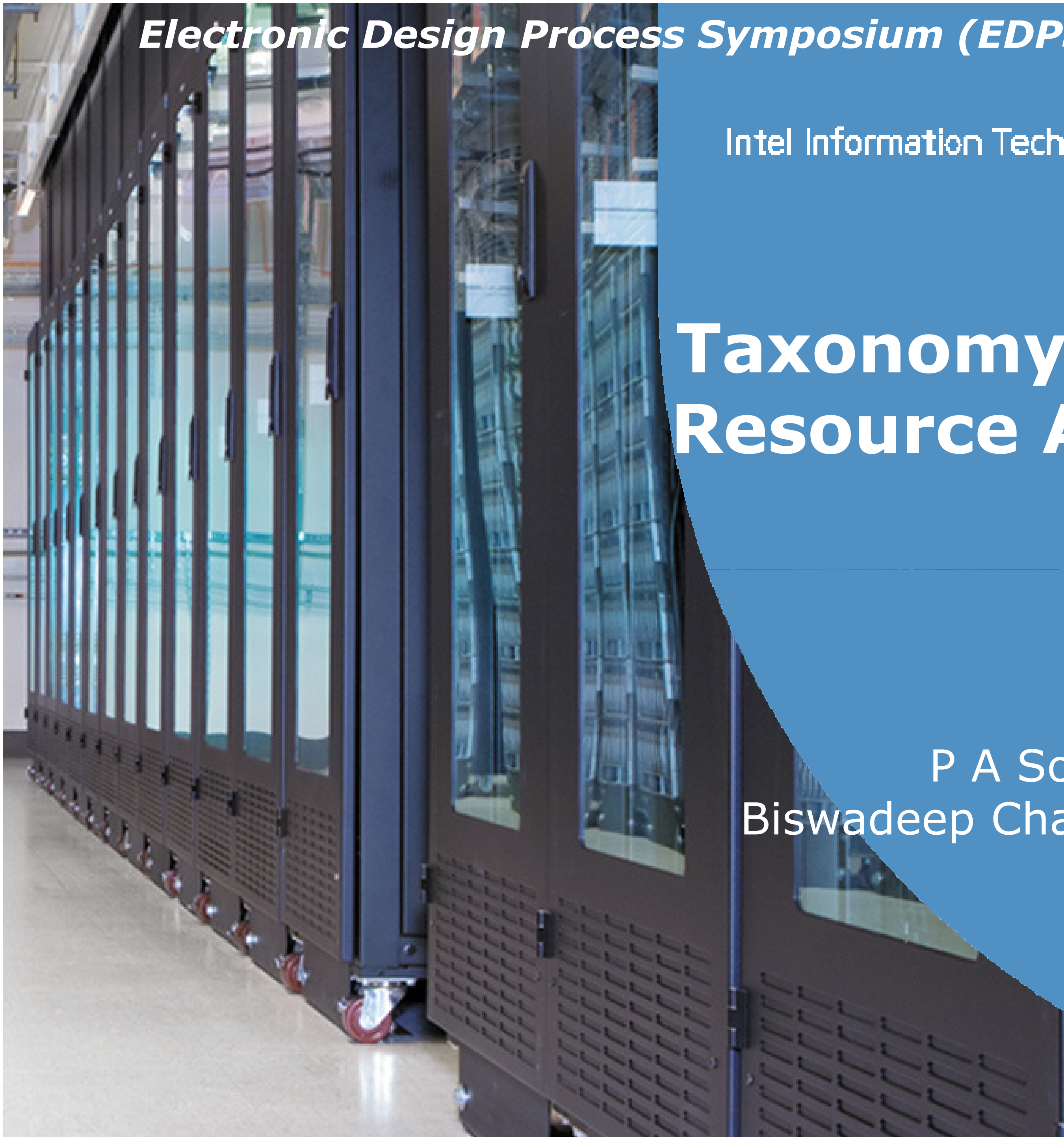
Electronic Design Process Symposium (EDPS) 2011

