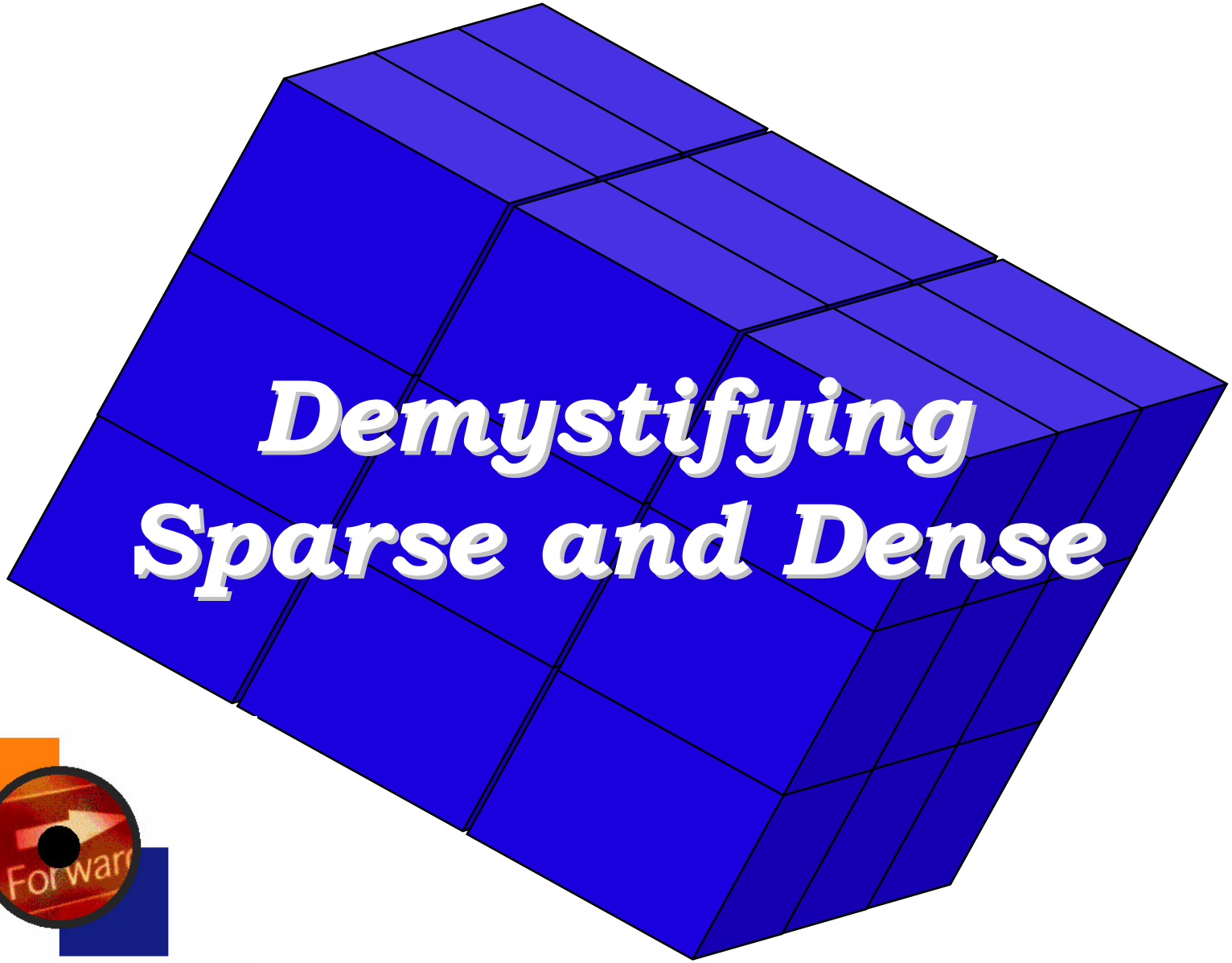
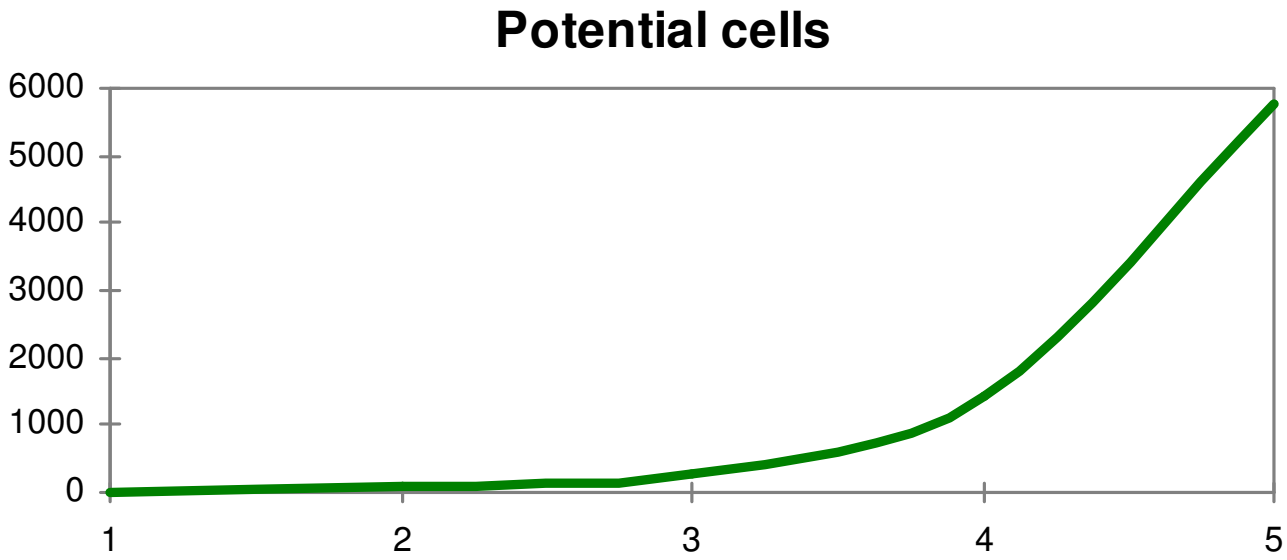


