CMSC 818S Grid Computing

SNAP: A Protocol for Negotiating Service Level Agreements and Coordinating Resource Management in Distributed System

Karl Czajkowshi Ian Foster Carl Kesselman Volker Sander Steven Tuecke

Binh Viet Nguyen

SLA

- A common means for reconciling two competing demands between users and owners' policies.
- To negotiate a service-level agreement (SLA). A resource provider "contracts" with a client to provide some measurable capacity or to perform a task.

Types of SLAs

- Task service level agreements (TSLA): negotiates for the performance of an activity or task. E.g. A TSLA is created by submitting a job description to a queuing system.
- Resource service level agreements (RSLA): negotiate for the right to consume a resource.
- Binding service level agreement: negotiates for the application of a resource to a task.
  - E.g. an RSLA promising network bandwidth might be applied to a particular TCP socket.
  - BSLA associates a task identified by TSLA and the resource capacity by RSLA.

Community Scheduler Scenario

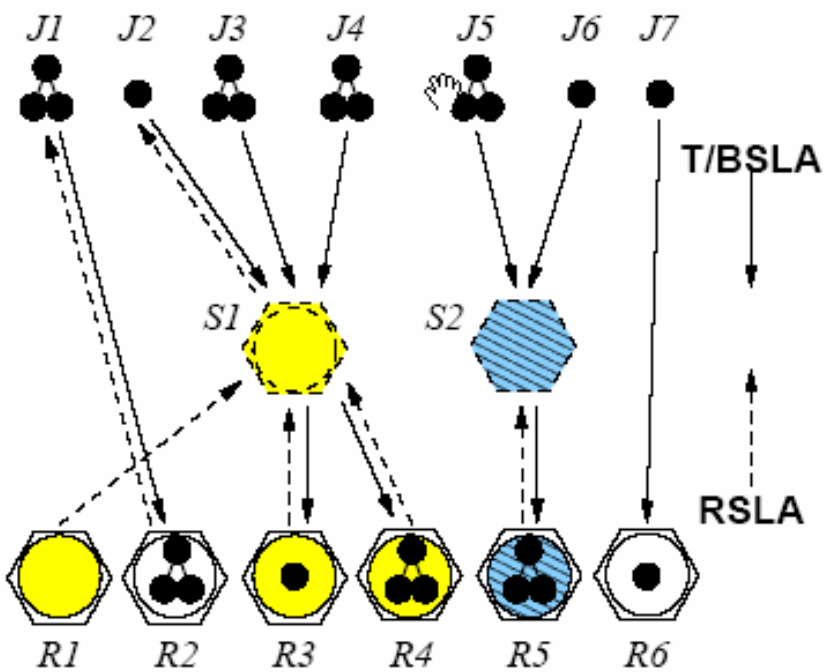

Figure 2. Community scheduler scenario. Multiple users (J1–J7) gain access to shared resources (R1–R6). Community schedulers (S1–S2) mediate access to the resources by making TSLAs with the users and in turn making RSLAs and TSLAs with the individual resources.

File transfer Scenario

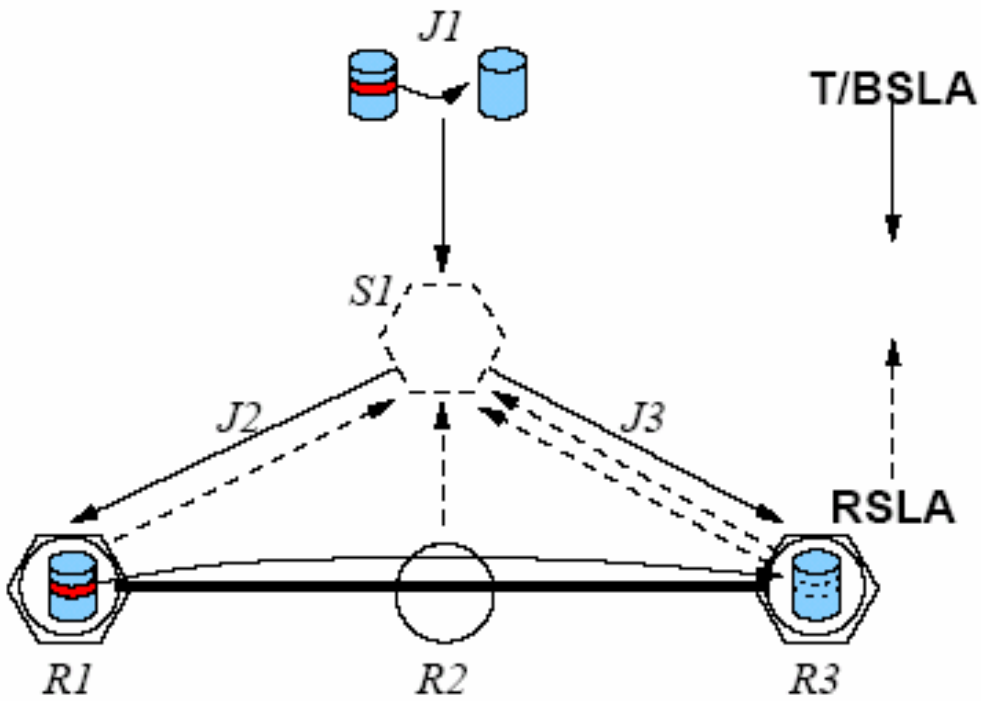

Figure 3. File transfer scenario. File transfer scheduler obtains disk and network reservations before submitting transfer endpoint jobs to perform transfer jobs for clients.

