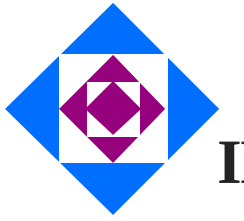
Performance Modeling of Storage

Ariel Landau
Nava Aizikowitz
Daniel Fishkov
Bilha Mendelson

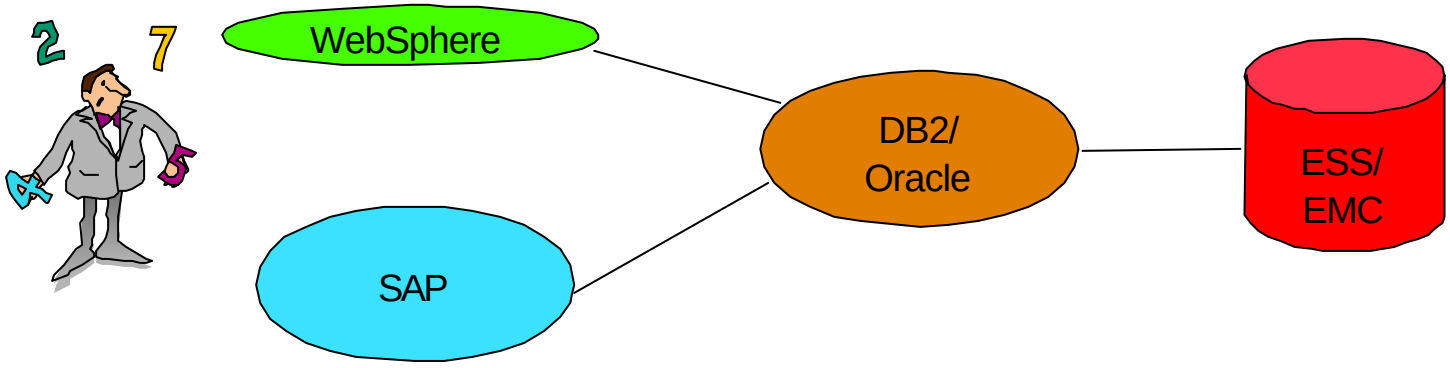
November 21, 2002



IBM Research Lab in Haifa

Background

- **Application stack components:**
  - Application - Database - Storage
- **Existing configuration tools**
  - adjust each component separately
  - are complicated
  - require high expertise
  - focus on ease-of-administration rather than performance
- **None for the application stack as a whole**



Our Goal

- **Address performance issues of system-level interactions in application stacks**
  - Focus on I/O aspects
- **Provide knowledge and tools for the development and integration of improved performance policies**
- **Expected interest**
  - Performance-tuning experts
  - Marketing teams
  - Architects

How

- **Storage performance model**
  - Simulation-based
- **Performance of system-level interactions**
  - Feed storage model with I/O traces from real and synthetic hosts
- **Impact of I/O on performance**
  - Analyze sensitivity to configurations
  - Data placement

Related Work

- **Building actual environment for measurements**
  - A model is more flexible and amenable to modifications
  - A modeling solution can address future directions
- **Analytic queuing model for steady state behavior**
  - A simulation model captures dynamic behavior

Current Status

- **Storage performance model - working prototype**
  - Configured to represent ESS
  - Focused on Open System hosts, Fibre Channel attachments and RAID-5

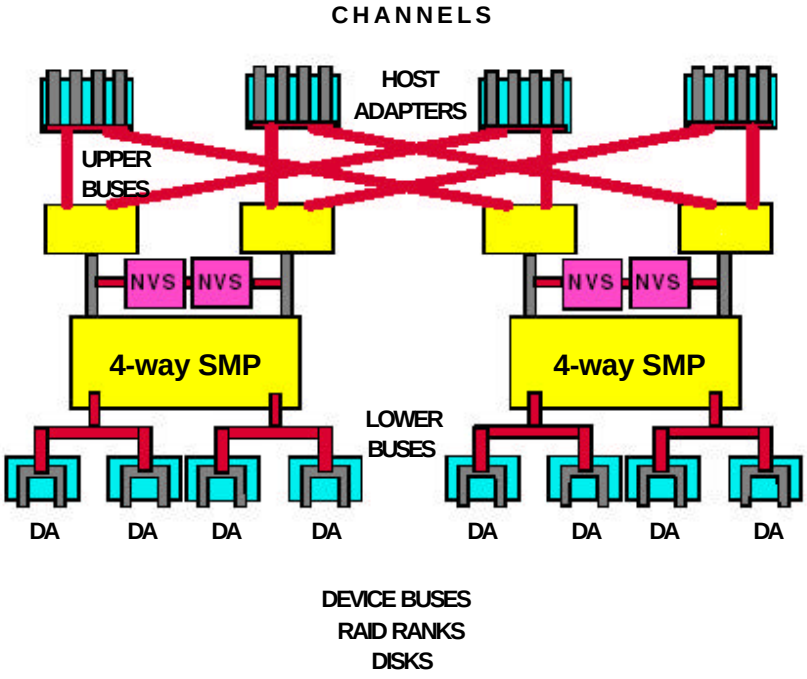
- **Stack analysis**
  - Tracing of DB2 I/O on AIX
  - Through AIX trace facility
  - Experimetation with
    - data placement
    - host configuration