MARKETING TECHNOLOGIES GROUP
New York Information Technology Center:
55 Broad Street, 10th Floor, New York, NY 10004

Even Small Models Grow Fast

Dimensions	Members	Potential cells
1	8	8
2	12	96
3	3	288
4	5	1440
5	4	5760



Data Explosion

- Enter a single input value to a 2-D array

	North	South	East	West	All Regions
Hardware					
Software		100			100
Services					
All Products		100			100

- Three calculated values result

8 :1 in Three Dimensions

8 to 1 in three dimensions

Prod Total

	North	South	East	West	All Regions
Hardware					
Software		100			100
Services					
All Products		100			100

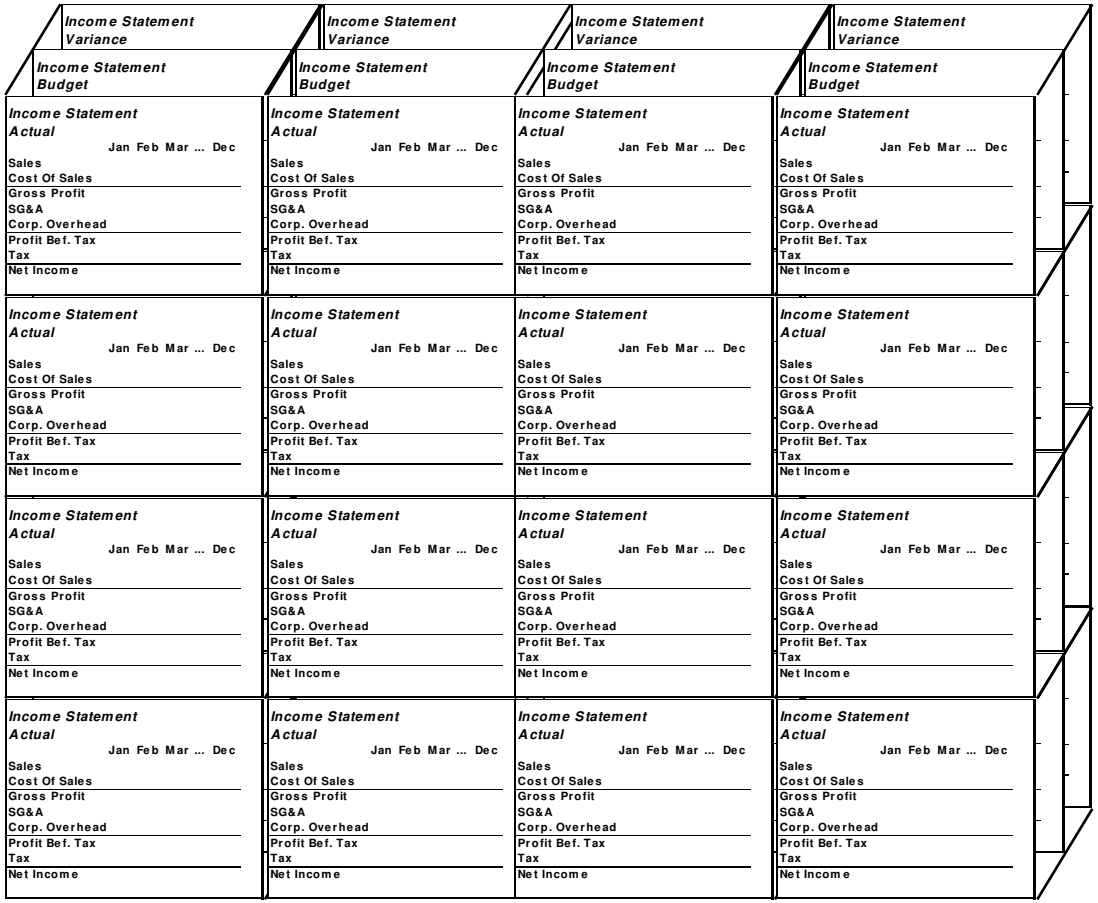
Prod 2

	North	South	East	West	All Regions
Hardware					
Software					
Services					
All Products					

Prod 1

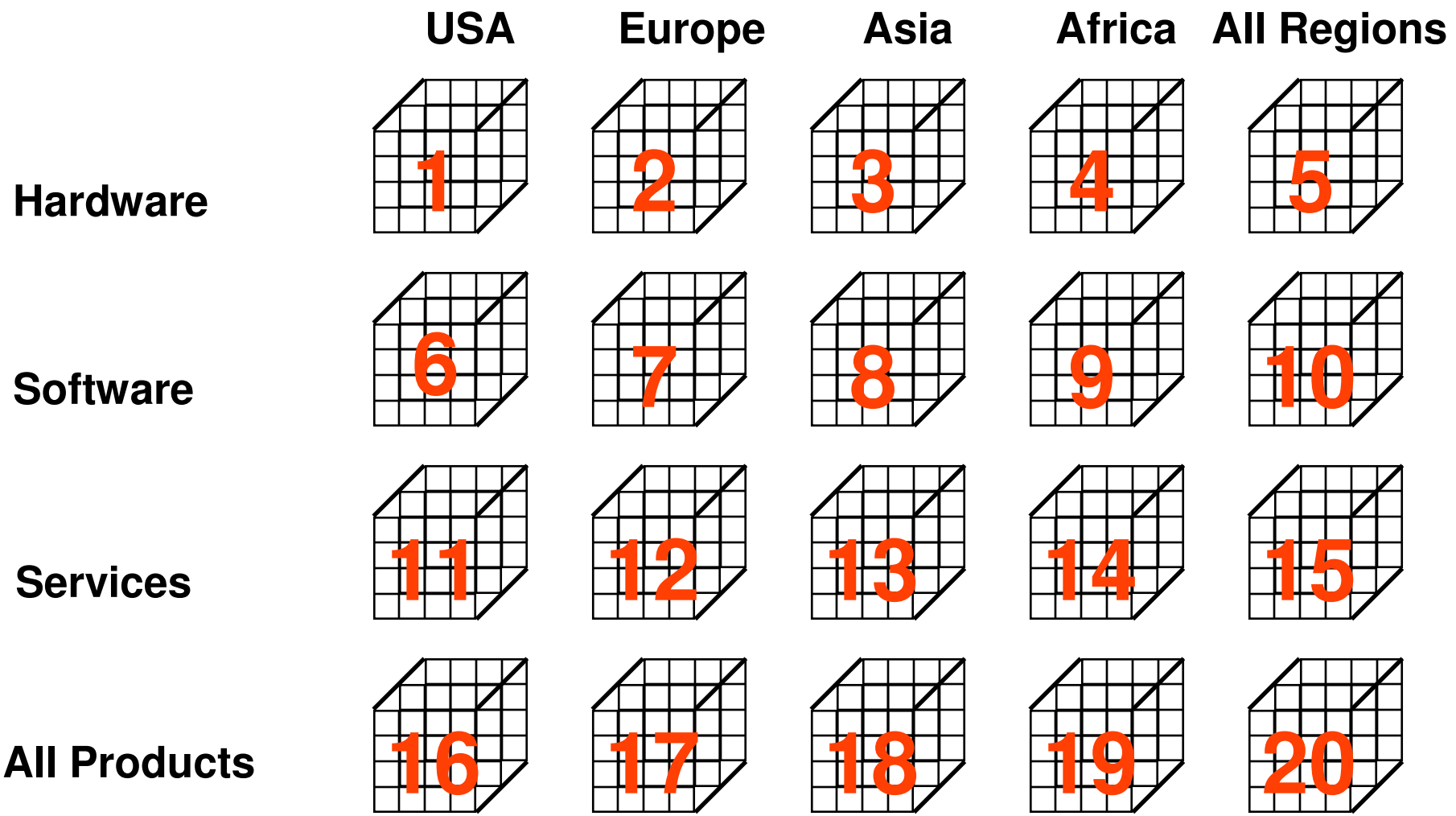
	North	South	East	West	All Regions
Hardware					
Software		100			100
Services					
All Products		100			100

