

**Veri tabanlı projeleri artık hızlı,
kolay ve ekonomik: Netezza**

Feriyeye - İstanbul
7 Haziran 2012, Perşembe

Netezza Teknolojisine Kısa Bir Yolculuk

Ayhan Önder
Netezza Teknik Uzmanı

Saatlerce süren sorgular

düzenli optimizasyon

“

*Mevcut Veri Ambarlarının **yaklaşık %70**'i farklı tiplerde performans problemleri yaşamakta....*

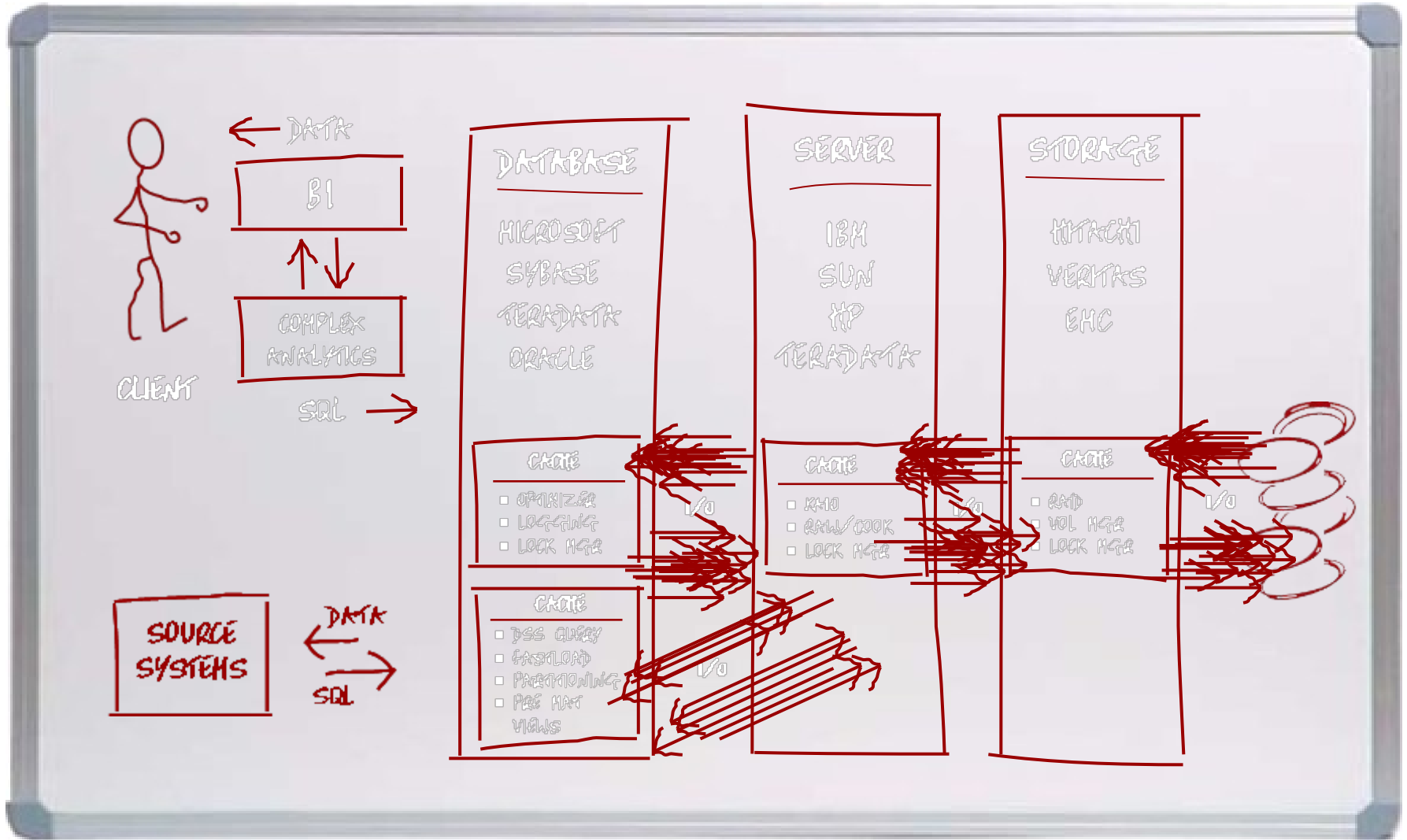
”