Intel Information Technology

Taxonomy Oriented Resource Allocation

P A Soujanya, Intel IT
Biswadeep Chatterjee, Intel IT

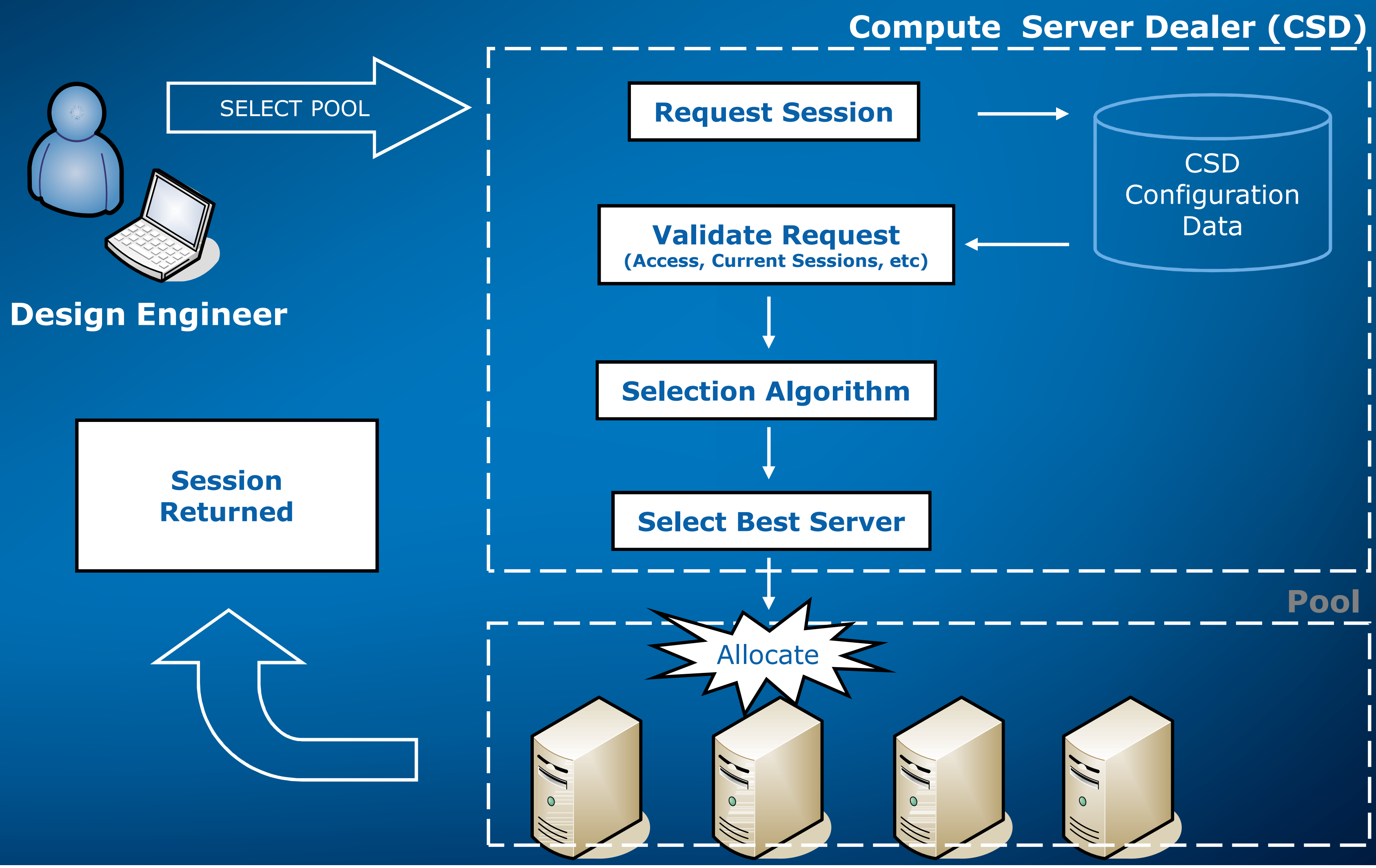


Agenda

- Introduction
- Problem Statement
- Proposed Solution
- High Level Architecture
- Summary

"All observations/comments expressed in this presentation represent the views of the speaker only in personal capacity and does not represent Intel in any manner whatsoever"

Interactive Computing - Workflow



Online design engineer time =~ Time spent in a interactive session

Challenges with Current Flow

- 'Start' and 'End' points of an interactive session not clearly defined – user owns the allocated session and there is no control on number of jobs that can be invoked.
  - "Utilization and Productivity" both are decided based on type and number of jobs run by user(s) on server.
- Users tend to request for resources without clear understanding of their computing tasks requirements
  - Users maintain the same session allocated to them by CSD (resource brokering tool) even when their resource needs have changed drastically

