Sexta semana de trabajo [05 - 09 de Mayo]

Milton Inostroza Aguilera

12 Mayo de 2008

Resumen

Se modifica borrador de protocolo para mejor adpatación entre pyTOD y TOD:

- parentId en el registro de probe cambia de posición.
- typeId se agrega como antecesor de cualquier valor que se envíe a la base de datos de TOD.

Se ha logrado obtener los primeros resultados de la depuración a un script escrito en Python, pero aún existen problemas para ver los registros desde el lado de TOD. Se detectó timestamp's que apuntaban ciclicamente a si mismos.

Indice

Lista de Figuras

Lista de Tablas

1 Desarrollo

Se hicieron pequeños cambios al borrador del protocolo de comunicación específicamente en el momento que se envía un valor desde pyTOD hacia TOD. Se agregó un identificador de tipo de datos para manipularlo de forma correcta, tanto para envíarlo como para recibirlo.

El socket servidor ya está completamente implementado en el lado Java y se comunica de forma satisfactoria con su cliente en el lado Python.

Empieza a ser necesaria la modificación de la máquina virtual de Python o la creación de un tipo de datos nativo especial que permita implementar el identificador único para ver como se comporta pyTOD con este tipo de objetos.

2 Protocolo de comunicación - Draft

Se muestran las tablas bases para el registro de los objetos y para el envío de eventos:

2.1 Identificadores bases

La siguiente tabla muestra que cada suceso tiene un identificador en el sistema de capturación de huella.

Suceso	Identificador
Registro	0
Llamada	1
Retorno	2
Asignacion	3

Tabla 1: Identificadores de sucesos

La siguiente tabla muestra que cada objeto tiene un identificador en el sistema de capturación de huella.

Id Objeto	Identificador
Clase	0
método	1
Atributo	2
Funcion	3
Variable local	4
Probe	5
Thread	6

Tabla 2: Identificadores de objetos

La siguiente tabla muestra que cada tipo de datos un identificador en el sistema de capturación de huella.

Type	Identificador
int	0
str	1
float	2
long	3
bool	4
other	5

Tabla 3: Identificadores de tipo de datos

2.2 Registro de objetos

A continuación se muestra el formato que tienen el registro de los diferentes objetos dentro del capturador de huellas:

Se describe el registro del objeto función:

eventId	objectId	Id	name	argsN	$\{\text{argName}_i$	$\text{argId}_i\}$
int	int	int	str	int	str	int

Tabla 4: Registro del objeto función

Se describe el registro del objeto variable local:

eventId	objectId	Id	parentId	name
int	int	int	int	str

Tabla 5: Registro del objeto variable local

Se describe el registro del objeto clase:

eventId	objectId	Id	name	classBases
int	int	int	str	–[1]

Tabla 6: Registro del objeto clase

[1] Aún no se toma decisión para poder registrar las super clases que pueda tener la clase registrada.

Se describe el registro del objeto método:

eventId	objectId	Id	classId	name	argsN	{argName$_i$	argId$_i$}
int	int	int	int	str	int	str	int

Tabla 7: Registro del objeto método

Se describe el registro del objeto atributo:

eventId	objectId	Id	parentId	name
int	int	int	int	str

Tabla 8: Registro del objeto atributo

Se describe el registro del objeto thread:

eventId	objectId	Id	sysId
int	int	int	int

Tabla 9: Registro del objeto thread

Se describe el registro del objeto probe:

eventId	objectId	Id	parentId	currentLasti
int	int	int	int	int

Tabla 10: Registro del objeto probe

2.3 Llamada de objetos

A continuación se muestra el formato que tienen las llamadas de los objetos función y método dentro del capturador de huellas:

Se describe la llamada al objeto función:

eventId	objectId	Id	parentId	argsN	$typeId_i$	$argValue_i$
int	int	int	int	int	–[1]	

Tabla 11: Llamada al objeto función

Se describe la llamada al objeto método:

eventId	objectId	Id	parentId	classId	argsN	$typeId_i$	$argValue_i$
int	int	int	int	int	int	–[1]	

Tabla 12: Llamada al objeto método

Es importante señalar que todas estas llamadas estan acompañadas de los siguientes datos que se describen a continuación:

probeId	parentTimeStampFrame	depth	currentTimeStamp	threadId
int	double	int	double	int

Tabla 13: Coordenadas

[1] Aún no se toma decisión para poder almacenar los valores de objetos primitivos de Python como son: listas, tuplas, diccionarios, enumeraciones.