The SNAP Agreement Protocol

- Agreement State Transitions
- Agreement Meta-Language (J language)

Agreement State Transitions

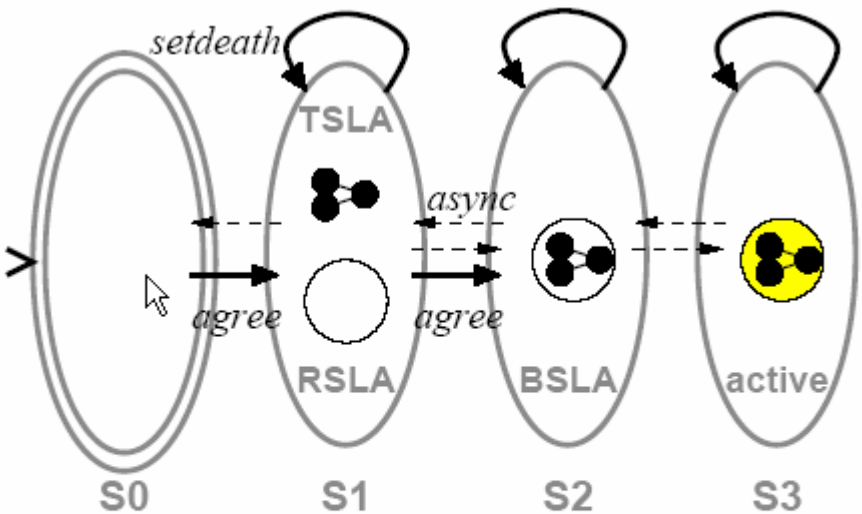

Figure 5. Agreement state transitions. State of SLAs is affected by client requests (solid arrows) and other internal behaviors in the manager (dashed arrows).

- S0: SLAs either have not been created, or have been resolved by expiration or cancellation of the SLAs.
- S1: The TSLAs and RSLAs have been agreed upon, but are not matched with each other.
- S2: The TSLA is matched with the RSLA, and this grouping represents a dependent BSLA to resolve the task
- S3: Resources are being utilized for the task and can still be controlled or changed.

Agreement Meta-Language

- <I,c, t_{dead}, a>
- I: SLA identifier
- c: client with whom the SLA is made.
- t_{dead}: an expiration time
- a: a specific TSLA, RSLA, or BSLA description.

Agreement Meta-Language (2)

- RSLA content: $<l,c, t_{dead}, <r>_R>$
- TSLA content: $<l,c, t_{dead}, <j>_T>$
- BSLA content: $<l,c, t_{dead}, <j>_B>$

Operations

- Allocate Identifier Operation:
  - getident(t_{dead})
  - useident(t_{dead})
- Agreement Operation:
  - request(I,c,t_{dead},a)
  - agree(I,c,t_{dead},a’)
- Set Termination Operation:
  - setdeath(I,t_{dead})
  - willdie(I,t_{dead})

Resource and Task Meta-Language

- Resource metrics
- Resource composites
- Resource Alternatives
- Resource configuration
- RSLA binding

Resource metrics

- Time metrics e.g. “Wed Apr 24 20:52:36 UTC 2002”
- Scalar metrics: x u expressed in x real-valued units u, e.g. 512 Mbytes.
- Max limit < m and <= m specify an exclusive or inclusive upper limit on the given metric m, respectively
- Min limit > m and >= m specify an exclusive or inclusive lower limit on the given metric m, respectively

Resource composites

- **Set [r1,r2..,..]:** combining arbitrary resources that are all required
- **Typed Set [r1,r2..,..]$_{type}$** combining type-specific resources. E.g. [x1 bytes, x2 bytes/s]$_{disk}$
- **Array nxr** is an abbreviation for the group of n identical resource instances [r,r,…,r]
- Complex time-varying description:

$$r = \left[[r_1]^{[t_0,t_1]}, [r_2]^{[t_1,t_2]}, \ldots, [r_n]^{[t_{n-1},t_n]} \right]^{[t_0,t_n]}$$

Others

- Resource Alternatives:
  - Alternative v (r1,r2,…) differs from a resource set in that only one element r_i must be satisfied
- Configuration:
  - Configure a:= v specifies an arbitrary configuration attribute a should have value v
- RSLA binding
  - RSLA binding $[r, I_B]_{bind}$ specifies requirement r but also says it should be satisfied using RSLA identified by I_B

Conclusion

- New model and protocol for managing the process of negotiating resources in a distributed system
- SNAP defines a general framework within which reservation, acquisition, task submission, and binding of task to resources can be expressed for any resource in a uniform fashion.
- SNAP has not been validated and implemented yet.