How Should Large Databases Be Stored?

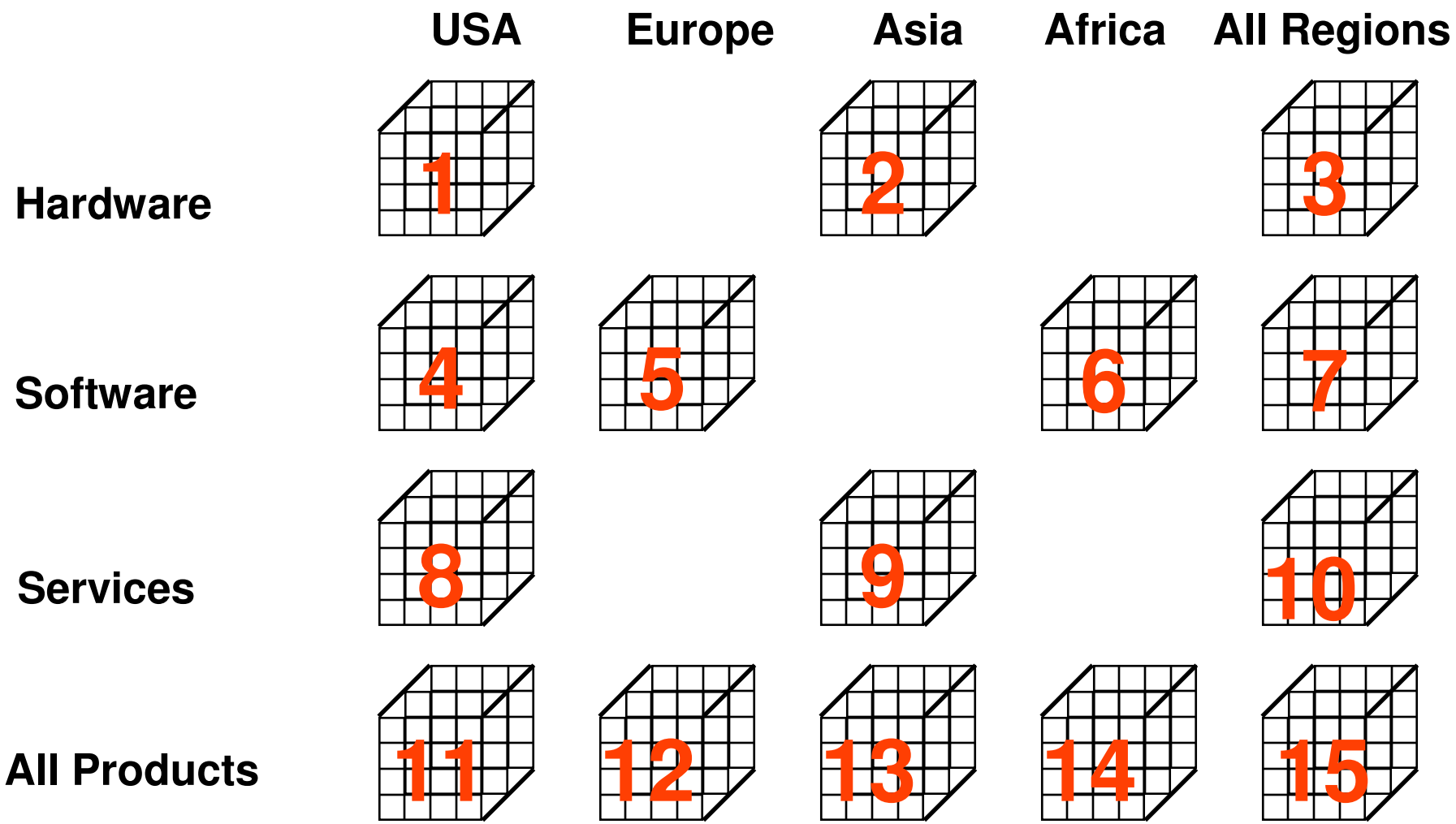


Marketing Technologies Group