Impact

•**Overall server performance**

- *High load can cause system crash/hang/reboot..*

•**Interactive pool balance**

- *Some machines in pool are "too high" where as others are "too low"*

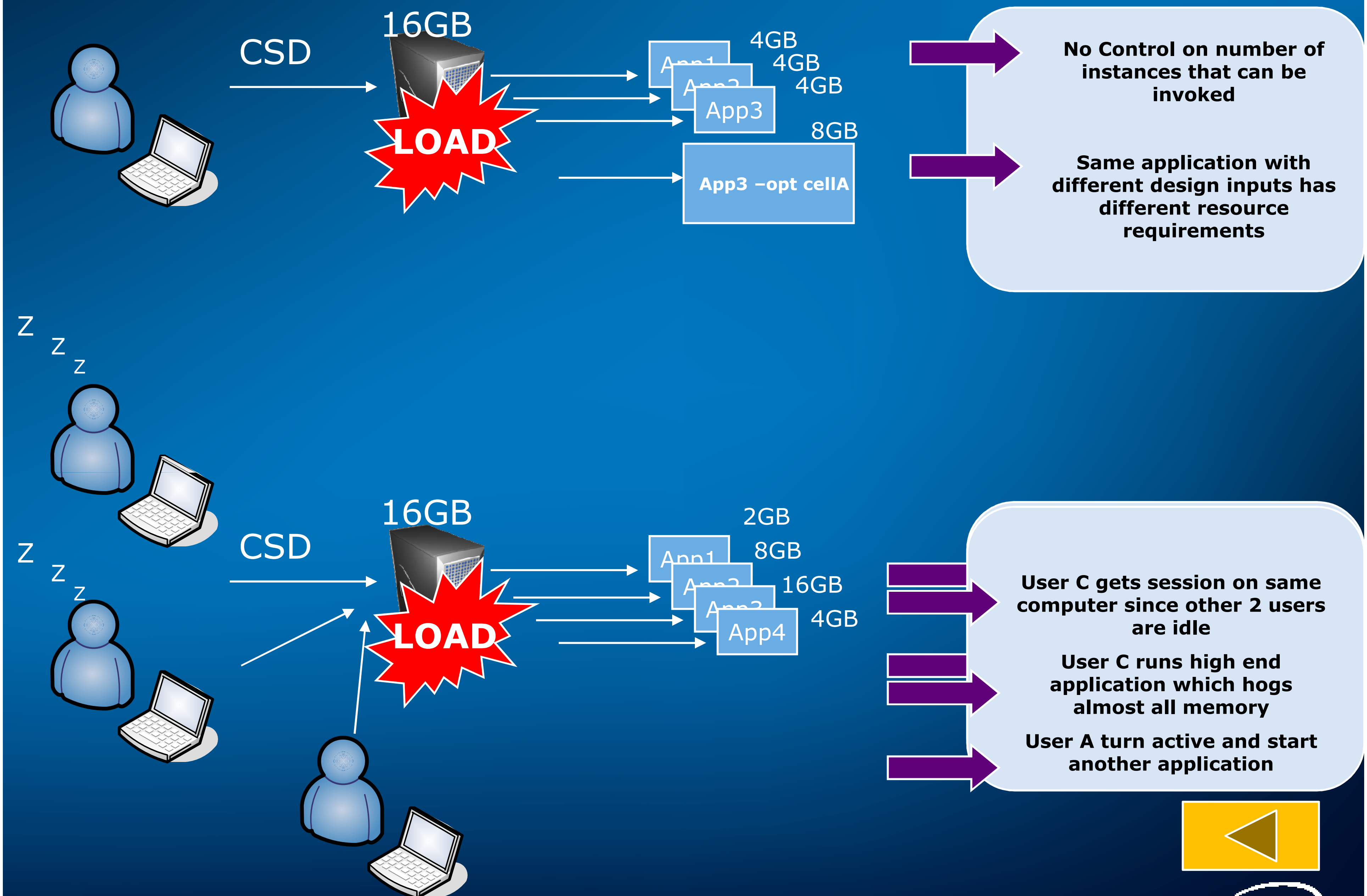
•**Interactive server utilization**

- *Resource allocation is not right sized to workloads. Low end jobs can hog high end servers thus preventing critical high end jobs from running on them when needed*