- Gartner Magic Quadrant

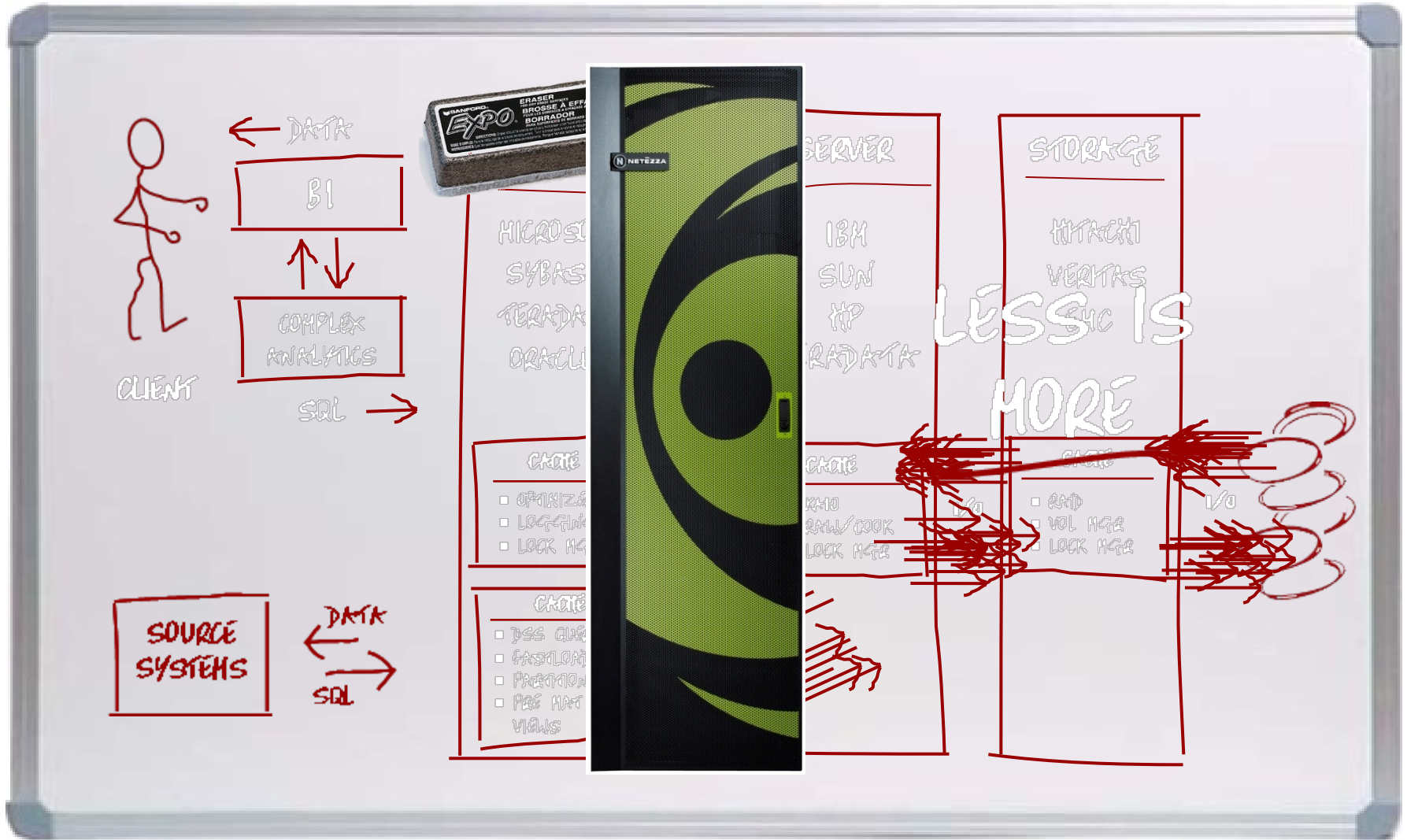
yetkin personel gereksinimi

aylarca süren geliştirmeler

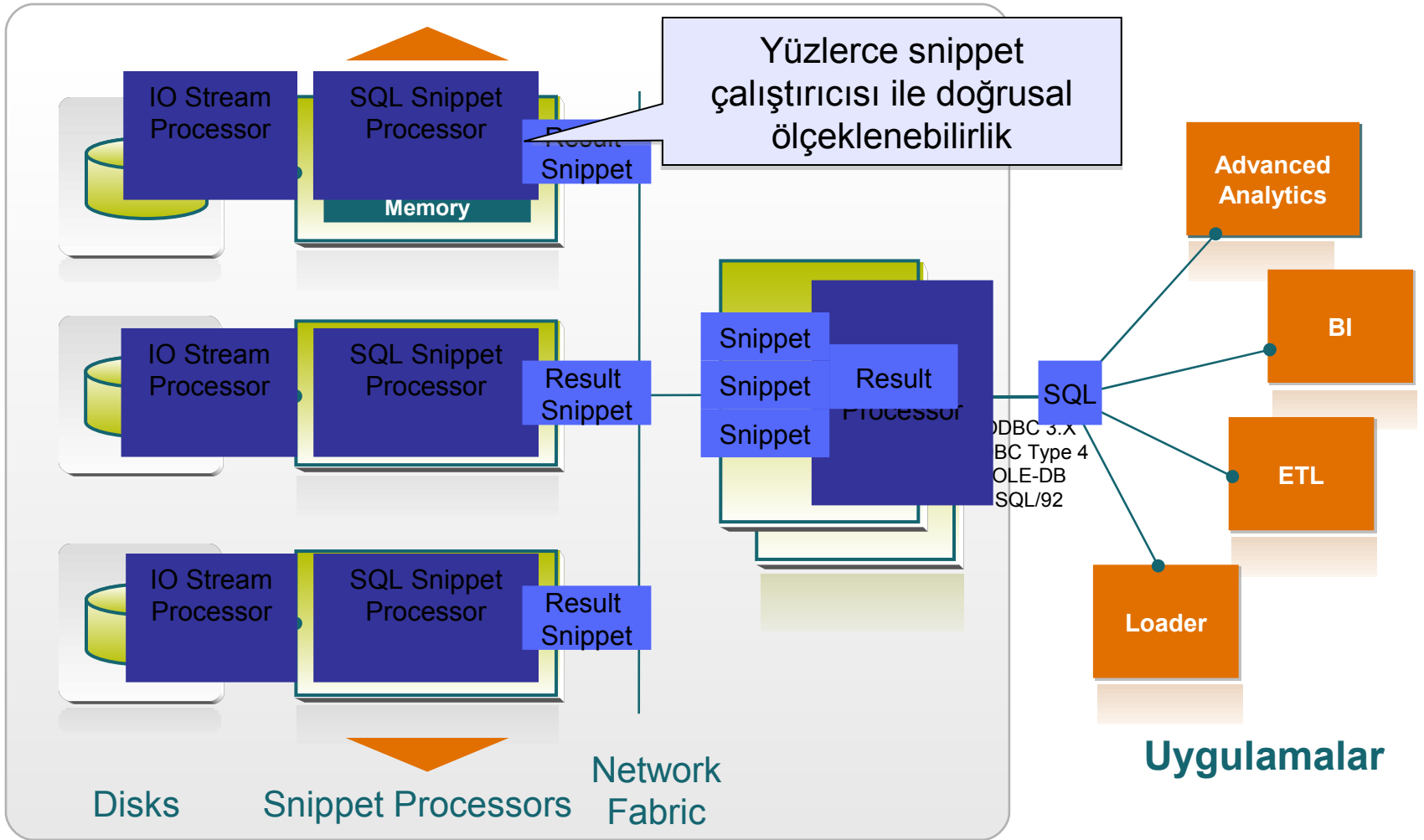
Geleneksel sistemlerle Veri Ambarı çözümü