Essbase Breaks the Database Into Numbered Blocks

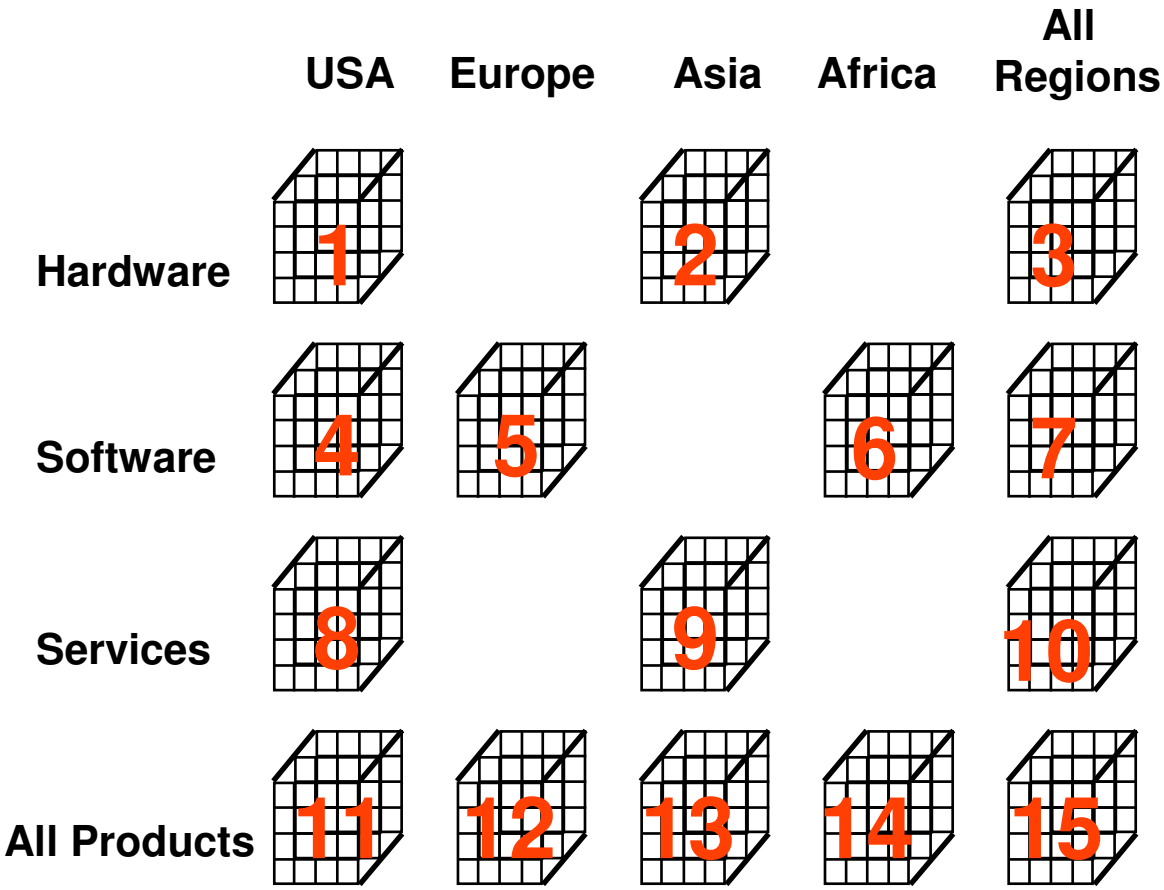


But Some Are Empty...