•**Design engineer productivity**

- *High load, resource imbalance, incorrect allocation*
- *Users sticking to same session will prevent another user in need for an interactive session wait for resources*

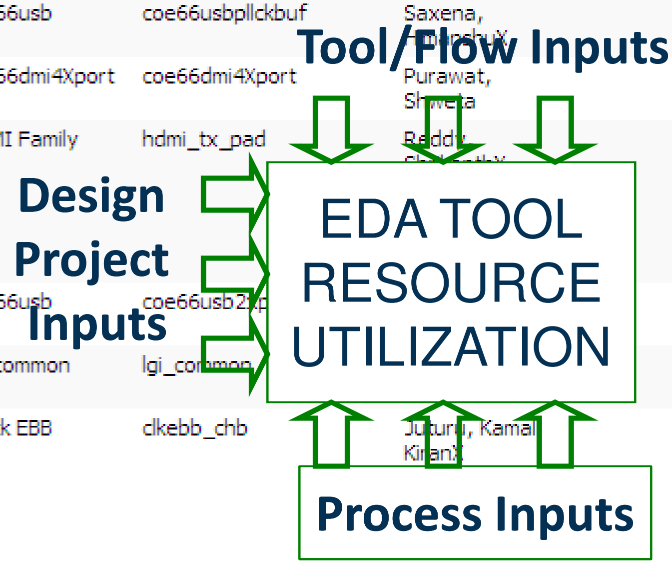
Example



16GB
CSD
LOAD
4GB
4GB
4GB
App3
8GB
App3 –opt cellA
No Control on number of instances that can be invoked
Same application with different design inputs has different resource requirements
Z Z Z
Z Z Z
16GB
CSD
LOAD
2GB
8GB
16GB
4GB
App4
User C gets session on same computer since other 2 users are idle
User C runs high end application which hogs almost all memory
User A turn active and start another application

What drives Design compute needs?