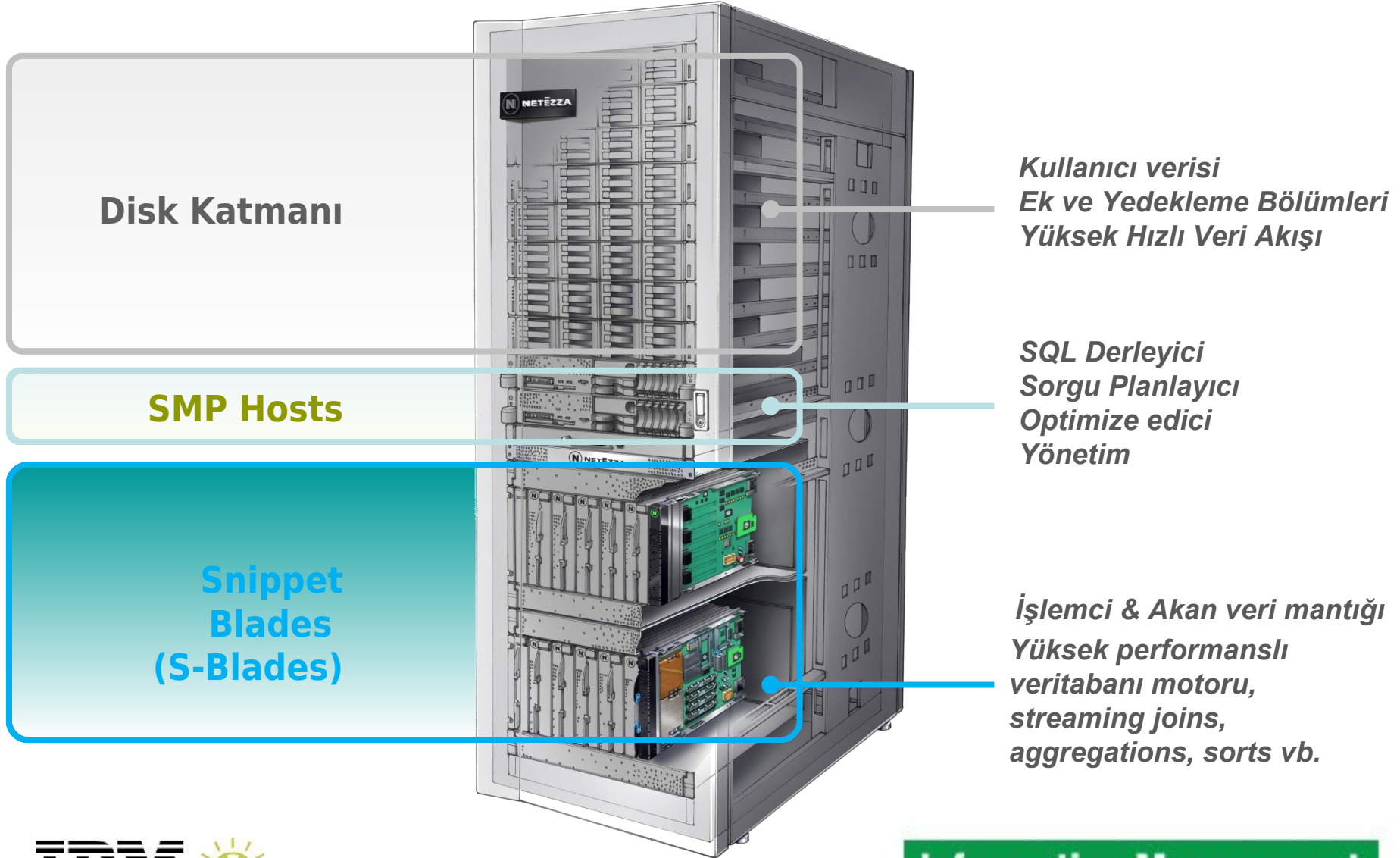
Netezza Yaklaşımı



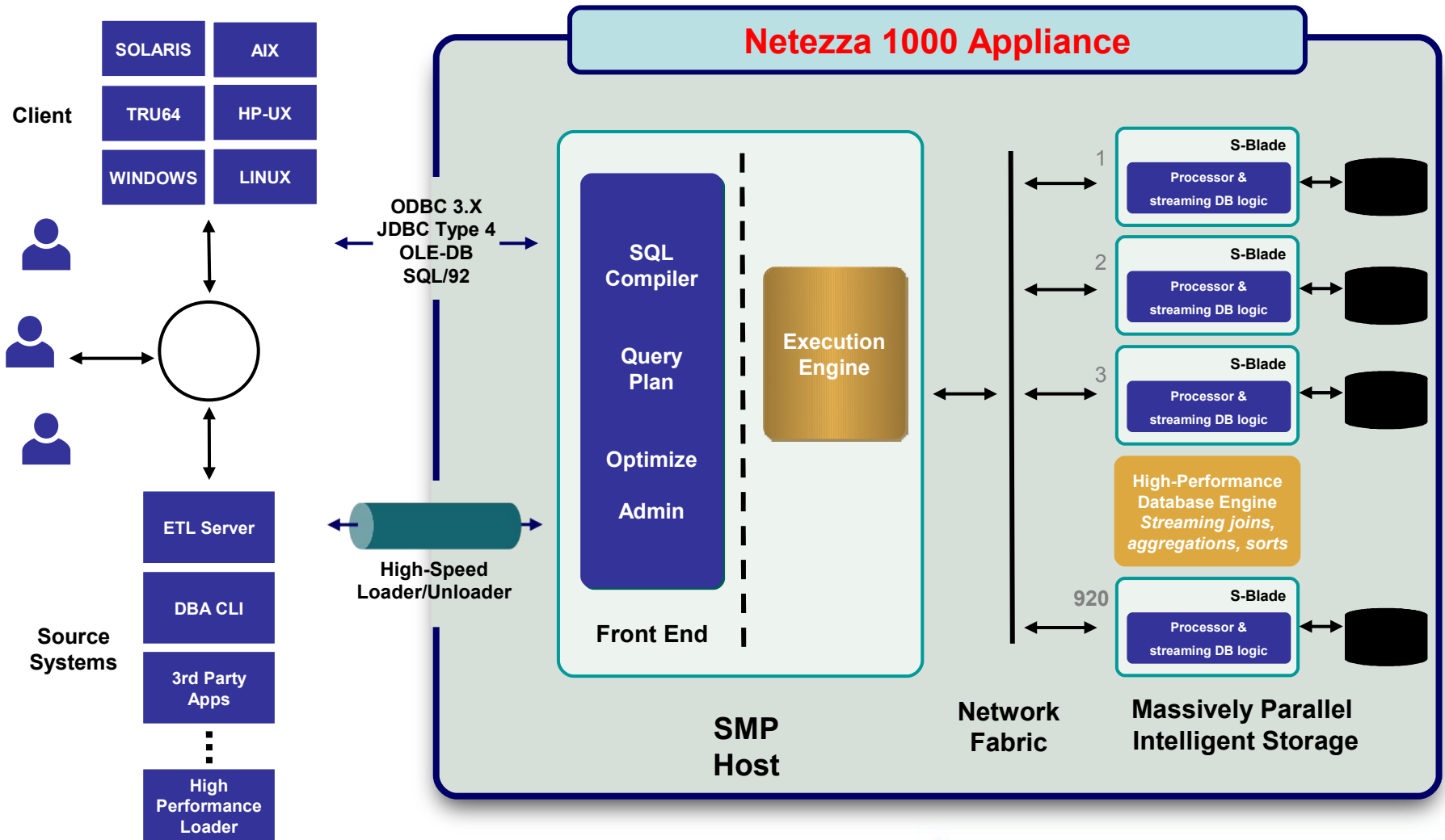
Netezza AMPP mimarisi



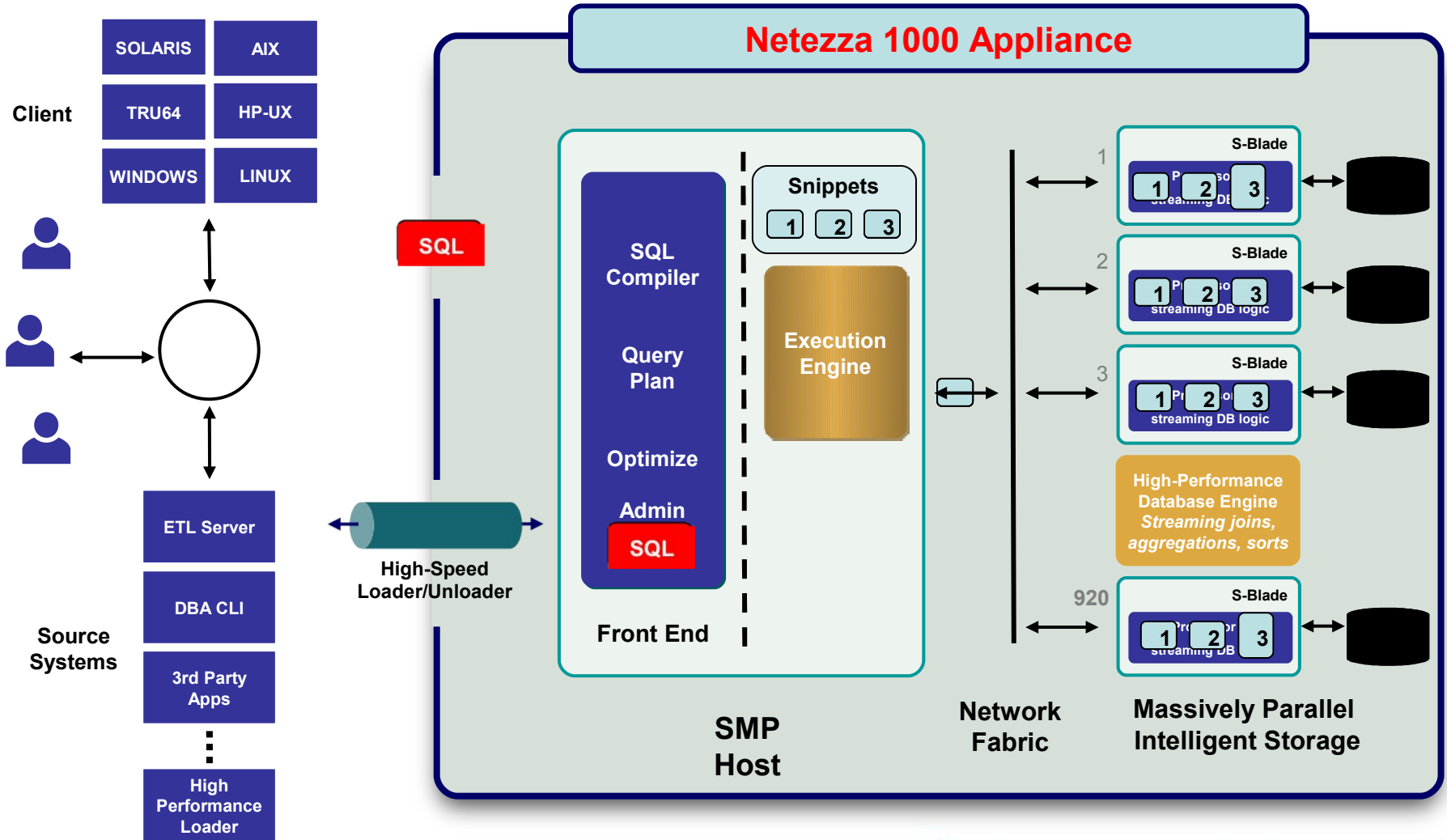
IBM Netezza 1000 Appliance



Netezza AMPP mimarisi



Netezza AMPP mimarisi



Derlenmiş kodlar, bütünüyle Paralel bir mimaride çalıştırılır

```
select c_name, sum(o_orderkey price) price from customer, orders where  
o_orderkey in (select o_orderkey from lineitem2 where
```

```
o_orderkey /***** Code *****/
```

```
l_shipmode
```

```
test_t
```

```
void GenPlan1(CPlan *plan, char *bufStarts, char *bufEnds, bool lastCall) {
```

```
//
```

```
// Setup for next loop (nodes 00..07)
```

```
//
```

```
// node 00 (TScanNode)
```

```
TScanNode *node0 = (TScanNode*)plan->m_nodeArray[0];
```

```
// For ScanNode:
```

```
    TScan0 *Scan0 = BADPTR(TScan0*);
```

```
    CTable *tScan0 = plan->m_nodeArray[0]->m_result;
