FBSNG and Disk Farm - parts of large cluster infrastructure

Large Cluster Computing Workshop Fermilab October 21-22, 2002

Igor Mandrichenko, FNAL/CD/ISD

FBSNG – Farm Batch System (Next Generation)

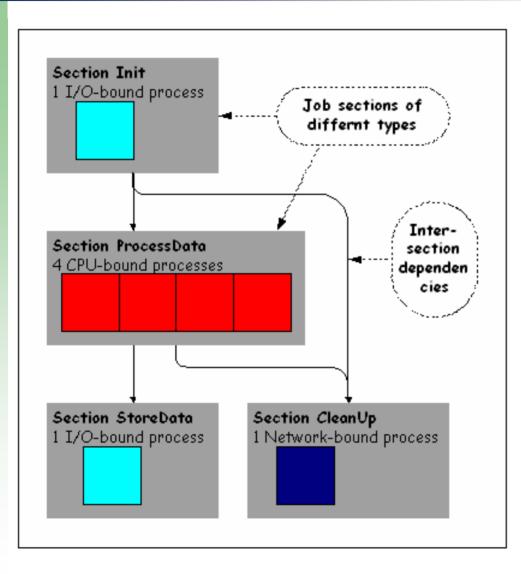
- Developed at FNAL in 1997-98 for Run II production PC farms
- From event parallelism (CPS) to file parallelism (FBS)
- FBS production since 1998
 - was dependent on LSF as scheduler
- In 2000 was redesigned not to use LSF FBSNG
 - in production since 2000

URL: http://www-isd.fnal.gov/fbsng

FBSNG Concepts: Resource Counting

- Instead of load measurement, resource counting:
 - Know resource capacity of farm nodes
 - Know process resources requirements
 - Know which process runs where
 - Start new process when and where resources are available
- Makes the system *simple, robust, flexible, portable*

FBSNG Concepts: Job Structure



- Unit of operation is an array of batch processes (job section)
- FBSNG job consists of (dependent) sections

FBSNG Concepts: Abstract Resources

- Abstract Resources
 - All resources in FBSNG are *abstract* counted semaphores
 - Local resources associated and available locally on farm nodes
 - CPU
 - Disk
 - Tape drives
 - **Node attributes** features of farm nodes
 - OS flavor
 - Installed software
 - Logical attributes ("red", "green", used to partition the farm)
 - Global resources resources shared by all the processes on the farm
 - network throughput
 - NFS-exported disks
 - global semaphores
- No predefined resources
- Allows high flexibility in farm/cluster configuration and management

FBSNG Features

- Scheduler
 - Task/project/group prioritization
 - Fair-share scheduling
 - Guaranteed scheduling
 - Resource utilization quotas
- Dynamic farm (re)configuration
- Robustness with respect to failures of individual farm nodes
- Kerberos support
 - Client authentication ability to access over WAN
 - Creates credentials for batch processes
- Easily portable: supported on Linux, IRIX, SunOS, OSF1
- Recently added Globus Job Manager interface
 - GSI support

FBSNG User Interface

X CURR	ENT JOBS(DO	Farm)											_ [] >	<
Actions	Status	Monitor	S	tatistics	l	.09	Refresh	Prefe	rence	Ph	int Sa	ave	Quit	
)B_ID SECTION_NAME USER QUE				EUE STATUS START				FINISHED						
	job	106		4 - · · 4 D · · · · · ·			C-4 4		2.00.2	0.01				
	_JOB_CONTROL START_SAM			tartQueue tartQueue			Sat_Aug_25 Sat_Aug_25							н
	WORKER_JOB			itaniumQ		-	Sat_Aug_25							Ш
	END	dOfar	m E	ndQueue	wai	ting	-			-				Ш
	job _JOB_CONTROI	d0far	n s	tartQueue	PUR	nina	Sat_Aug_25	: 14·4	2.16.2	001 -				4
6262 START_SAM		dOfarm StartQueue ru							-14:42:17_2001 -					
	WORKER_JOB	d0far	m T	itaniumQ	rur	ning	Sat_Aug_25	5_14:4	2:18_2	001 -				
	END *iob*				on	the	web							
6280	_JOB_CONTROI START_SAM			Farm: Time: Report:	DO Thu Aug 30 11:21:14 List of queues			2001						
6280	WORKER_JOB			Queues	lopz	Node	s <u>Process</u>	Type:	<u>5 Gra</u> g	o <u>hs</u>				
	END *job*	Other					Defau	1+						
	_SAM_JOB_SUE	farms: DO		Name		Status	Process		Share	Pr10	Waiting	Ready	Running	יו
	START_SAM	CDF		<u>BlueQ</u>		ОК	Worker_6		1.00	0	0	0		1
	WORKER_JOB	<u>fnsfo</u>		<u>CadmiumQ</u>		ОК	Worker_13	3	1.00	0	0	0	(0
	_SAM_JOB_END			Chartreus	seQ	OK	Worker_9		1.00	0	0	0	(D
6329	*job*	prototyp	e	<u>CobaltQ</u>		ОК	Worker_14	ŧ	1.00	0	0	0	(5
\triangleleft				DoveQ		ОК	Worker_te	st_1	1.00	0	0	0	(С
		-		EndQueue		ОК	Endsam		1.00	50	15	0	(5
				GreenQ		ОК	Worker_2		1.00	0	0	0	(5
												_		5
				CPV utilia	zati	on by	project (#eek)					(5
	120												(5
	100												(5
	100												(5
Number of CPUs	80		_ 1			_			1	1	· .			5
of	60													5
ber	60				- <u></u>								3′	
Num	40				<u>+</u>								12	
					-									
	20													-
	0-											Þ		
	Endsom		+ < 64					or 11_				er 13	, i	,
	20 0- EndSAM J Worker_14 Worker 7	Fri O Star Vorker_1	5	📕 Worker_:		Sun Worker_ ■ Worke	r_3 🛛 🗖 Wo	r_11 rker_4		ker_12 orker_		rker_6		0