The Index File Tracks Existing Blocks

Page File



Index File

Block # 1 USA->Hardware
Block # 2 Asia->Hardware
Block # 3 All Reg.->Hardware
Block # 4 USA->Software
Block # 5 Europe->Software
Block # 6 Africa->Software
Block # 7 All Reg.->Software
Block # 8 USA->Services
Block # 9 Asia->Services
Block # 10 All Reg.->Services
Block # 11 USA->All Products
Block # 12 Europe->All Products
Block # 13 Asia->All Products
Block # 14 Africa->All Products
Block # 15 All Reg.->All Products

A Block is…

- The smallest unit of I/O in Essbase (BSO)
- Essbase reads/writes entire blocks
- All data from a block is in RAM at once
- Should be a cluster of data used together

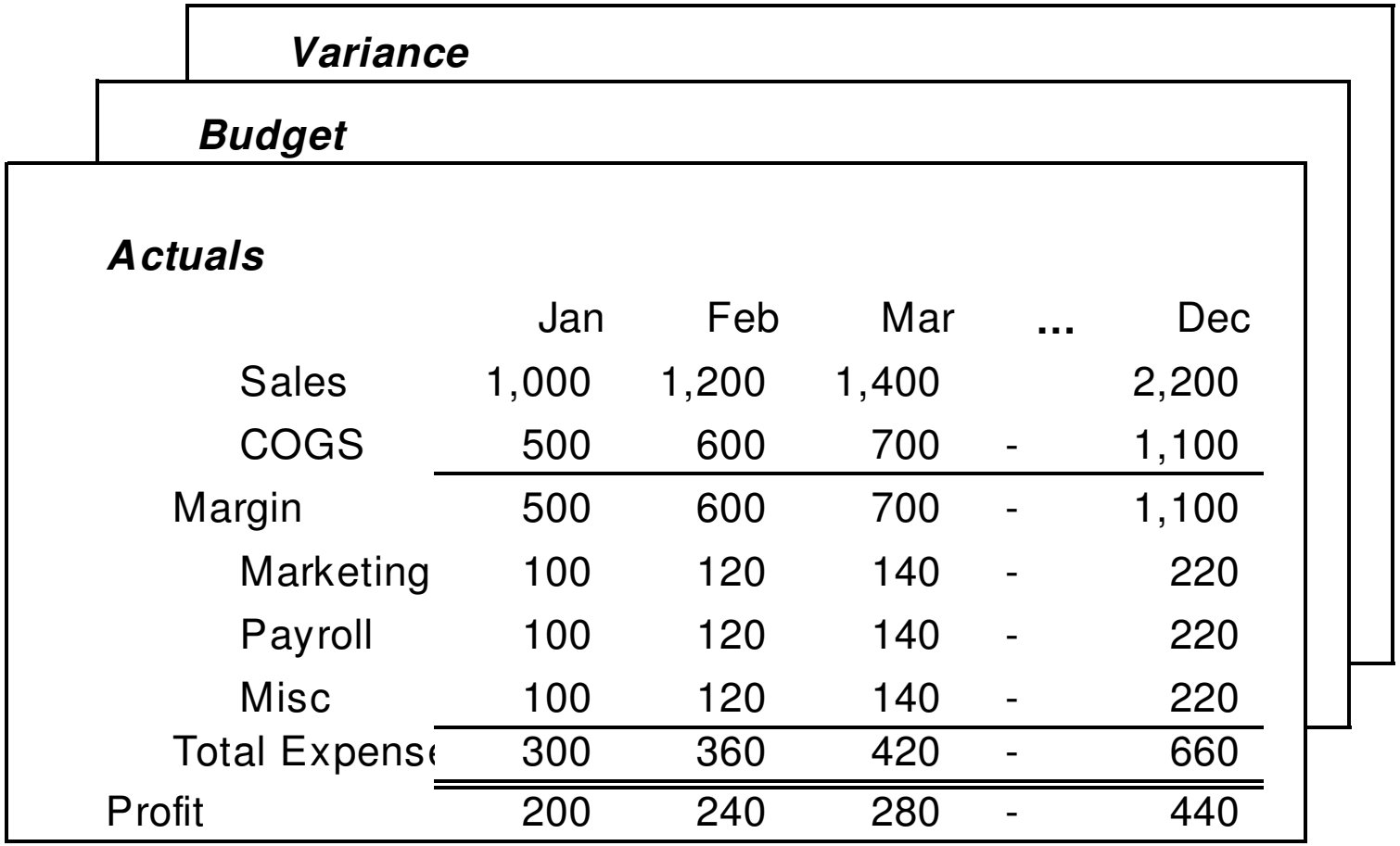
Densely Populated Time X Measures

95%

99%

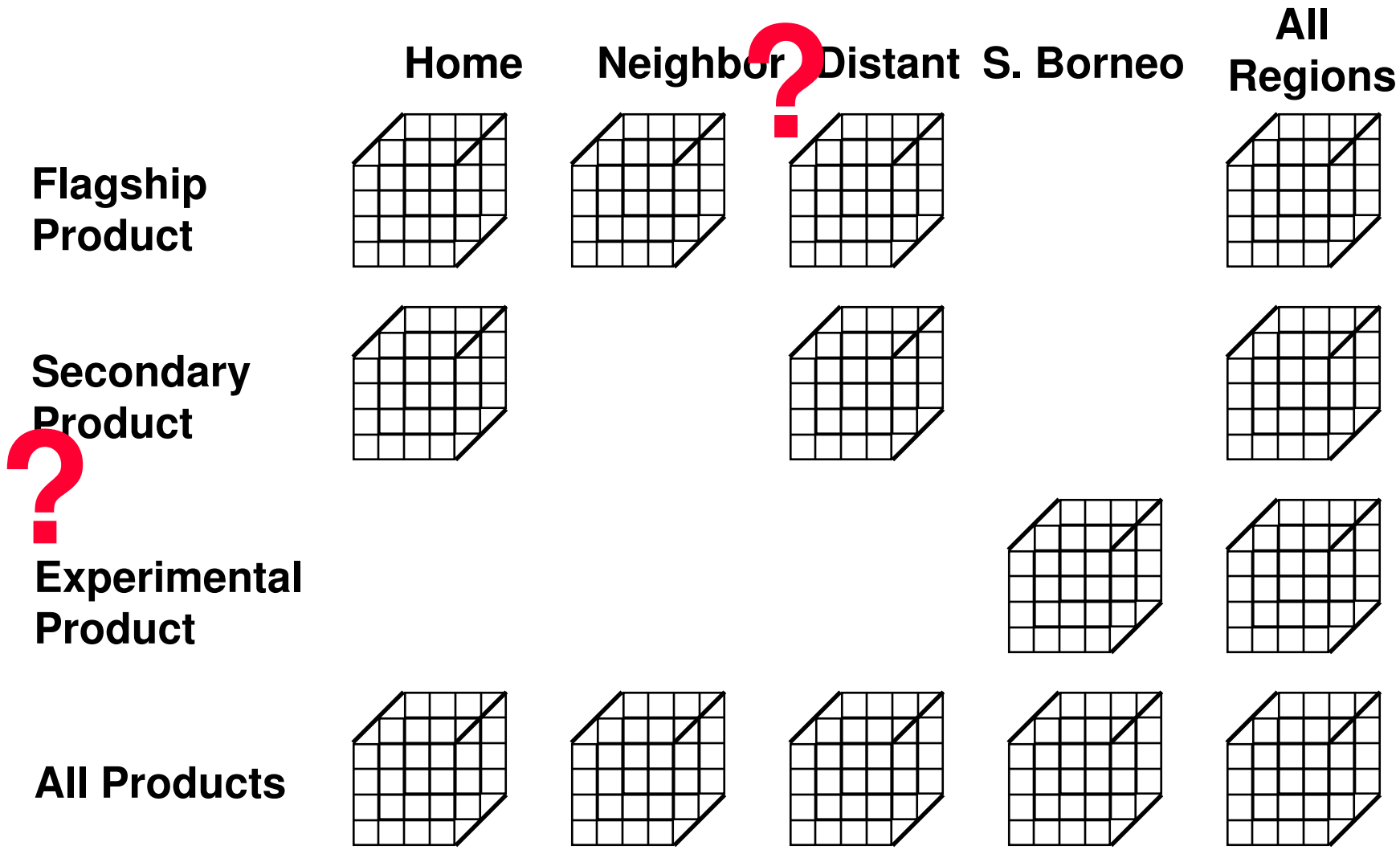
Actuals	Jan	Feb	Mar	…	Dec
Sales	1000	1200	1400	-	2,200
COGS	500	600	700	-	1,100
Margin	500	600	700	-	1,100
Marketing	100	120	140	-	220
Payroll	100	120	140	-	220
Misc.	100	120	140	-	220
Total Expenses	300	360	420	-	660
Profit	200	240	280	-	440