```

```
char *nullsScan0P = BADPTR(char *);
```

```
// node 01 (TRestrictNode)
```

```
TRestrictNode
```

```
// node 02 (TProjectNode)
```

```
TProjectNode *node2 = (TProjectNode*)plan->m_nodeArray[2];
```

```
// node 03 (TSaveTempNode)
```

```
TSaveTempNode *node3 = (TSaveTempNode*)plan->m_nodeArray[3];
```

```
// For SaveTemp Node:
```

```
c_name          | price
```

```
-----+-----  
Customer#000000796 | 318356.97
```

```
Customer#000001052 | 293680.56
```

```
Customer#000001949 | 215280.98
```

```
Customer#000002093 | 282531.93
```

```
Customer#000005656 | 335297.31
```

```
Customer#000005861 | 233691.03
```

```
Customer#000006002 | 260000.92
```

```
Customer#000006343 | 590000.82
```

```
Customer#000006532 | 440000.91
```

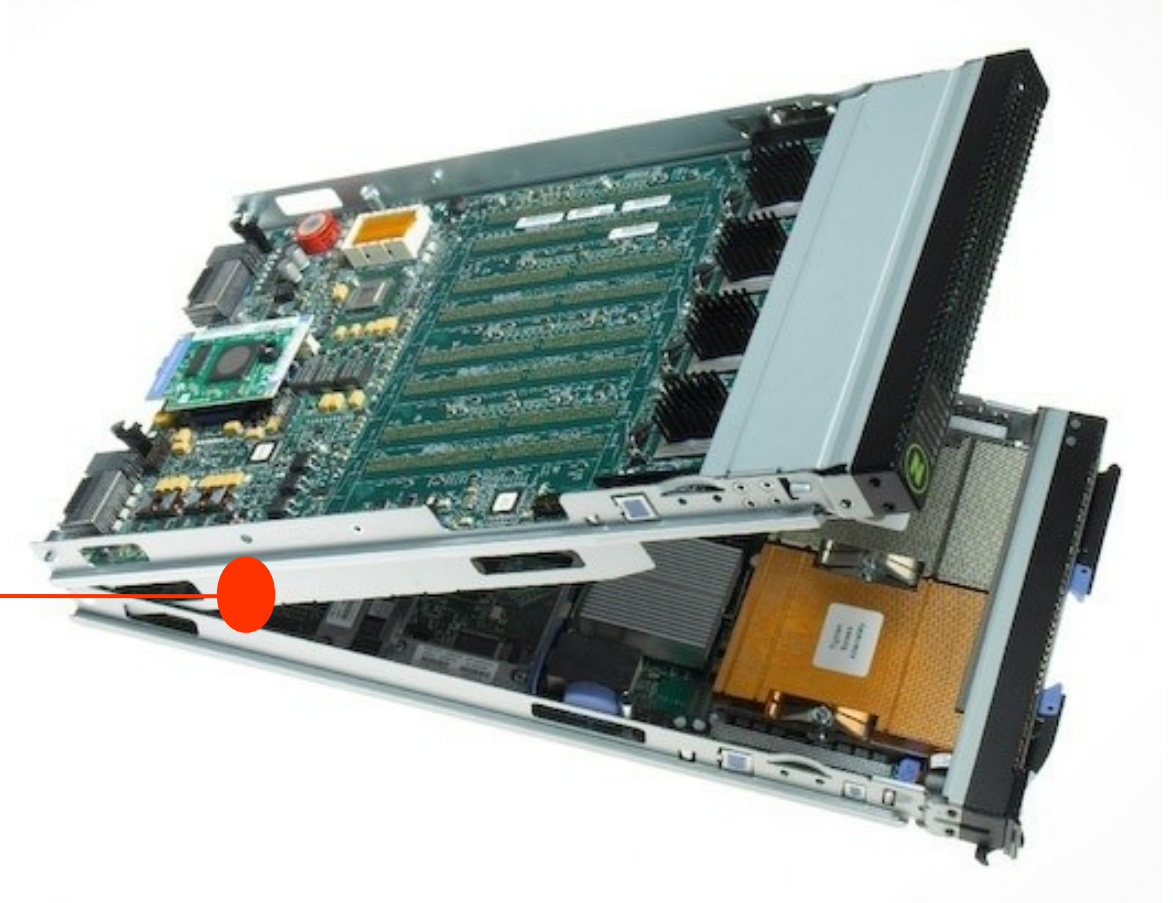
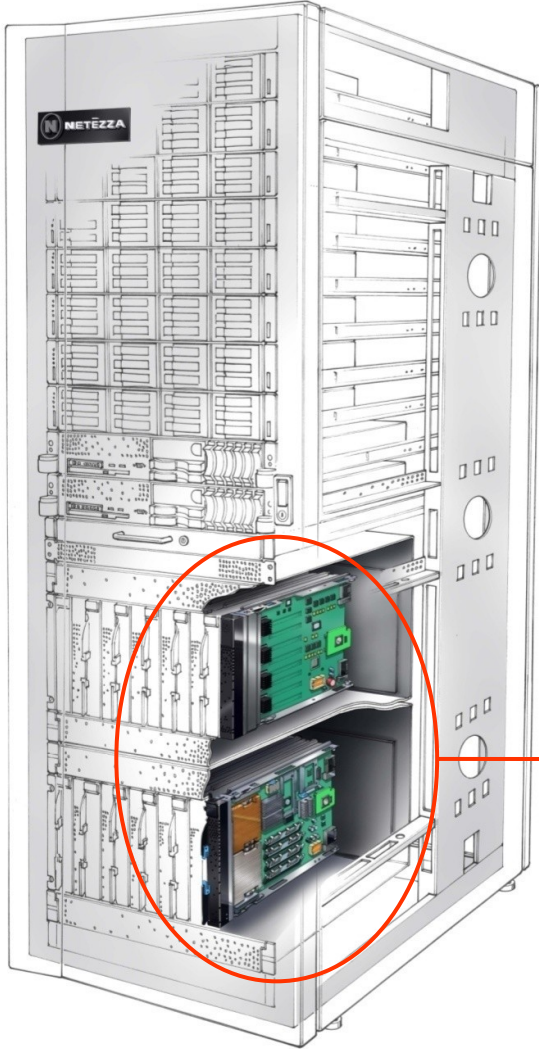
```
...
```

```
real      0m0.552s
```