Storage Performance Model

- **Simulation-based queuing model**
  - Built on top of CSIM simulation engine
  - Receives request attributes
    - read/write
    - target address
    - data amount
    - time stamp
    - channel
  - Simulates transaction processing paths
    - read, fast write, prestage, destage
  - Collects relevant statistics

Model Paths

- **Read**
  - Cache hit due to prestaging
  - Data staging due to cache miss
- **Write**
  - Fast write into non-volatile storage (NVS) and cache
- **Prestage**
  - Data staging triggered by sequential read pattern
- **Destage**
  - Flushing of cache written data to storage triggered by NVS and cache thresholds



Model Output

- **Detailed per-resource output**
  - Utilization
  - Interarrival times
  - Service times
- **Configurable statistical output**
  - Transaction response time
  - Data throughput
  - Cache-hit rates

FACILITY SUMMARY

facility name	service disc	service time	util.	through-put	queue length	response time
CHAN 0	fcfs	51.47727	0.000	0.00000	0.00012	52.26980
CHAN 1	fcfs	49.21096	0.002	0.00003	0.00160	50.24845
CHAN 4	fcfs	62.04545	0.000	0.00000	0.00008	65.53918
CHAN 5	fcfs	55.66645	0.005	0.00008	0.00467	57.52539
CHAN 8	fcfs	91.00000	0.000	0.00000	0.00003	91.00000
HA 0	fcfs	63.58651	0.000	0.00000	0.00015	65.31378
HA 1	fcfs	63.57987	0.002	0.00003	0.00204	64.19720
HA 4	fcfs	89.70000	0.000	0.00000	0.00015	94.36667
HA 5	fcfs	65.50598	0.005	0.00008	0.00553	66.59390
HA 8	fcfs	104.00000	0.000	0.00000	0.00007	104.00000
UPBUS 0	fcfs	23.05805	0.002	0.00007	0.00161	23.53411
UPBUS 1	fcfs	26.28246	0.004	0.00017	0.00468	27.67147
UPBUS 2	fcfs	30.50000	0.000	0.00000	0.00004	30.50000
SMP 0	fcfs	181.93348	0.025	0.00014	0.03330	239.55201
> server	0	169.52879	0.021	0.00012		
> server	1	259.36551	0.003	0.00001		
> server	2	272.17193	0.001	0.00000		
> server	3	327.40426	0.001	0.00000		
SMP 1	fcfs	171.26770	0.009	0.00005	0.01024	191.40590
> server	0	164.53591	0.008	0.00005		
> server	1	237.23529	0.001	0.00000		
> server	2	219.40000	0.000	0.00000		
> server	3	340.00000	0.000	0.00000		
DA 0	fcfs	306.51122	0.013	0.00004	0.09599	2265.56512
DA 1	fcfs	215.29003	0.004	0.00002	0.01497	856.25925
HDD 0	fcfs	5128.28721	0.004	0.00000	0.00604	8165.79142
HDD 2	fcfs	5178.70855	0.002	0.00000	0.00164	5178.70855
HDD 3	fcfs	4973.74915	0.024	0.00000	0.02796	5752.09267

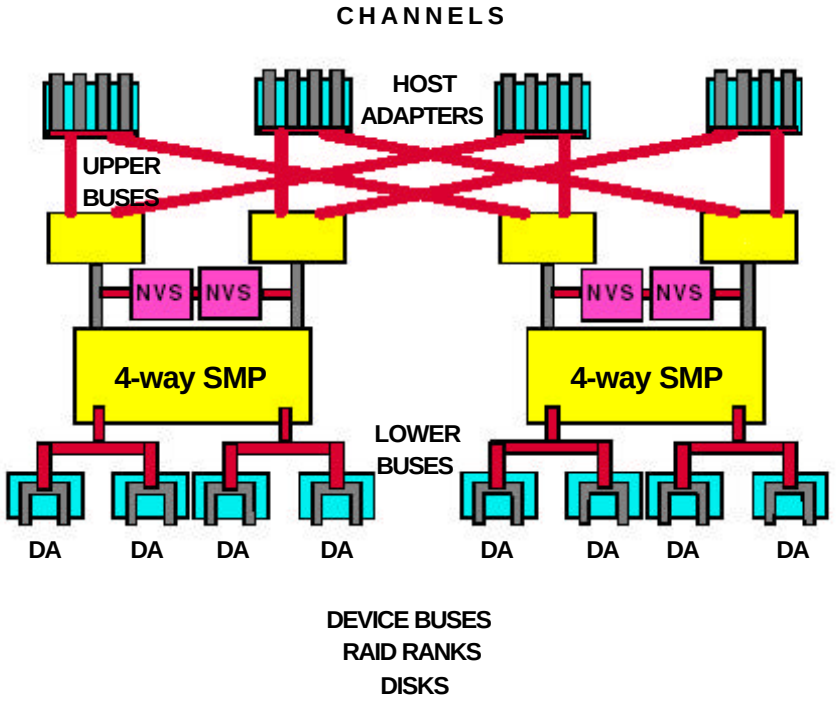
...