Add a Scenario Dimension



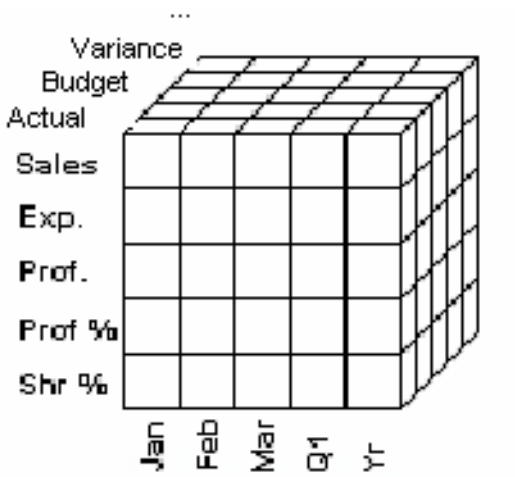
	Jan	Feb	Mar	...	Dec
Sales	1,000	1,200	1,400		2,200
COGS	500	600	700	-	1,100
Margin	500	600	700	-	1,100
Marketing	100	120	140	-	220
Payroll	100	120	140	-	220
Misc	100	120	140	-	220
Total Expense	300	360	420	-	660
Profit	200	240	280	-	440

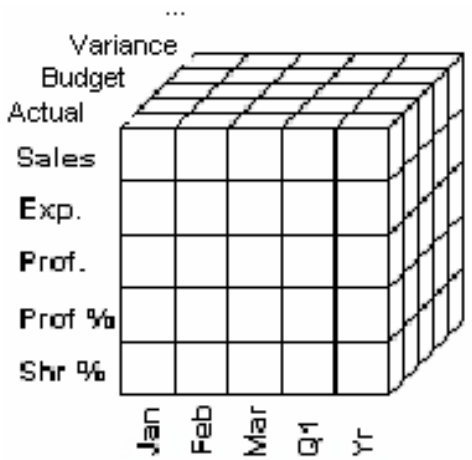
Sparsely Populated Products & Regions



A Block Contains…

- ALL DENSE MEMBER intersections





Asia ->Services **USA->Hardware**

- ONE SPARSE MEMBER intersection

Calculating Block Size

- Cells:
  product of stored members on all dense dimensions,
  e.g. 100 accounts x 12 months = 120 cells

- Bytes:
  cells x 8 bytes per cell
  e.g. 120 cells x 8 bytes per cell = 9600 bytes

Setting Sparse and Dense Dimensions: Rules of Thumb

- Time and Accounts are dense. Everything else is sparse.
- Target block size is 10k to 60k.
- If initial block size is:

  Too small

  -change a small sparse dimension to dense

  Too big

  -change a dense dimension to sparse

Target Block Size: 8k – 80k

8k Bytes/Block	80k Bytes/Block
/ 8 Bytes/Cell	/ 8 Bytes/Cell
1000 Cells/Block	10,000 Cells/Block
/ 10 Time Periods	/ 10 Time Periods
100 Accounts	1000 Accounts

Setting Sparse and Dense Dimensions: Additional Rules of Thumb

- Large dimensions should be sparse.
- Dimensions with many formulas should be dense.
- Dimensions where data is loaded sequentially should be sparse.
- Dimensions with data that is reported together should be dense.

Large Accounts Dimensions

- Accounts is calculated first
- Sparse accounts dim require extra passes
- Larger block is usually better than a sparse accounts dimension
- Easily push block size to 160k (perhaps 2000 stored members) or more
- With dynamic calcs maybe 3000 or more

Large Time Dimensions

- Break apart "All Years" and "Year Total"
- Year Total = dense
- All Years = probably sparse

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Year Total
2003	74	36	72	98	94	34	8	3	75	13	75	60	641
2004	11	121	85	41	41	21	35	44	34	141	95	93	763
2005	105	134	161	193	12	95	98	159	72	132	27	82	1,270
2006	29	30	7	23	187	28	104	242	9	220	234	37	1,150

Sparse Time Dimensions

- Probably good for time based incremental loads using intelligent calc
- Probably bad for growth rates, activity based cash flow statements etc.

Choosing Dense/Sparse Settings

- You pay by the block
- Fewer blocks is good

  (smaller index)
- Denser blocks is good

  (fewer blocks, RAM used more efficiently)
- Ideal block size; less than 100k (flexible).