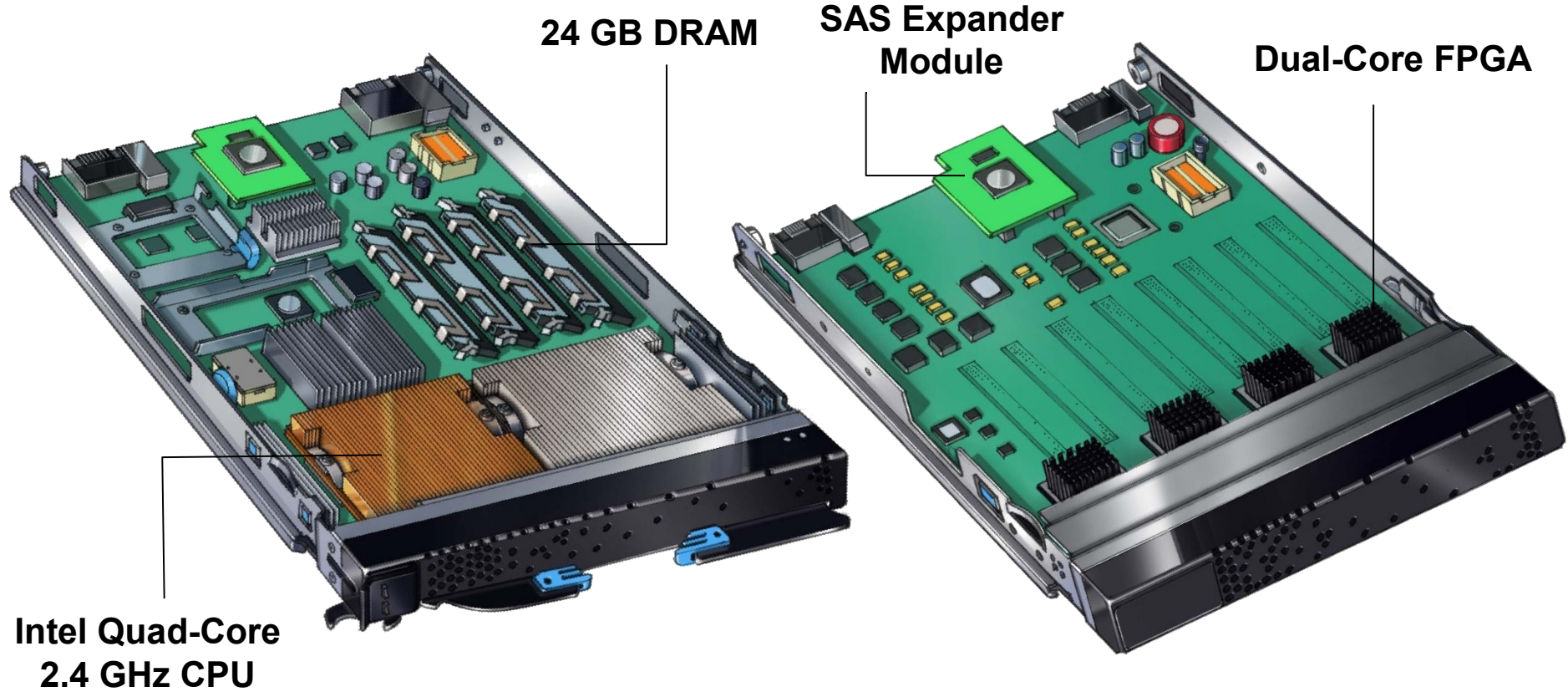
```
user      0m0.010s
```

```
sys       0m0.000s
```