Model Flexibility

- **May represent different storage subsystems (ESS, EMC)**
  - Configurable collection of resources
  - Configurable time overheads for each resource

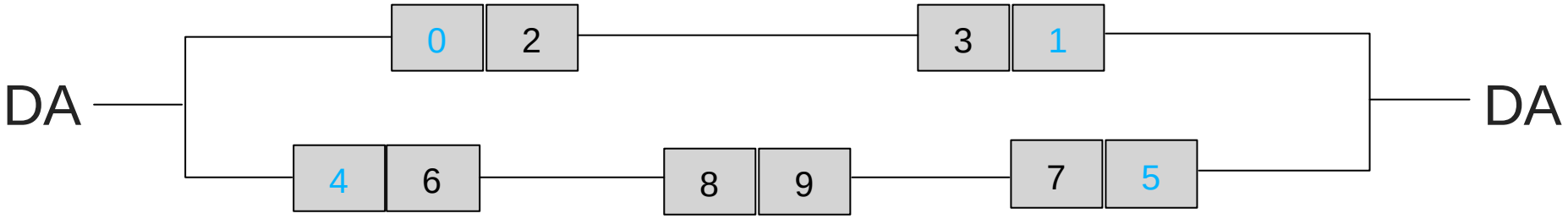
```
// Resources
#define CHAN_NUM 16
#define HA_NUM 16
#define UPBUS_NUM 4
#define SMP_NUM 2
#define CPU_NUM 4 // CPUs per SMP
#define NVS_NUM 2
#define LOWBUS_NUM 4
#define DA_NUM 8
#define DEVBUS_NUM 32
#define HDD_NUM 384

// Overhead values for Ess Fxx Fibre1
// (in microseconds)
#define CHAN_OVHD 46
#define HA_OVHD 104
#define HA_ONBUS_OVHD 8
#define HA_CLNUP 50
#define HA_CLNUP_ONBUS 50
#define SMP_CLNUP 20
#define SMP_CLNUP_ONBUS 6
#define SMP_OVHD 110
#define SMP_ONBUS_OVHD 50
#define NVS_OVHD 390
#define SMP_ALLOC_OVHD 100
#define DA_OVHD 333
#define NVS_ASYNCH_OVHD 160
```



Example - Data Placement

- **Assignment of Logical Volumes (LV) to RAid-5 Ranks (RR)**
- **An example**



- "blue" RRs are 6 + P + S
- "black" RRs are 7 + P
- "even" RRs are handled by cluster 0
- "odd" RRs are handled by cluster 1

Data Placement - Results

- **Semi-synthetic database trace with intensive I/O bursts**

Configuration	**Average Read Response Time** (ms)	**Improvement** (w.r.t. previous configuration)
Empty-system read-miss	11	--
One LV 6+P RR	43	--
One LV 7+P RR	35	19%
Two LVs Two 7+P RRs Same cluster	23	34%
Two LV Two 7+P RRs Different clusters	21	9%

Example - Join

- **Join I/O requests**

- Example

	Timestamp	LV	Blk-Add	Amount	
►	920433	0	BF168	1000	READ
	922033	0	BF190	1000	READ
	923916	0	BF198	3000	READ
	924154	0	BF1B0	4000	READ
	924531	0	BE200	1000	READ
	925755	0	BE210	3000	READ

Example - Join

- **Join I/O requests if**
  - consecutive in space - space difference (SD) = 0

- Example

	Timestamp	LV	Blk-Add	Amount		SD
►	920433	0	BF168	1000	READ	
	922033	0	BF190	1000	READ	
	923916	0	BF198	3000	READ	0
	924154	0	BF1B0	4000	READ	0
	924531	0	BE200	1000	READ	30
	925755	0	BE210	3000	READ	8
►	924154	0	BF190	8000	READ	

Example - Join

- **Join I/O requests if**
  - consecutive in space - space difference (SD) = 0, **and**
  - not too "close in time" - timestamp difference (TD) > 10 us
- Example

	Timestamp	LV	Blk-Add	Amount		SD	TD
►	920433	0	BF168	1000	READ		
	922033	0	BF190	1000	READ		
	923916	0	BF198	3000	READ	0	1883
	924154	0	BF1B0	4000	READ	0	238
	924531	0	BE200	1000	READ	30	377
	925755	0	BE210	3000	READ	8	1224
►	924154	0	BF190	8000	READ		

Join - Results

- **Semi-synthetic database trace with intensive I/O bursts**

Configuration	**Number of Joins**	**Average Read Response Time** (ms)	**Improvement** (w.r.t. base configuration)
Base Trace	--	39	--
All Joins	60	35	9%
Safe Join	33	32	18%
Best Join	50	29	26%

Future Directions

- Extend storage model
- Extend analysis of application stack
- Integrate into network performance model
- Integrate with monitoring environment
- Develop modeling-based reasoning