- Command line interface
 - Job submission, control
 - Farm management
- GUI
 - Job monitoring, control
 - Farm management
- Python API
 - Provides full functionality
 - UI, GUI, FBSWWW use API
- Minimal requirements for client-only installation.
 - Access over WAN
- Web interface (FBSWWW)
 - Resource/job monitoring
 - Node status monitoring

Updated: Thu Aug 30 11:33:13 2001

0 33 14

0

0

FBSNG – Experience

- Currently managed farms:
 - CDF, D0 on-line production farms (150+ computers each, growing)
 - "Fixed Target" (common use) farm (106+2 computers)
 - CMS USA Tier 1 center (3 farms ? see Hanses talk)
 - CAF CDF Analysis Farm
 - NIKHEF
 - Other HEP sites, one known corporate site
- FBSNG is full scale batch system for farms and clusters
 - Has worked well on farms of different sizes with various kinds of users and resource utilization patterns
 - Robust, low maintenance, easy to deploy, support and manage

Disk Farm - Distributed Data Storage

- Typical computing farm can be viewed as an array of disks controlled by an array of CPUs, or **disk farm**:
 - Capacity: 100 nodes * 2 * 30 GB = 6TB
 - Throughput: 10 MB/s / node * 100 nodes = 1 GB/s
- Price: ~100GB disk for \$100 -> \$1/GB
- Typically, each 30 GB disk is managed by 1GHz CPU
 - Compare to ~1TB and 8 300MHz CPUs (d0bbin)
- BUT, utilization is difficult:
 - Highly distributed storage
 - Unreliable components
 - Access and allocation must be coordinated
- Disk Farm is a product which helps utilize large unused disk capacity of farm nodes

Disk Farm – Distributed Data Storage

- Organizes distributed disk space spread over nodes of the farm into global *virtual* name space
 - Virtual file path: /e123/runll/data/mc123.dat
 - Physical file path:

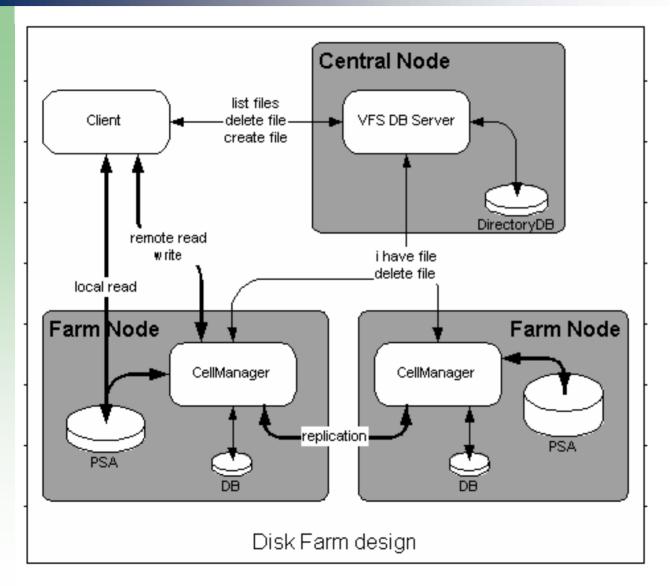
node1:/local/stage1/dfarm/xyz123

- User interface:
 - get, put, mkdir, rmdir, rm, ls commands similar to UNIX FS access commands. E.g.:
 - \$ dfarm mkdir /e123/runII/data
 - \$ dfarm put /scratch/mc123.dat /e123/runII/data
 - \$ dfarm ls /e123/runII/data
 - On the node where the data happens to be:
 - "get" is local faster, cheaper
 - Ability to read data without copying out of disk farm
 - "put" is amlost always local
 - Users are limited by (optional) global quotas, not physical sizes of individual volumes
 - POSIX semantics access being considered

Disk Farm Features

- Data replication:
 - User: here is my file, make 3 copies of it on 3 different nodes
 - In case 2 nodes go down, 1 copy is still available
 - Replication is performed off-line, without user waiting
 - Data can be re-replicated later to maintain desired number of replicas
 - If a node is to go down, its contents can be replicated
- Load management and balancing
 - Automatically chooses one of least busiest nodes
 - Configurable limits on number of active transfers per node
 - Nodes can be made read-only (put on hold)
- Scalability
 - Scales naturally with the farm size
- WAN access
 - Kerberized FTP server
 - Work on GSI GridFTP interface

Disk Farm Design



Disk Farm - Experience and Status

- Robust, reliable, low maintenance product
- Best if used as a temporary storage of data on the farm
- Current installations:
 - CFD production farm 6.7 TB on 170 nodes
 - D0 farm: 2.1 TB on 180 nodes
 - Fixed Target farm: 1.8 TB on 90 nodes

URL: http://www-isd.fnal.gov/dfarm

Farm Resources Management Tools

- FBSNG and Disk Farm are farms infrastructure tools used to organize and coordinate use of resources provided by computing farms:
 - CPU power
 - Local disk
 - Inter-node network bandwidth
- They provide local and, trough Grid interface, remote access to computational resources