Ölçeklenebilir Mimari : IBM Netezza S-Blade



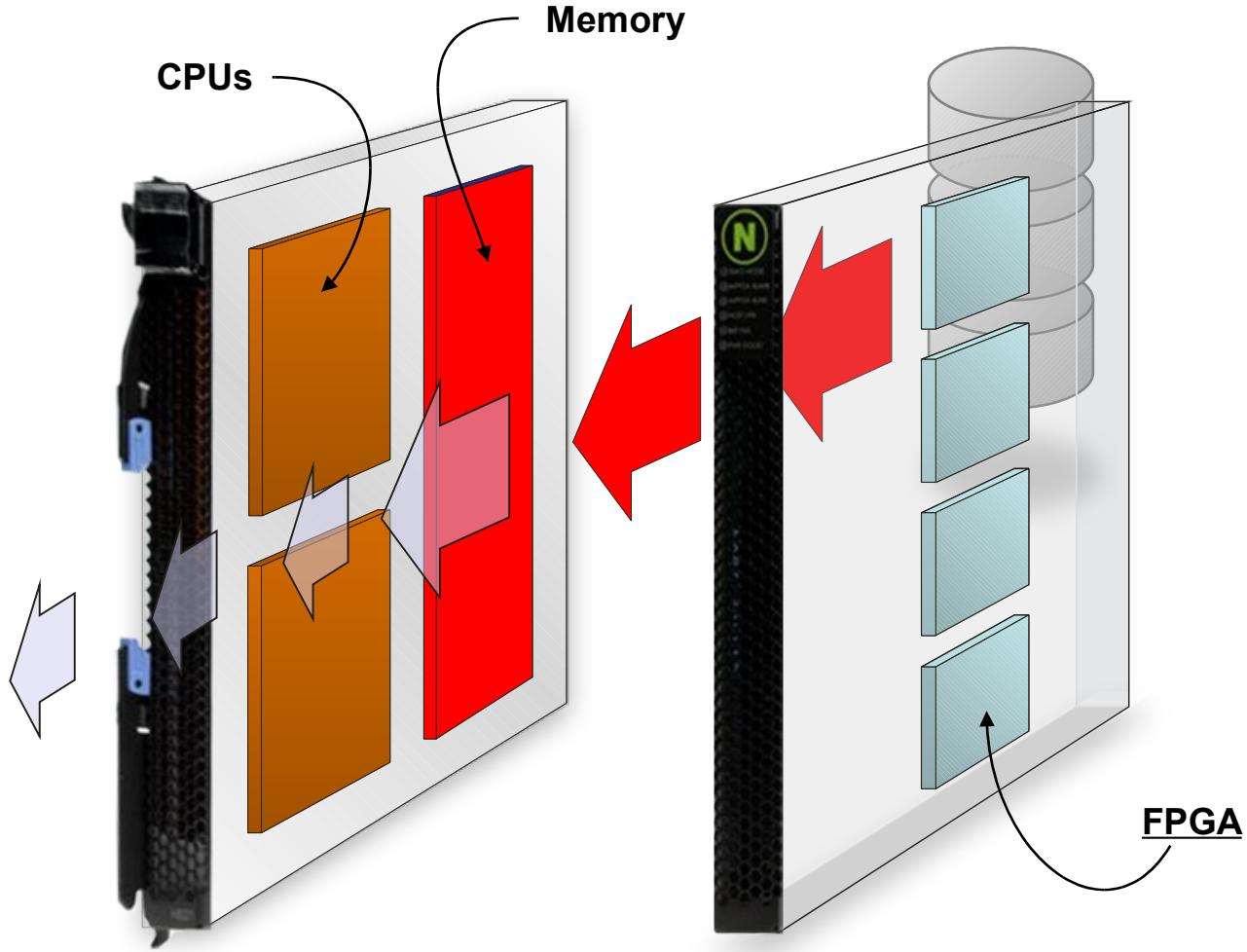
S-Blade Bileşenleri



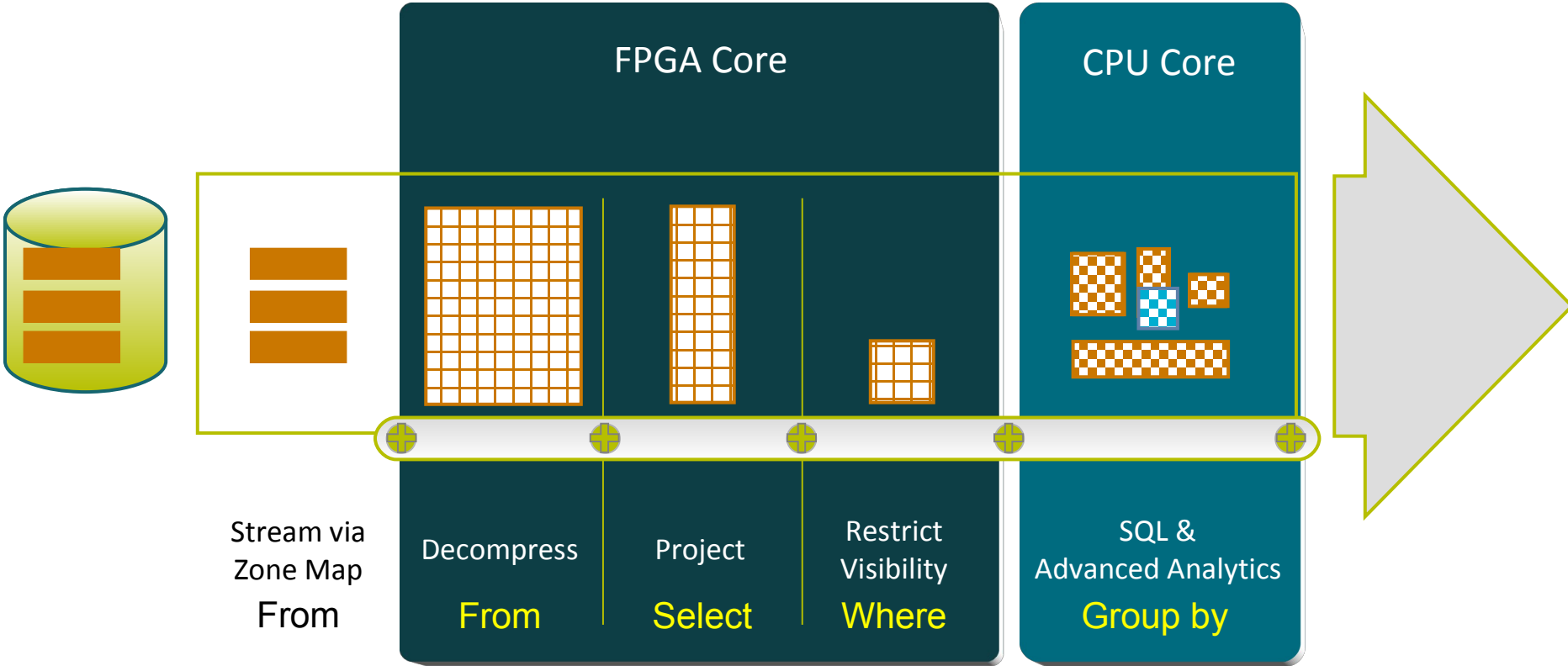
**IBM BladeCenter
Sunucusu**

**Netezza VT
Hızlandırıcısı**

Netezza Veritabanı Hızlandırıcısı, İş Paylaşımı



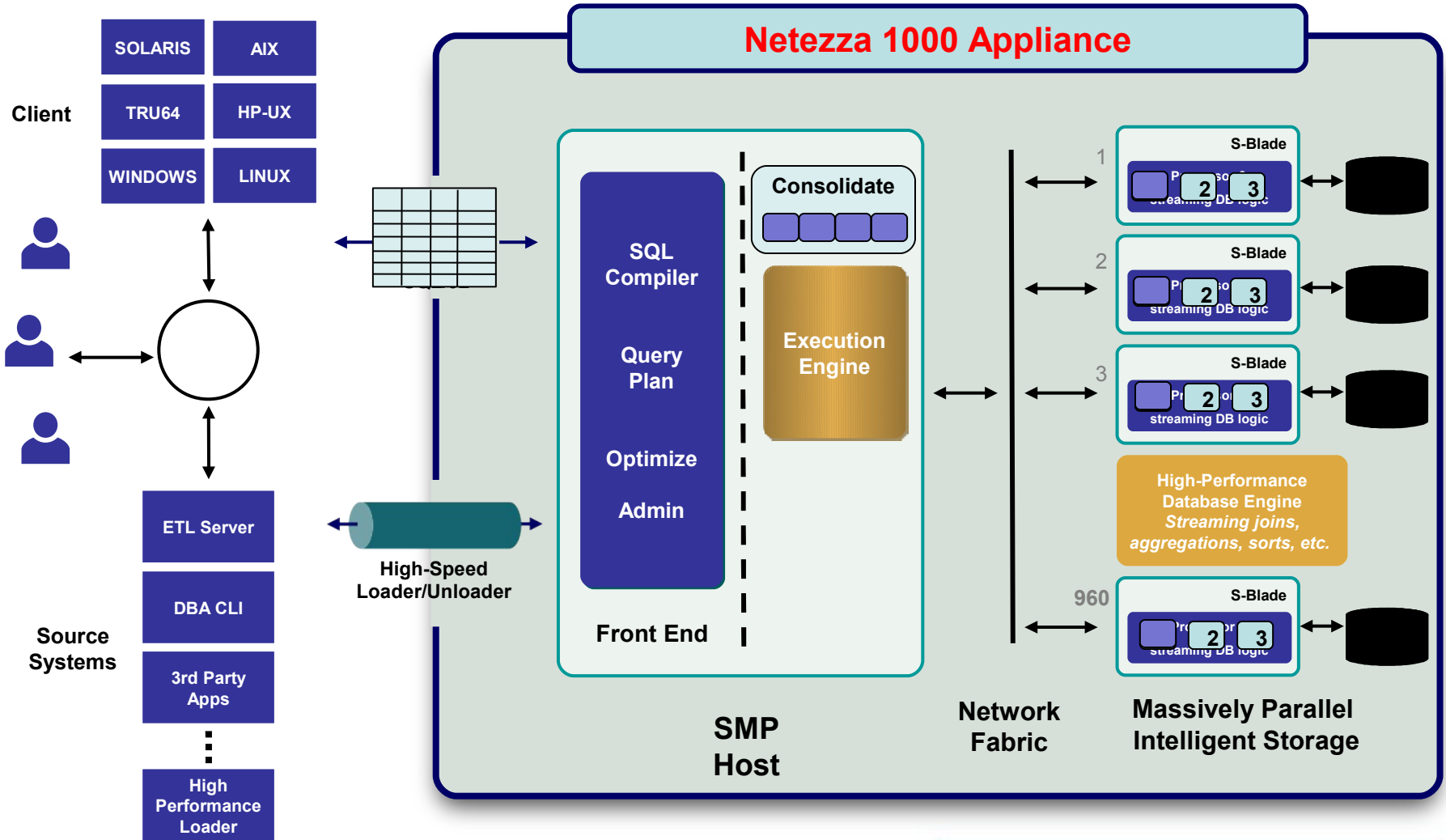
