно паралелните процесори и симетричните мултипроцесори.**



Клъстерът или масивно паралелният процесор могат да бъдат използвани като *единичен ресурс т.е. като една високопроизводителна работна станция* докато разпределената система продуцира множество образи на системата поради автономността на индивидуалните компютри в мрежата. Съвременните разпределени системи и *масивно паралелните процесори съдържат до няколко хиляди възела. Повечето клъстери съдържат десетки възела, като много малко клъстери имат повече от стотина възела.*





*Възлите на масивно паралелните процесори често изпълняват микроядро докато възлите в другите архитектури използват пълни операционни системи!!!*

Операционните системи в разпределената система обикновено са хетерогенни, но за другите класове са предпочитани хомогенните. *При масивно*

*паралелните процесори и клъстерите комуникацията между възлите се осъществява чрез обмен на съобщения. При разпределените системи често за тази цел се използват общи файлове в сървъра.*

***Процесорите в симетричните мултипроцесори комуникират посредством общата памет. Само симетричните мултипроцесори поддържат едно адресно пространство. Масивно***

***паралелните процесори могат да имат само едно адресно пространство само ако се поддържа хардуерно разпределена обща памет.***



***Симетричните мултипроцесори имат една единствена опашка от процеси за изпълнение за всички процесори. При клъстерите и възлите на масивно паралелните процесори съществуват множество такива опашки, но те се координират с цел баланс на работния товар.*** Множеството опашки при разпределените системи са почти независими. Разпределените системи трябва да използват стандартни комуникационни мрежи и протоколи за съгласуване на хетерогенните платформи на възлите. Разпределената система често обслужва множество организации, което изисква мерки за защита. Това изискване не е в сила за масивно паралелните процесори и клъстерите.



Индивидуалните възли на клъстера имат традиционни платформи, така че потребителите могат да създават и изпълняват приложенията си в позната среда. *Тези платформи осигуряват мощните средства за развитие на програмите на високопроизводителните работни станции, което позволява съществуващите последователни приложни програми да се изпълняват без промяна.* По този начин **клъстерът може да се разглежда като огромна работна станция, осигуряваща значително по-висока пропускателна способност и намалено време за отговор за множество последователни потребителски програми.** За паралелните приложения, програмирането на клъстера е много по-сложно от това на масивно паралелен процесор с обмен на съобщения.



*При симетричните мултипроцесори общата памет (и шината на паметта, съответно) представляват тясно място. В случаите, когато се изпълняват няколко програми, те се конкурират за ползване на паметта, дори когато тези програми са напълно независими. Когато същото множество програми се изпълняват в клъстери или масивно паралелни процесори, всяка програма се изпълнява в един възел, като се използва само локалната памет. Архитектурата на паралелния компютър трябва да бъде отворена с осигурен стандартен интерфейс към останалата част на системата. Възелът не трябва да зависи от мрежата, а мрежовият интерфейс трябва да бъде независим от мрежовата топология.*



Основната разлика между масивно паралелните процесори и клъстерите от работни станции се състои в комуникационната мрежа между възлите. При клъстерите от работни станции възлите най-често се свързват чрез стандартна локална мрежа. При масивно паралелните процесори възлите се свързват чрез специализирана комуникационна мрежа с висока пропускателна способност и ниска латентност като с цел осигуряване на висока производителност се използва специализиран софтуер за комуникацията. Следователно, масивно паралелните процесори превъзхождат по високоскоростната си комуникация клъстерите от работни станции.



## Клъстери

Това е **най-популярният архитектурен стил при съвременните високопроизводителни системи. От списъка на 500-те най-мощни суперкомпютри в световен мащаб, 377 имат клъстерна архитектура ([www.top500.org](http://www.top500.org))**. Понастоящем повече от **100 000 компютърни клъстера** се използват в световен мащаб като техният брой непрекъснато нараства с високи темпове. Възлите в клъстера са преобладаващо персонални компютри, работни станции и SMP сървъри. Високопроизводителните клъстери съдържат повече от 100 възела. Повечето клъстери използват *Fast* или *Gigabit Ethernet*, *FDDI* рингове, *ATM* или *Myrinet* комутатори за осигуряване на високоскоростна комуникация между възлите като допълнение към локалната мрежа, използвана в клъстера. В общия случай високопроизводителните клъстери съдържат **специализирана комуникационна подсистема, включваща високоскоростна мрежа, мрежов интерфейс и комуникационен софтуер**. Примери на такива системи са DEC TruCluster и IBM SP2.



*Особено популярен подход е клъстеризацията на хомогенни SMP сървъри в рамките на интегриран суперсървър. Такива суперсървъри са Sun Ultra Enterprise 1000 и SGI POWER CHALLENGE. Характерно са суперсървърите е съчетаването на техниките обмен на съобщения и обща памет, както и използването на йерархия от програмни модели: обща памет с паралелен Fortran, HPF, C и обмен на съобщения с PVM или MPI. Едно от важните предимства на суперсървърите е, че те осигуряват високоскоростна поддръжка на графичните приложения, което е от особено значение за интерактивните възможности на суперсистемите.*



**Суперкомпютърът IBM ASCI White** (2000 г.) представлява *клъстер*, свързващ 512 комерсиални компютри RS/6000 SP, съдържа общо 8192 процесора, **отделният възел съдържа 16 процесора**, а общият капацитет на паметта е **6 TB**, общ капацитет на дисковете – **160 TB**.

Максималната производителност е 7 226 GFlops или **7,226 TFlops**.

*Производителността му се равнява на 50 000 десктоп компютъра.*



# КРАЙ