2.4 Asignación - Modificación de objetos

A continuación se muestra el formato que tienen las asignaciones — modificaciones de los objetos variable local y atributo dentro del capturador de huellas:

Se describe la asignación - modificacion al objeto variable local:

eventId	objectId	Id	parentId	typeId	value
int	int	int	int	–[1]	

Tabla 14: Registro del objeto variable local

Se describe la asignación - modificación al objeto atributo:

eventId	objectId	Id	parentId	typeId	value
int	int	int	int	–[1]	

Tabla 15: Registro del objeto atributo

Es importante señalar que todas estas asignaciones - modificaciones estan acompañadas de los siguientes datos que se describen a continuación:

probeId	parentTimeStampFrame	depth	currentTimeStamp	threadId
int	double	int	double	int

Tabla 16: Coordenadas

2.5 Return

A continuación se muestra el formato que tiene el return dentro del capturador de huellas:

Se describe return:

eventId	typeId	value	probeId	hasThrown
int	–[1]	int	bool	

Tabla 17: Registro de return

[1] Aún no se toma decisión para poder almacenar los valores de objetos primitivos de Python como son: listas, tuplas, diccionarios, enumeraciones.

3 Primeras depuraciones

Se ha logrado realizar las primeras depuraciones, pero aún no son lo completas que deseamos. Se está trabajando en encontrar las incompatibilidades que se están generando entre el modelo actual de huella de TOD con el modelo propuesto de huella de pyTOD. Aún teniendo ciertos problemas y bugs es posible ver algunas opciones gráficas ya disponibles.

Se ha depurado parcialmente el siguiente trozo de código:

```
class clase1(Descriptor):

    def __init__(self, y):
        self.x = 1
        self.c = 2
        self.z = 3
        return

    def metodo(self, h, i, j, k):
        self.casa = 1
        k = i + j
        return
```

TOD, nos muestra la siguiente información:

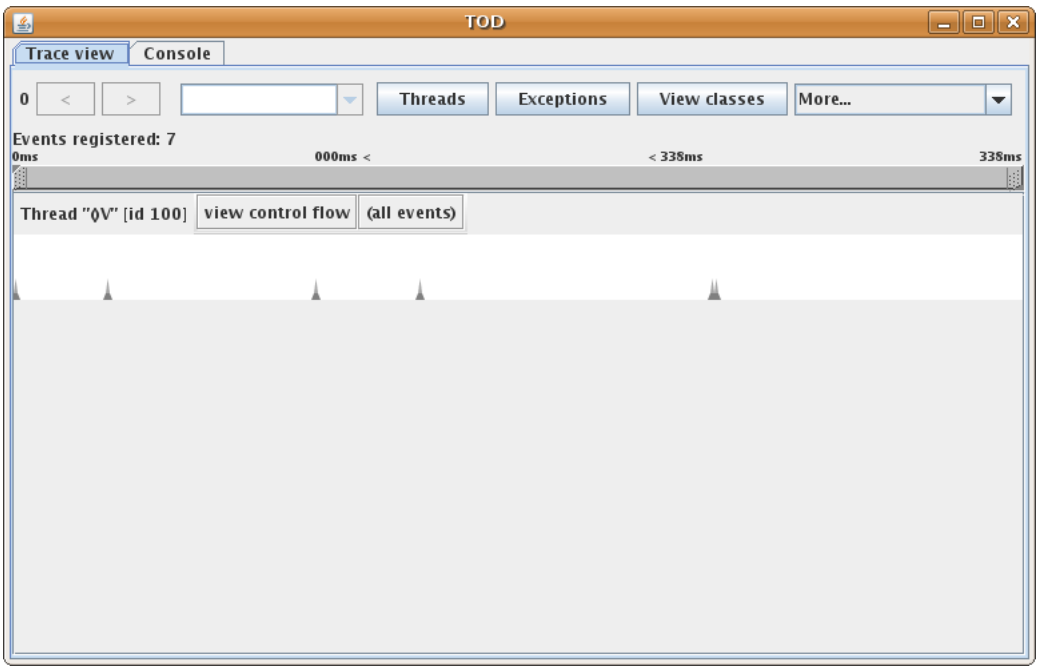

Figura 1: Pantalla principal de TOD

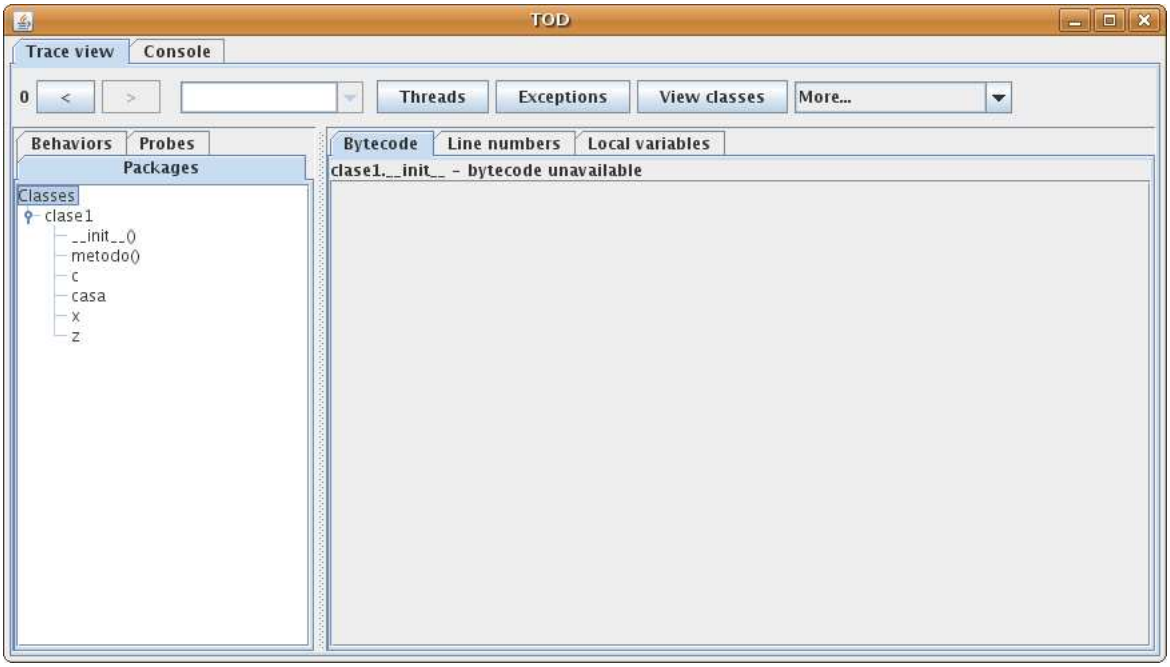

Figura 2: Clases son sus métodos y atributos

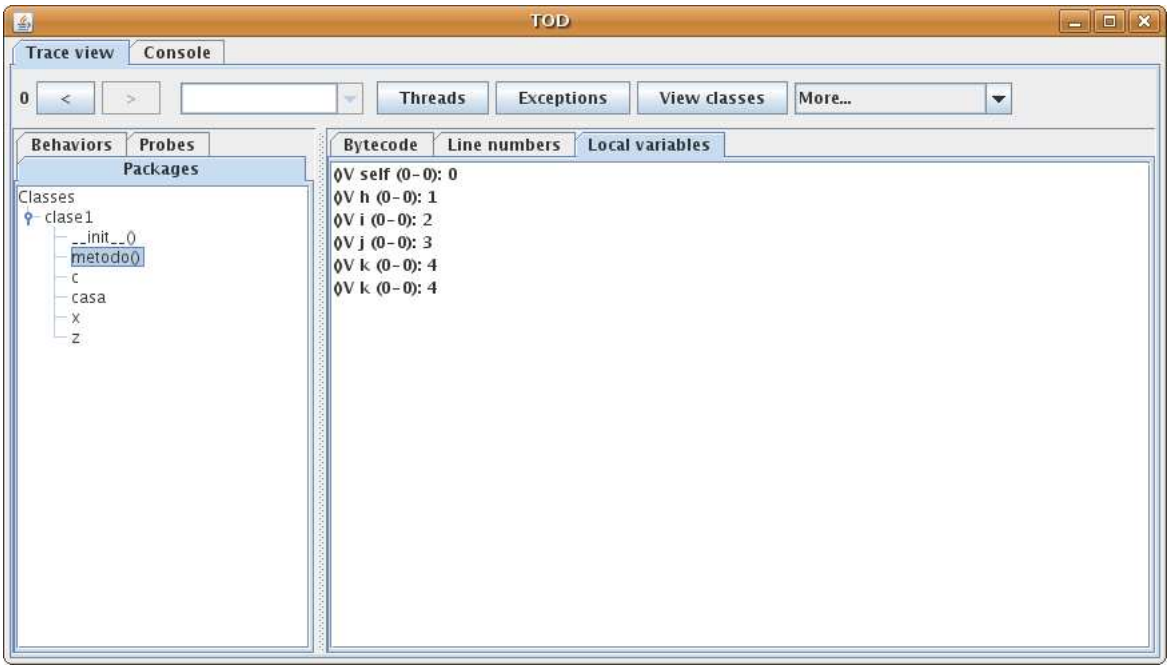

Figura 3: Métodos con sus variables locales

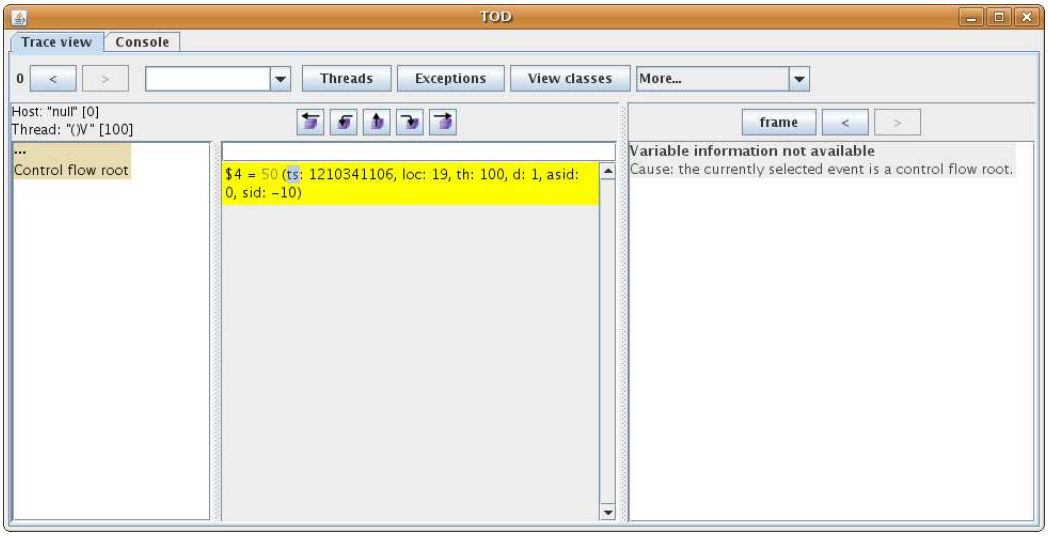

Figura 4: Vista de los eventos ocurridos

Lamentablemente en esta última pantalla se debieran ver siete eventos y sólo se ve uno. Esto es parte del problema descrito anteriormente.

Bibliografía

[1] `http://docs.python.org/lib/built-in-funcs.html`

[2] `http://docs.python.org/ext/ext.html`