S-Blade Akan Veri İşlemesi



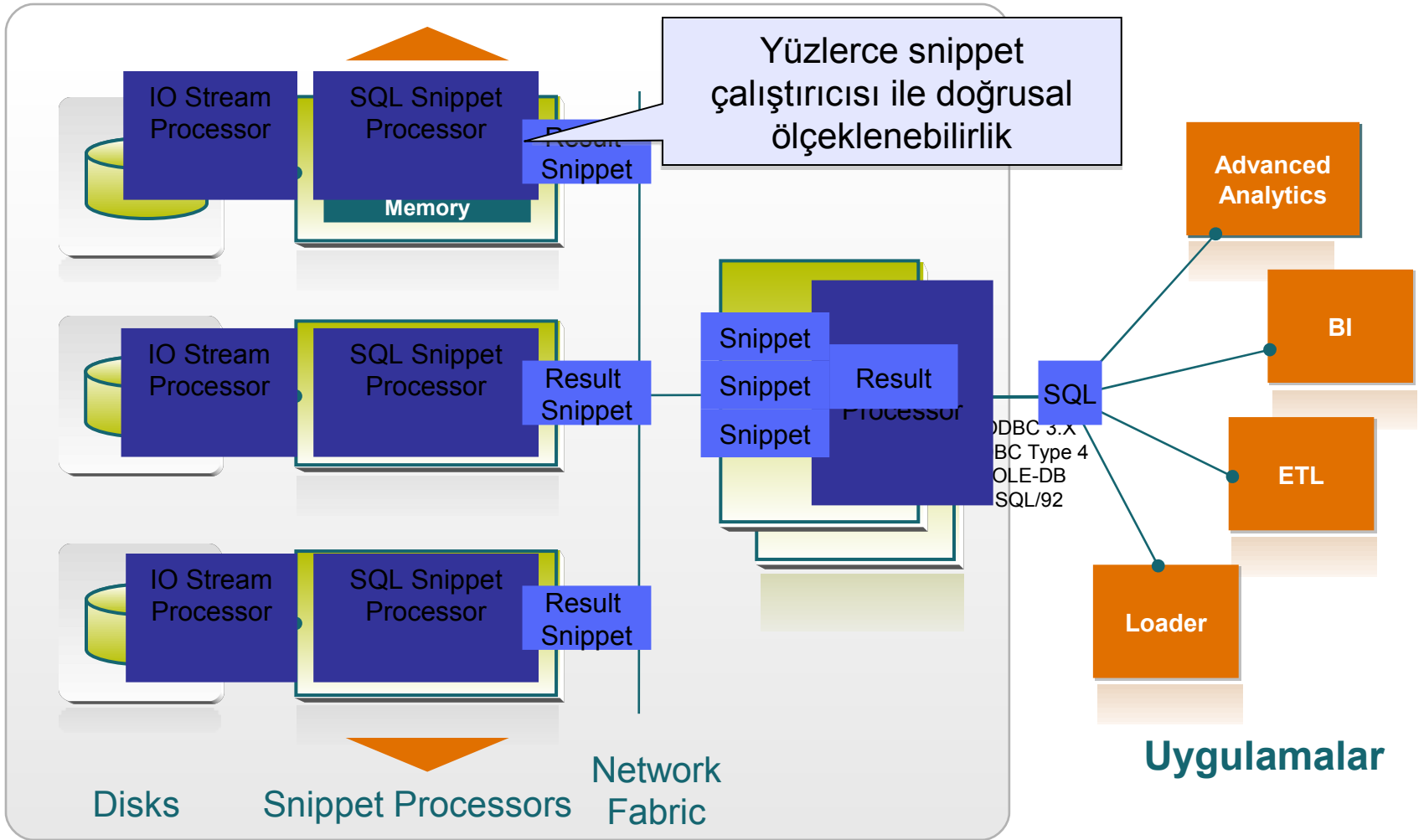
```

Select İlçe, Yas, Cinsiyet, count(*) From Milya da la r a s k a y i t k e r e T a l o W h e r e D o g u m T a r i h i < '01/01/1960'
And ilim ((Istanbul, 'Ankara')) Group by İlçe, Yas, Cinsiyet Order by İlçe, Yas, Cinsiyet
    
```