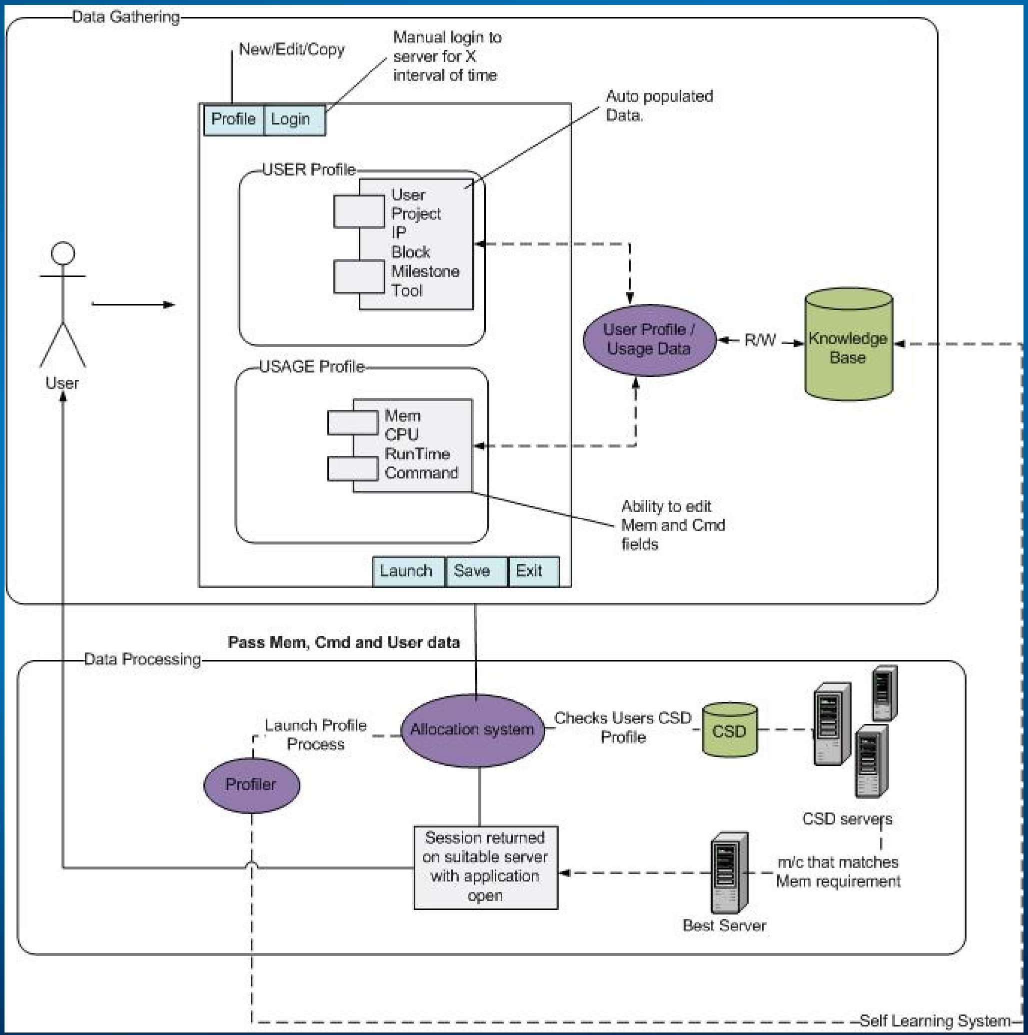
Tool/Flow	IP	IP Lead	Blocks	Cell Names	Owner	Memory Usage (MB Real)	Total Run Time	Total CPU time	Job id
Carmel	DAC-CRT	Thakare, Shivraj; Sahoo, Ranjan, Rao, Venkatesh	crt-dac	dactop	Juturu, Kamal KiranX	59,964	5D:19H:56M:8S	4D:20H:13M:17S	Job6620
Carmel	SDV_USB	Validation - Gaurav	coe66usb	coe66usbpllckbuf	Saxena, [illegible]	41,243	3D:17H:24M:44S	2D:7H:28M:16S	Job 6684
Carmel	PNV-B0-DMI	Validation - Gaurav	coe66dmi4Xport	coe66dmi4Xport	Purawat, Shweta	32,316	4D:2H:24M:5S	3D:15H:31M:48S	Job 5887
Carmel	DDR	Thakare, Shivraj; Sahoo, Ranjan, Rao, Venkatesh	HDMI Family	hdmi_tx_pad	Reddy, [illegible]	31,334	6D:22H:6M:17S	6D:1H:12M:44S	Job 7429
Carmel	SDV_USB	Validation - Gaurav	coe66usb	coe66usb2[illegible]	[illegible]	27,501	2D:19H:28M:1S	1D:21H:18M:2S	Job 6657
Carmel	PNV-B0-LGIO	Validation - Gaurav	lgi_common	lgi_common	[illegible]	20,975	2D:14H:3M:31S	0D:18H:29M:39S.	Job 5674
Carmel	DDR	Thakare, Shivraj; Sahoo, Ranjan, Rao, Venkatesh	Clock EBB	clkebb_chb	Juturu, Kamal KiranX	20,480	16:27:39	16:05:14	Job6615
Carmel	DDR	Thakare, Shivraj; Sahoo, Ranjan, Rao, Venkatesh	DLL-FIFO	dqdllfifo	Juturu, Kamal KiranX	19,443	0D:18H:15M:13S	0D:11H:7M:17S	Job 6635
Carmel	DDR	Thakare,	Clock EBB	clkebb_cha	Juturu, Kamal	18,000	24:00:00	16:00	Job6524



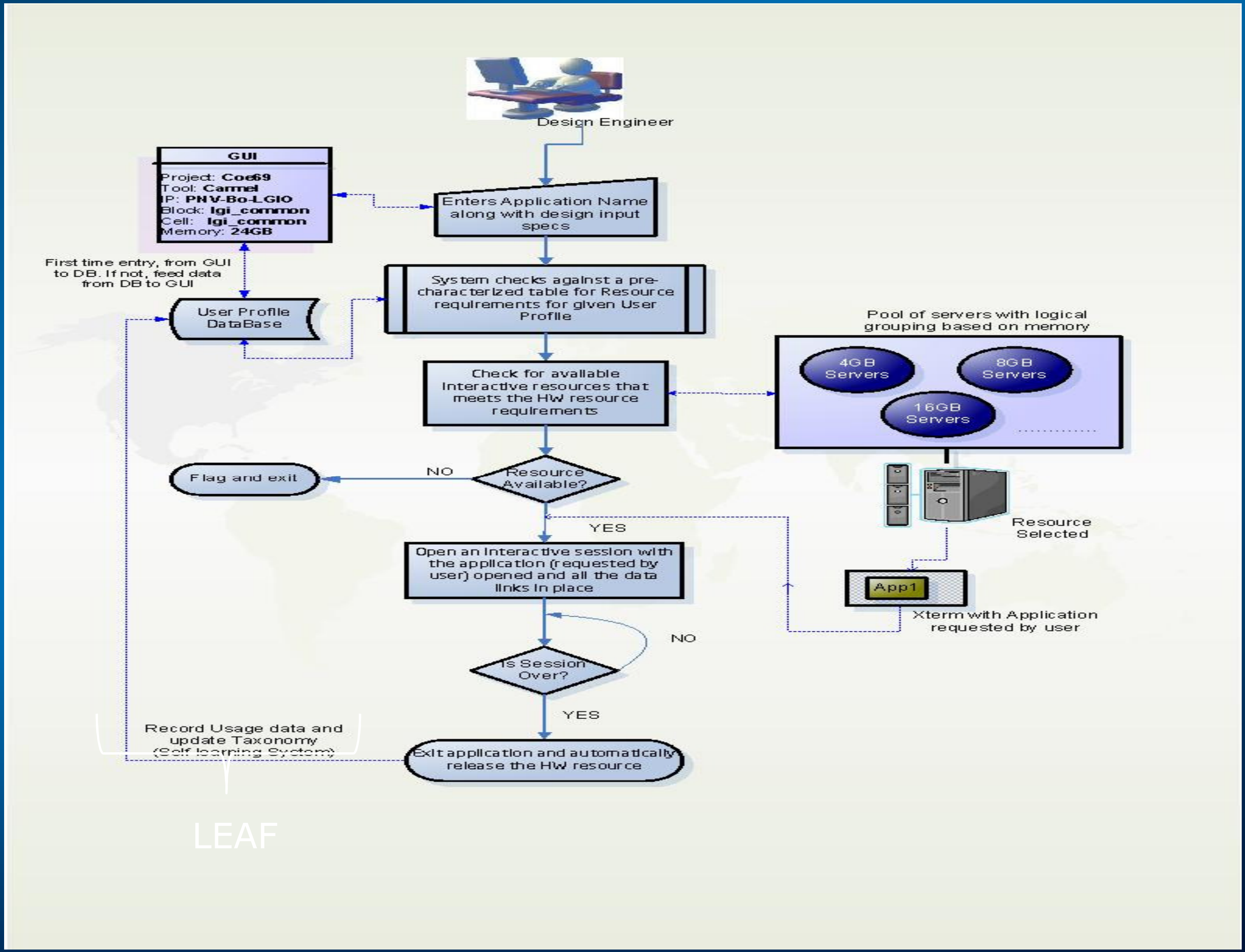
Majority of Interactive applications resource consumption varies based on design inputs and design flows (combination of EDA tools, methodology and process collaterals).

For example in above data, same tool can consume memory between 18GB to 60GB based on Block and Cell Name and type of analysis

System Diagram for TORA



Control Flow for TORA



Assumptions with Proposed Architecture

- User profile in a given project remains static in major project life cycle
- Interactive applications are single threaded in majority but memory needs and runtime are dynamic
- User knows the design inputs (block, cell and tool name he/she will be working on) and approximate resource needs for the first time entry.
- User will have to use a resource brokering system to get a resource allocated (user will not know machine details otherwise)

Benefits

- Maximize sharing of interactive resources by right-sizing interactive sessions
- Minimize disparity between application needs and allocated session
- Optimize purchase decisions by providing access to granular level of usage data
- Minimize system crashes due to load
- Improve design engineer productivity by striking right balance between utilization and performance

Does Proposed solution address all challenges???

- Definite start and end points of session
  - No xterm returned but the actual application ✓
- Control on User sessions
  - Through slot/session restrictions ✓
- User using high end m/c to run low end jobs
  - Any scheduler running in the backend takes care of this automatically ✓
- User running high end job on low end machine
  - Any scheduler running in the backend takes care of this automatically ✓
- Clear Idle session
  - Through idle session detection policies ✓

Long term roadmap…

- Return a VM session with user application
  - Enables user isolation and prevents problems with a single user's session from affecting other sessions
  - Allows user to invoke a CAD tool that needs a non standard OS (say, RH)
  - Allows addition of specific resources (like memory) to an existing VM when required by the application
- Enable session migration to create "high available" environment
  - Conclusively proven session migration capability of VMs – same technology can be used here too

intel®