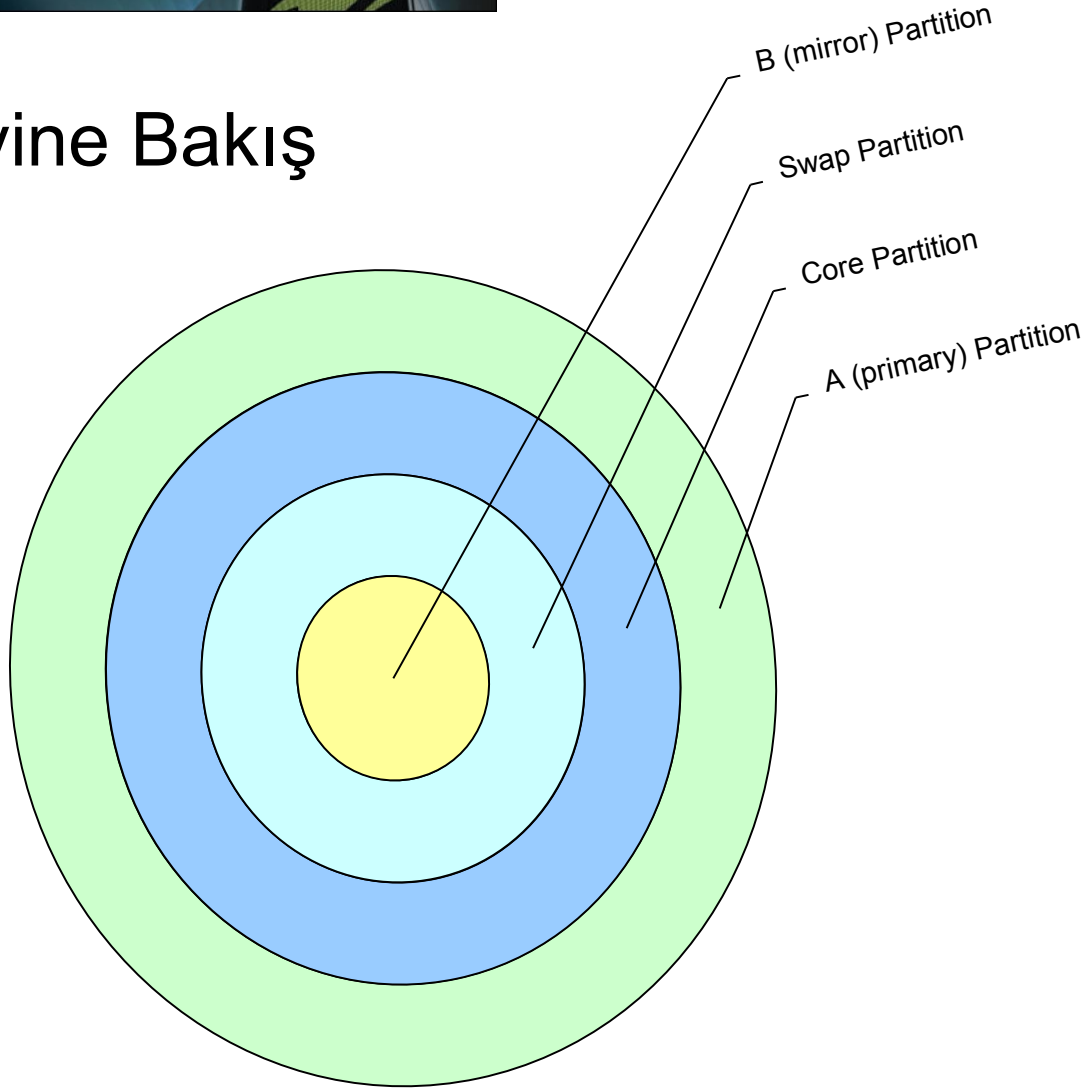
Netezza AMPP mimarisi



Netezza AMPP mimarisi

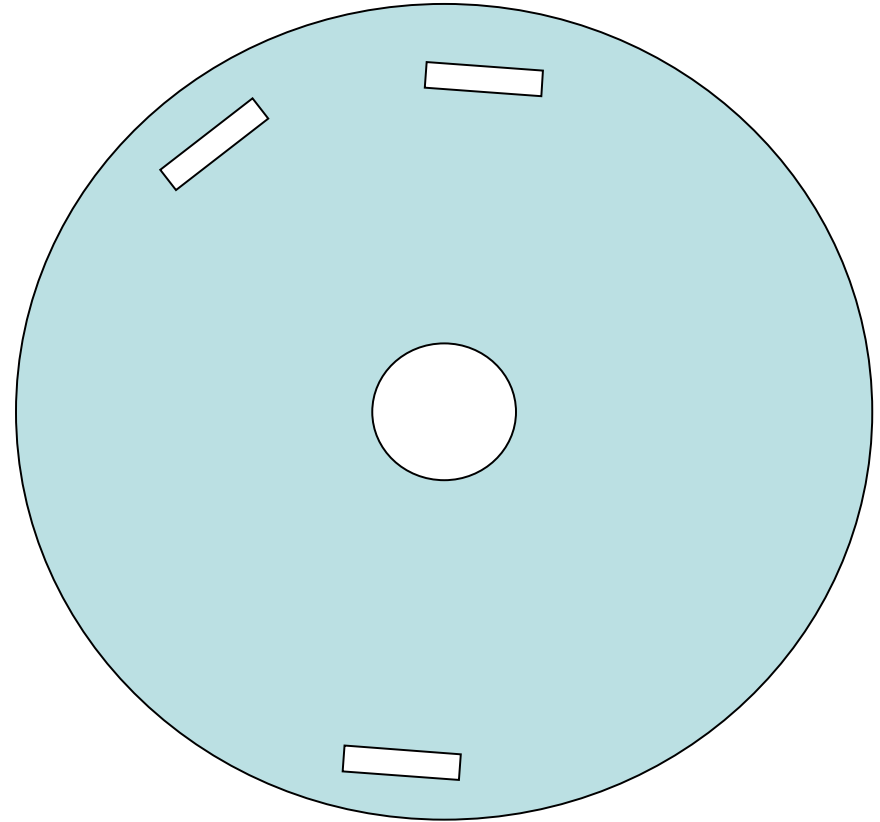


Disk Yüzeyine Bakış



Zone Maps: Gereksiz veri trafiğini önlemek

- Tablodaki alanlara ait min ve maksimum değerler diskteki her extend için hesaplanmaktadır
- Veri yükleme, güncelleme ve silme işlemlerinde zone maps otomatik olarak güncellenir
- Yükleme / silme / güncelleme performansından ödün verilmez



Zone Maps: Gereksiz veri trafiğini önlemek

- Aşağıdaki örnekte sadece 2. disk bölümü (extent) verileri okunuyor olacaktır.

```
SELECT * FROM TradeTable WHERE date=02
```

TradeTable

	Date	Cust_ID	Symbol	----
Extent 1	01	205	NZ	---
	01	100	IBM	---
	01	400	NZ	---
Extent 2	01	100	NCR	---
	02	700	SUN	---
	02	100	LTXX	---
	02	300	NZ	---
	03	210	IBM	---
	03	850	ANX	---
Extent 3	03	205	BP	---
	03	700	NZ	---

Zone Map

Date		Cust_ID	
Min	Max	Min	Max
01	01	100	400
01	03	100	700
03	03	205	850

Verilerin disk yüzeyinde organize edilmesi

- CBT benzer verileri disk yüzeyinde birbirine yakın bölümlere yerleştirirerek optimum sıralı okumayı sağlar
- Mevcut tablo üzerinde gerçekleşir ve ekstra bir yer kaybı söz konusu olmaz

```
select MUSTERI,  
       TUTAR  
from   SATIS  
where  YIL = 2009  
or     YIL = 2010  
and    URUN = 'LOTUS'
```

Tarih Bazlı Yüklenen bir

2008	Lotus
2008	IM
2008	Rational
2008	Tivoli
2009	IM
2009	Rational
2009	Tivoli
2009	Lotus
2010	IM
2010	Tivoli
2010	Rational
2010	Lotus
2011	Tivoli
2011	Rational
2011	Lotus
2011	IM

Tarih

Ürün

Netezza Clustered Base Table

Tarih	Tivoli 2011	Lotus 2011	IM 2011	Rational 2011
	Tivoli 2010	Lotus 2010	IM 2010	Rational 2010
	Tivoli 2009	Lotus 2009	IM 2009	Rational 2009
	Tivoli 2008	Lotus 2008	IM 2008	Rational 2008
	Tivoli 2008	Lotus 2008	IM 2008	Rational 2008

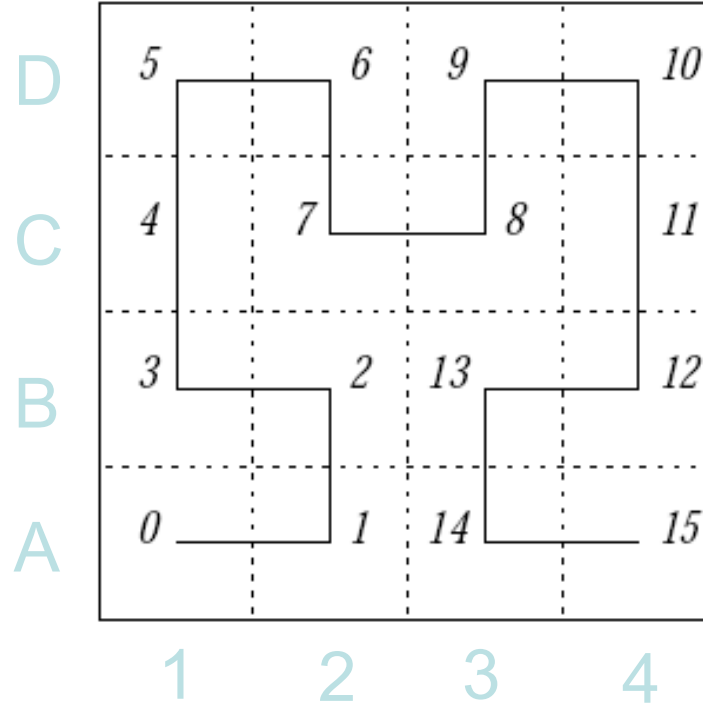
Ürün

Clustered Base Tables: Örnek

ORDER BY
C1, C2

A	1
A	2
A	3
A	4
B	1
B	2
B	3
B	4
C	1
C	2
C	3
C	4
D	1
D	2
D	3
D	4

Çok boyutlu verileri optimum doğrusal sıraya sokar



ORGANIZE ON
C1, C2

0	A	1
1	A	2
2	B	2
3	B	1
4	C	1
5	D	1
6	D	2
7	C	2
8	C	3
9	D	3
10	D	4
11	C	4
12	B	4
13	B	3
14	A	3
15	A	4

Arka planda Hilbert Curve Algoritması kullanılmaktadır

Veri hayat döngüsü için farklı çözümler



IBM Netezza 100	IBM Netezza 1000	High Capacity
Giriş Seviyesi Departmantal Test, Geliştirme Ortamı	Data Warehouse Yüksek Performanslı Analitik	Sorgulanabilir Arşivleme
1 TB to 10 TB	1 TB to 1.5 PB	100 TB to 10 PB

IBM Netezza 1000 Ölçekleme



	1000-3	1000-6	1000-12	1000-24	...	1000-48	...	1000-120
Snippet Processors	24	48	96	192		384		960
Kapasite (TB)	8	16	32	64		128		320
Sıkıştırma a (TB)	32	64	128	256		512		1280

Kapasite = Kullanıcı verisi alanı
Sıkıştırma = Sıkıştırma sonrası kullanıcı verisi alanı

Teşekkürler

Soru ve Cevaplar